Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 298 (2013) 257–282

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Rational Operational Models

Stefan Milius[a,1] Marcello M. Bonsangue[b,2]
Robert S.R. Myers[c,3] Jurriaan Rot[b,4]

[a] *Lehrstuhl für Theoretische Informatik, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany*

[b] *LIACS – Leiden University and Centrum Wiskunde en Informatica (CWI), The Netherlands*

[c] *Institut für Theoretische Informatik, TU Braunschweig, Germany*

Abstract

GSOS is a specification format for well-behaved operations on transition systems. Aceto introduced a restriction of this format, called *simple GSOS*, which guarantees that the associated transition system is locally finite, i.e. every state has only finitely many different descendent states (i.e. states reachable by a sequence of transitions).

The theory of *coalgebras* provides a framework for the uniform study of systems, including labelled transition systems but also, e.g. weighted transition systems and (non-)deterministic automata. In this context GSOS can be studied at the general level of distributive laws of syntax over behaviour. In the present paper we generalize Aceto's result to the setting of coalgebras by restricting abstract GSOS to *bipointed specifications*. We show that the operational model of a bipointed specification is locally finite, even for specifications with infinitely many operations which have finite dependency. As an example, we derive a concrete format for operations on regular languages and obtain for free that regular expressions have finitely many derivatives modulo the equations of join semilattices.

Keywords: coalgebra, distributive law, regular process, simple GSOS, rational behaviour

---

1 Introduction

GSOS [13] is a popular specification format for operations on transition systems, which guarantees that bisimilarity is a congruence. Every GSOS specification induces an *operational model*, which is a concrete transition system on the closed terms of the syntax. Aceto's *simple GSOS* [1] is a restriction of this format which guarantees the operational model to be locally finite. This means that any state in this model is contained in a finite subsystem, i.e. it has only finitely many different descendent states. Consequently, the behaviour of each term is some kind

---

[1] Email: mail@stefan-milius.eu

[2] Email: marcello@liacs.nl

[3] Email: me.robmyers@gmail.com

[4] Email: jrot@liacs.nl. The research of this author has been funded by the Netherlands Organisation for Scientific Research (NWO), CoRE project, dossier number: 612.063.920.

http://dx.doi.org/10.1016/j.entcs.2013.09.017

of regular tree modulo bisimilarity. Simple GSOS rules differ from ordinary GSOS in that the target of a conclusion is either a single operation or a variable, rather than an arbitrary term. Moreover, while the number of operations can be infinite, each operation may only depend on finitely many others. Most operations used in practice can be specified in simple GSOS [2].

Operations that preserve finiteness are of considerable importance in automata theory. In order to provide a uniform mathematical treatment of operations on different types of systems, including those from automata theory, we use the theory of universal coalgebra, where the type of a system is completely specified by an endofunctor F. In this context, the *rational fixpoint* of an endofunctor F on Set is the subcoalgebra of the final F-coalgebra which consists of the behaviours of all finite F-coalgebras. *Bipointed specifications* were introduced in [14] as a format which, for a given finite signature of operations with finite arity, defines algebraic operations on the rational fixpoint. This provides an easy syntactic criterion for the preservation of finite behaviour in terms of a format which is a restriction of Turi and Plotkin's generalization of GSOS via distributive laws [31,20]. Under the assumption that the signature is finite, bipointed specifications for labelled transition systems coincide with simple GSOS. However, the operational model was not considered in [14].

In this paper we complete the generalization of Aceto's results: (a) we extend the results of [14] from specifications for finitely many algebraic operations to specifications that may define infinitely many operations, but with *finite dependency* (cf. [2]); (b) we prove that for a bipointed specification having finite dependency its operational model is locally finite. Result (a) allows e.g. to treat *all* real numbers as constants in the stream calculus [28], while (b) gives a construction of a finite model for each term, thus paving the way for decidability results.

For the Set functor whose coalgebras are deterministic automata, the rational fixpoint is carried by the set of regular languages. At this point one might expect that all the operators of regular expressions might be specified by bipointed specifications for this functor. However, the corresponding rule format is not expressive enough to capture concatenation or the Kleene star. So as a final result we derive a concrete rule format for operations on regular languages, by instantiating our results in the category of join semilattices. Operations defined by rules in this format preserve regular languages, examples being the shuffle product or sequential composition. In fact, the format allows us to define the behaviour of regular expressions. Consequently we obtain for free the well-known result [16] that regular expressions modulo the axioms of join semilattices have only finitely many derivatives.

2 Preliminaries

We assume that the reader is familiar with basic notions from category theory, including (initial) algebras and (final) coalgebras for endofunctors. Let us now fix notation and briefly mention some examples. We denote by Set the category of sets and functions and by Jsl the category of join semilattices and their morphisms.

We denote the initial algebra for a functor $\Sigma : \mathcal{A} \to \mathcal{A}$ by $\iota : \Sigma(\mu\Sigma) \to \Sigma$. In most cases in this paper, Σ will be a polynomial functor on Set given by a (finitary, yet not necessarily *finite*) signature of operation symbols, each with prescribed finite arity. Algebras and homomorphisms for such a functor are precisely the general algebras and homomorphisms for the signature.

The final coalgebra for a functor $F : \mathcal{A} \to \mathcal{A}$ is denoted by $t : \nu F \to F(\nu F)$. We consider several examples of coalgebras for $\mathcal{A} = \mathsf{Set}$ (see [27] for many more):

Example 2.1 (1) Deterministic automata with input alphabet A are coalgebras for $FX = 2 \times X^A$, where $2 = \{0, 1\}$. The final coalgebra is carried by the set of formal languages $\mathcal{P}(A^*)$.

(2) Finitely branching labelled transition systems (LTS) with actions from the set A are coalgebras for $FX = \mathcal{P}_{\mathsf{f}}(A \times X)$, where $\mathcal{P}_{\mathsf{f}}$ is the finite powerset functor. The final coalgebra for F exists and can be thought of as consisting of processes modulo strong bisimilarity of Milner [25].

(3) Weighted transition systems (WTS) are labelled transition systems where transitions have weights (modelling multiplicities, costs, probabilities, etc.) in a monoid $\mathbb{M} = \langle M, +, 0\rangle$. They can be seen as coalgebras (see e.g. Klin [19]): one considers the functor $\mathcal{F}_{\mathbb{M}}$, which acts on a set X and a function $f : X \to Y$ as $\mathcal{F}_{\mathbb{M}}(X) = \{\phi : X \to M \mid \phi \text{ has finite support}\}$ and $\mathcal{F}_{\mathbb{M}}(f)(\phi)(y) = \sum_{x \in f^{-1}(y)} \phi(x)$. Weighted transition systems with actions from the set A are then precisely coalgebras for $FX = (\mathcal{F}_{\mathbb{M}}X)^A$.

2.1 Locally finitely presentable coalgebras.

We are interested in algebraic operations on regular behaviour, i.e. behaviour of *finite* coalgebras (S, f) for a functor F. As previously in [14] we present our results for endofunctors on general categories $\mathcal{A}$ in which it makes sense to talk about "finite" objects and the ensuing rational behaviour of "finite" coalgebras. So we work with the *locally finitely presentable* categories of Gabriel and Ulmer [17] (see also Adámek and Rosický [7]), and we now briefly recall the basics.

A functor $F : \mathcal{A} \to \mathcal{B}$ is called *finitary* if $\mathcal{A}$ has and F preserves filtered colimits. An object X of a category $\mathcal{A}$ is called *finitely presentable* if its hom-functor $\mathcal{A}(X, -)$ is finitary. A category $\mathcal{A}$ is *locally finitely presentable* (lfp) if (a) it is cocomplete, and (b) it has a set of finitely presentable objects such that every object of $\mathcal{A}$ is a filtered colimit of objects from that set.

Example 2.2 (1) Examples of lfp categories include the category Set, the category of posets and monotone functions, and the category of (multi)graphs and graph morphisms. Their finitely presentable objects are the finite sets, finite posets and finite graphs, respectively.

(2) Fix any finitary signature and also a set of equations between terms over this signature. This induces a finitary variety, i.e. a category whose objects are the algebras for this signature which satisfy the equations, e.g. groups, monoids, join semilattices etc. Its morphisms are the usual algebra morphisms for the signature. Such categories are lfp: the finitely presentable objects are those algebras presented by finitely many generators and finitely many relations.

(3) As a special case consider *locally finite* varieties, where the free algebras on finitely many generators are finite. Examples include join semilattices, distributive lattices, boolean algebras and the two-sorted variety of multigraphs. Here the finitely presentable objects are precisely the finite algebras.

(4) Another special case of point (2) is the category $\mathsf{Vec}_{\mathbb{F}}$ of vector spaces over any fixed field $\mathbb{F}$, where the finitely presentable objects are precisely the finite dimensional vector spaces.

Remark 2.3 On the category Set, a finitary functor is determined by its behaviour on finite sets. More precisely, a functor $F : \mathsf{Set} \to \mathsf{Set}$ is finitary iff it is *bounded* (see, e.g. Adámek and Trnková [10]), i.e. for every set X and every element $t \in FX$, there exists a finite subset $i : Y \hookrightarrow X$ such that $t \in Fi[FY] \subseteq FX$.

Example 2.4 The finite powerset functor $\mathcal{P}_{\mathsf{f}}$ is finitary, whereas the ordinary powerset functor $\mathcal{P}$ is not. The functor $FX = X^A$ is finitary if and only if A is a finite set. More generally, the class of finitary endofunctors on Set contains all constant functors and the identity functor, and it is closed under finite products, arbitrary coproducts and composition. Thus, a polynomial functor Σ is finitary iff every operation symbol of the corresponding signature has finite arity (but there may be infinitely many operations). The functor $FX = \mathbb{R} \times X$ is finitary both on Set and on $\mathsf{Vec}_{\mathbb{R}}$.

Assumption 2.5 Throughout the rest of this paper, we assume, unless stated otherwise, that $\mathcal{A}$ is a locally finitely presentable category and $F : \mathcal{A} \to \mathcal{A}$ is a finitary functor. So F has a final coalgebra $t : \nu F \to F(\nu F)$ (see Makkai and Paré [23]).

For a functor F on an lfp category $\mathcal{A}$, the notion of a "finite" coalgebra is captured by requiring the carrier to be finitely presentable. That is, we denote by $\mathsf{Coalg}_{\mathsf{f}}(F)$ the full subcategory of $\mathsf{Coalg}(F)$ consisting of those F-coalgebras $f : S \to FS$ whose carrier S is a finitely presentable object in $\mathcal{A}$. In order to talk about the behaviour of finite coalgebras in this setting, we would like to consider a coalgebra that is final amongst all coalgebras in $\mathsf{Coalg}_{\mathsf{f}}(F)$. However, $\mathsf{Coalg}_{\mathsf{f}}(F)$ need not have a final object; for example, in the case of deterministic automata (see Example 2.1(1)), the desired final coalgebra for finite automata should be formed by all regular languages, but this coalgebra is itself not finite. For this reason we take the closure of $\mathsf{Coalg}_{\mathsf{f}}(F)$ under filtered colimits in $\mathsf{Coalg}(F)$, in which the desired final object exists. It is often useful to view these filtered colimits as directed unions of machines, taken at the level of their carrier. We will write $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ for this closure. The objects of $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ were called *locally finitely presentable* coalgebras in [24,15,14]; they are precisely the filtered colimits of diagrams over $\mathsf{Coalg}_{\mathsf{f}}(F)$, i.e. colimits of filtered diagrams of the form $\mathcal{D} \to \mathsf{Coalg}_{\mathsf{f}}(F) \hookrightarrow \mathsf{Coalg}(F)$.

Example 2.6 We recall from [24,15] concrete descriptions of the objects of $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ in some categories of interest.

(1) A coalgebra for a functor on Set is locally finitely presentable iff it is *locally finite*, i.e. every finite subset of its carrier is contained in a finite subcoalgebra.

(2) For an endofunctor on a locally finite variety, a coalgebra is locally finitely presentable iff every finite subalgebra of its carrier lies in a finite subcoalgebra.

(3) A coalgebra (S, f) for a functor on $\mathsf{Vec}_{\mathbb{F}}$ is locally finitely presentable iff every finite dimensional subspace of its carrier S is contained in a subcoalgebra of (S, f) whose carrier is finite dimensional.

Recall from [23], that the Ind-completion of a category is the free completion of that category under filtered colimits. We will make use of the following non-trivial fact:

Theorem 2.7 *The category* $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ *is the* Ind*-completion of* $\mathsf{Coalg}_{\mathsf{f}}(F)$.

Proof We use a result from Johnstone's book [18] i.e. the theorem in Subsection VI.1.8. This theorem states that if (a) the category $\mathcal{C}$ has finite colimits, and (b) $I : \mathcal{C} \to \mathcal{E}$ is a full embedding into a cocomplete category $\mathcal{E}$ whose image consists of finitely presentable objects in $\mathcal{E}$, then the unique filtered colimit preserving extension $I^* : \mathsf{Ind}(\mathcal{C}) \to \mathcal{E}$ is also a full embedding.

So let $\mathcal{E}$ be $\mathsf{Coalg}(F)$ which is certainly cocomplete, and let $\mathcal{C}$ be $\mathsf{Coalg}_{\mathsf{f}}(F)$. First of all, $\mathcal{C}$ has finite colimits. For a finite colimit of objects from $\mathsf{Coalg}_{\mathsf{f}}(F)$ evaluated in $\mathsf{Coalg}(F)$ gives another object in $\mathsf{Coalg}_{\mathsf{f}}(F)$ (since colimits are constructed in the base category and finitely presentable objects are closed under finite colimits). Then since $\mathsf{Coalg}_{\mathsf{f}}(F)$ is a full subcategory, these colimits restrict.

Secondly, from [6] we know that for any finitary functor F on an lfp category, those F-coalgebras with finitely presentable carrier are actually finitely presentable objects in $\mathsf{Coalg}(F)$.

Then we can apply the theorem from [18]: the unique (filtered colimit preserving) extension of the full embedding $I : \mathcal{C} \to \mathcal{E}$ is itself a full embedding $I^* : \mathsf{Ind}(\mathcal{C}) \to \mathcal{E}$. The definition of this extension is that it takes formal filtered diagrams of objects in $\mathcal{C}$ and constructs their colimit. Therefore its image is precisely $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ (as we defined it), so restricting we obtain the desired equivalence. □

2.2 The rational fixpoint.

The category $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ has a final object given by the filtered colimit of the inclusion functor $\mathsf{Coalg}_{\mathsf{f}}(F) \hookrightarrow \mathsf{Coalg}(F)$. We denote this coalgebra by $r : \rho F \to F(\rho F)$. This coalgebra captures the behaviour of all coalgebras in $\mathsf{Coalg}_{\mathsf{f}}(F)$. It has been shown in [5] that it is a fixpoint of F, i. e., its structure morphism r is an isomorphism. Following [24,15] we call the coalgebra $(\rho F, r)$ the *rational fixpoint* of F.

Remark 2.8 For $\mathcal{A} = \mathsf{Set}$ the rational fixpoint ρF is the union of all images $f^\dagger[S] \subseteq \nu F$, where $f : S \to FS$ ranges over the *finite* F-coalgebras and $f^\dagger : S \to \nu F$ is the unique coalgebra homomorphism (see [5, Proposition 4.6 and Remark 4.3]). So, in particular, we see that ρF is a subcoalgebra of νF.

For endofunctors on different categories than Set, this need not be the case as shown in [15, Example 3.15]. However, for functors preserving monomorphisms on categories of vector spaces over a field and on locally finite varieties such as Jsl the rational fixpoint always is a subcoalgebra of νF (see [15, Proposition 3.12]).

Example 2.9 We give a number of examples of ρF; for more, see [5,15].

(1) For the functor $FX = \mathbb{R} \times X$ on Set whose final coalgebra is carried by the set $\mathbb{R}^\omega$ of all streams over $\mathbb{R}$, the rational fixpoint consists of all streams that are *eventually periodic*, i.e., of the form $\sigma = vwwww\dots$ for words $v \in \mathbb{R}^*$ and $w \in \mathbb{R}^+$. For the similar functor $FV = \mathbb{R} \times V$ on the category of vector spaces over $\mathbb{R}$, the rational fixpoint consists of all *rational streams* (e. g., Rutten [29]).

(2) The carrier of the rational fixpoint of the deterministic automata functor $FX = 2 \times X^A$ is the set of all languages accepted by *finite* automata, viz. the set of all *regular* languages. If we define F instead on the category Jsl of join semilattices, its rational fixpoint is still given by all regular languages, this time with the join semilattice structure given by union and $\emptyset$.

(3) For $FX = \mathcal{P}_\mathsf{f}(A \times X)$ on Set the rational fixpoint contains all *finite-state* processes (modulo bisimilarity); more precisely, ρF is the coproduct of all *finite* F-coalgebras modulo the largest bisimulation.

(4) For the functor $FX = (\mathcal{F}_\mathbb{M} X)^A$ of weighted transition systems the rational fixpoint is obtained as the coproduct of all finite WTS's modulo weighted bisimilarity.

2.3 Bipointed specifications.

In [14] we introduced *bipointed specifications*, which are natural transformations of the form $\Sigma(F \times Id) \to F(\Sigma + Id)$, where $\Sigma : \mathcal{A} \to \mathcal{A}$ is a given functor. We also showed that for Σ a polynomial endofunctor for a finite signature on Set and for $FX = \mathcal{P}_\mathsf{f}(A \times X)$ bipointed specifications are equivalent to transition system specifications in the simple GSOS format of Aceto [1]. In order to understand Aceto's theorem below and to give a first intuition on bipointed specifications we now recall GSOS and simple GSOS. Given a signature Σ, a GSOS rule for an operator $f \in \Sigma$ of arity n is of the form (1)

$$\frac{\{x_{i_j} \xrightarrow{a_j} y_j\}_{j=1..m} \qquad \{x_{i_k} \overset{b_k}{\not\to}\}_{k=1..l}}{f(x_1, \dots, x_n) \xrightarrow{c} t} \tag{1}$$

where m is the number of positive premises, l is the number of negative premises, and $a_1, \dots, a_m, b_1, \dots, b_l, c \in A$ are labels. The variables $x_1, \dots, x_n, y_1, \dots, y_m$ are pairwise distinct; let V denote the set of these variables. Finally t is a Σ-term over variables in V. In the *simple GSOS* format, t is restricted to be either a variable in V or a flat term $g(z_1, \dots, z_p)$, where g is a p-ary operation symbol in Σ and $z_1, \dots, z_p \in V$. Additionally there is a finiteness condition on the dependency of operators, which we recall below in Section 4. Examples of GSOS rules which adhere to the simple GSOS format include the parallel operator, choice, action prefixing, relabelling and many more.

In the mathematical operational semantics of Turi and Plotkin [31] (see also Bartels [12]) one considers for a specification in the form of a natural transformation as above (and more general formats; see Klin [20] for an overview) an *operational* model and a *denotational* model. The operational model is an F-coalgebra structure on the initial Σ-algebra $(\mu\Sigma, \iota)$ and the denotational model is given by a Σ-algebra structure on the final F-coalgebra $(\nu F, t)$; we denote those structures by $c : \mu\Sigma \to F(\mu\Sigma)$ and $\alpha : \Sigma(\nu F) \to \nu F$. Notice that c is uniquely determined by the commutativity of the diagram below[5]:

$$\begin{tikzcd}
\Sigma(\mu\Sigma) \arrow[rr, "\iota"] \arrow[d, "\Sigma\langle c,id\rangle"'] & & \mu\Sigma \arrow[d, "c"] \\
\Sigma(F(\mu\Sigma) \times \mu\Sigma) \arrow[r, "\lambda"'] & F(\Sigma(\mu\Sigma) + \mu\Sigma) \arrow[r, "{F[\iota,id]}"'] & F(\mu\Sigma)
\end{tikzcd} \tag{2}$$

Similarly, α is uniquely determined by the commutativity of the "dual" diagram (replacing $\mu\Sigma$ by νF and reversing and renaming arrows as appropriate).

In concrete instances, c provides behaviour on closed terms over the signature of the algebraic operations specified, and α provides the denotational semantics of the algebraic operations as specified by λ, taking input from the final coalgebra.

In the previous paper [14] we assumed that a bipointed specification $\lambda : \Sigma(F \times Id) \to F(\Sigma + Id)$ is given, where Σ is a *strongly* finitary functor [4], i. e., Σ is finitary and it preserves finitely presentable objects.

Example 2.10 (1) The class of strongly finitary functors on Set contains the identity functor, all constant functors on finite sets, the finite power-set functor $\mathcal{P}_\mathsf{f}$, and it is closed under finite products, finite coproducts and composition. A polynomial functor Σ on Set is strongly finitary iff the corresponding signature has finitely many operation symbols of finite arity.

(2) The functor $FX = 2 \times X^A$ is strongly finitary iff A is a finite set.

(3) The type functor $FX = \mathbb{R} \times X$ of stream systems as coalgebras is finitary but not strongly so. However, if we consider F as a functor on $\mathsf{Vec}_\mathbb{R}$, then it is strongly finitary; in fact, for every finite dimensional real vector space V, $\mathbb{R} \times V$ is finite dimensional, too.

The main result in [14] is the following:

Theorem 2.11 *Let λ be a bipointed specification where Σ is strongly finitary. Then there is a unique Σ-algebra structure $\beta : \Sigma(\rho F) \to \rho F$ such that the following diagram commutes:*

$$\begin{tikzcd}
\Sigma(\rho F) \arrow[r, "\Sigma\langle r,id\rangle"] \arrow[d, "\beta"'] & \Sigma(F(\rho F) \times \rho F) \arrow[r, "\lambda_{\rho F}"] & F(\Sigma(\rho F) + \rho F) \arrow[d, "{F[\beta,id]}"] \\
\rho F \arrow[rr, "r"] & & F(\rho F)
\end{tikzcd} \tag{3}$$

It then follows that the unique F-coalgebra homomorphism $(\rho F, r) \to (\nu F, t)$ is a Σ-algebra homomorphism from $(\rho F, \beta) \to (\nu F, \alpha)$. So in those cases where ρF is a subcoalgebra of νF, β is a restriction of α to ρF. This shows that the rational fixpoint is closed under operations on the denotational model specified by bipointed specifications.

---

[5] In diagrams we will omit indices of natural transformations (here λ) indicating the component.

In [14], we also provided a number of applications, which we briefly recall. In each case Σ is a polynomial functor for a finite signature.

Labelled transition systems. As already mentioned in the discussion above, for $FX = \mathcal{P}_{\mathsf{f}}(A \times X)$ and a polynomial endofunctor Σ on Set corresponding to a finite signature, bipointed specifications correspond precisely to transition system specifications in Aceto's simple GSOS format. As a special case of Theorem 2.11 we thus obtain the well-known result that for a finite signature, finite state processes (i. e., the elements of ρF) are closed under operations specified by simple GSOS rules. This includes for example all CCS combinators and many other operations. But the results on the simple GSOS format are not restricted to finite signatures. So one aim of the present paper is to extend our previous results to infinite signatures, and we do this in Section 4.

Streams. For the functor $FX = \mathbb{R} \times X$ and Σ a polynomial functor, we worked out a concrete rule format which is equivalent to bipointed specifications. So Theorem 2.11 yields the result that the coalgebra ρF of eventually periodic streams is closed under operations specified by rules in our format. Concrete examples include the well-known zipping operation and many others.

Non-deterministic automata. This application considers $FX = 2 \times (\mathcal{P}_{\mathsf{f}}X)^A$, and here we provide a concrete rule format that yields bipointed specifications (but not necessarily conversely). Theorem 2.11 then yields the result that the rational fixpoint ρF (of finite state branching behaviours) is closed under operations specified in our format. This includes examples such as the shuffle product. But one would wish for formats defining operations on formal languages—so our results would then yield that regular languages are closed under such operations. However, if one works out what bipointed specifications mean for deterministic automata (i. e., $FX = 2 \times X^A$), then the format is not powerful enough to capture interesting operations like the shuffle product. So another aim of this paper is to work in the category Jsl in lieu of Set to obtain a more powerful format; we do this in Section 5.

Weighted transition systems. For $FX = (\mathcal{F}_{\mathbb{M}}X)^A$ we obtain a concrete rule format corresponding to bipointed specifications by restricting a general GSOS format for weighted transition system given by Klin [19]. Then Theorem 2.11 specializes to the result that the coalgebra ρF of all finite weighted transitions systems modulo weighted bisimilarity is closed under operations specified in our format.

Remark 2.12 Turi's and Plotkin's original specifications in abstract GSOS format are natural transformations

$$\lambda : \Sigma(F \times Id) \Rightarrow FT_\Sigma,$$

where T_Σ is the free monad on Σ; for a polynomial functor Σ on Set, $T_\Sigma X$ is the set of all terms of operations in Σ over variables of X. Clearly, this format is more general than bipointed specifications where instead of T_Σ we only allow $\Sigma + Id$ in the codomain; for a polynomial functor Σ on Set this is the restriction to terms of depth at most one. Abstract GSOS specifications also induce an operational model, i.e. a Σ-algebra on the final F-coalgebra. However, this algebra usually does not restrict to the rational fixpoint of F; in [14, Example 3.4], we gave an abstract GSOS specification (involving a term of depth two in the rule conclusion) for $FX = \mathbb{R} \times X$ yielding an operation on the final F-coalgebra of streams that does not restrict to the eventually periodic or rational streams (cf. Example 2.9(3)).

3 Operational model and behaviour on free Σ-algebras

We will now make a first step towards proving our main result, the generalization of Aceto's theorem to mathematical operational semantics. We will prove in this section that for a bipointed specification the operational model is a locally finitely presentable coalgebra, our notion of regularity.

Actually, we will prove a more general result concerning free algebras first. In fact, we will show that the free monad on Σ lifts to a functor on $\mathsf{Coalg}_{\mathsf{lfp}}(F)$. This means that for every locally finitely presentable coalgebra (S, f) the free algebra $\hat{\Sigma}S$ of "terms in S" carries an operational model.

Assumption 3.1 In this section we assume that $\lambda : \Sigma(F \times Id) \to F(\Sigma + Id)$ is a bipointed specification, where $F : \mathcal{A} \to \mathcal{A}$ is finitary and $\Sigma : \mathcal{A} \to \mathcal{A}$ a strongly finitary functor on the lfp category $\mathcal{A}$.

Since Σ is (strongly) finitary, on every object X of $\mathcal{A}$ a free Σ-algebra $\hat{\Sigma}X$ exists. As proved by Barr [11], free algebras yield free monads. Indeed, $\hat{\Sigma}$ is the object assignment of a free monad on Σ. Recall from [3] the free algebra construction by which $\hat{\Sigma}X$ is obtained as the colimit of the chain

$$X \xrightarrow{\mathsf{inr}} \Sigma X + X \xrightarrow{\Sigma\mathsf{inr}+X} \Sigma(\Sigma X + X) + X \longrightarrow \cdots \tag{4}$$

Furthermore, it follows that as a functor $\hat{\Sigma}$ can be constructed as the colimit of the chain

$$Id \xrightarrow{\mathsf{inr}} \Sigma + Id \xrightarrow{\Sigma\mathsf{inr}+Id} \Sigma(\Sigma + Id) + Id \longrightarrow \cdots \tag{5}$$

More precisely, we define functors $T^n : \mathcal{A} \to \mathcal{A}$, $n < \omega$, by induction: $T^0 = Id$ and $T^{n+1} = \Sigma T^n + Id$. The connecting natural transformations are defined by $t_{0,1} = \mathsf{inr}$ and $t_{n+1,n+2} = \Sigma t_{n,n+1} + Id$. In order to prove the main result of this section further below we first need the following auxiliary property

Lemma 3.2 *The chain (5) lifts to a chain of endofunctors on* $\mathsf{Coalg}_{\mathsf{f}}(F)$.

Proof We will prove that each functor T^n lifts to an endofunctor on $\mathsf{Coalg}(F)$ and that each connecting natural transformation $t_{n,n+1} : T^n \to T^{n+1}$ is a natural transformation between the lifted functors. That these lifted functors restrict to $\mathsf{Coalg}_{\mathsf{f}}(F)$ is easy to see by induction on n using that Σ preserves finitely presentable objects and that these objects are closed under finite coproducts.

(1) T^n lifts to $\mathsf{Coalg}(F)$. This is proved by induction on n. The base case $T^0 = \mathrm{Id}$ is trivial. For the induction step let an F-coalgebra (S, f) be given and let $T^n(S, f) = (T^nS, f_n)$. Now define $T^{n+1}(S, f) = (T^{n+1}S, f_{n+1})$ to be the following

F-coalgebra

$$\begin{tikzcd}
T^{n+1}S = \Sigma T^n S + S \arrow[r, "\Sigma\langle f_n, id\rangle + f"] & \Sigma(FT^nS \times T^nS) + FS \arrow[d, "\lambda + id"] \\
FT^{n+1}S = F(\Sigma T^n S + S) & F(\Sigma T^n S + T^n S) + FS \arrow[l, "{[F[\mathsf{inl}, t_{n,n+1}], F\mathsf{inr}]}"]
\end{tikzcd}$$

That T^{n+1} is functorial can be proved with a straightforward diagram chase using that T^n is functorial as well as naturality of λ and $t_{n,n+1}$. We omit the details.

We also omit the details of the proof that $t_{n,n+1} : T^n \to T^{n+1}$ is a natural transformation between lifted functors. Here one must prove that each component is an F-coalgebra homomorphism, and this is done by induction on n. $\square$

Theorem 3.3 *The free monad $\hat{\Sigma} : \mathcal{A} \to \mathcal{A}$ lifts to a functor on $\mathsf{Coalg}_{\mathsf{lfp}}(F)$.*

Proof (1) $\hat{\Sigma}$ lifts to $\mathsf{Coalg}(F)$. By Lemma 3.2, all the functors T^n in the chain (5) lift to $\mathsf{Coalg}(F)$. Now colimits of functors are computed objectwise and the forgetful functor $\mathsf{Coalg}(F) \to \mathcal{A}$ creates all colimits. This implies that the colimit $\hat{\Sigma}$ of the chain (5) canonically lifts to a functor on $\mathsf{Coalg}(F)$.

(2) $\hat{\Sigma}$ restricts to $\mathsf{Coalg}_{\mathsf{lfp}}(F)$. Let (S, f) be a coalgebra in $\mathsf{Coalg}_{\mathsf{f}}(F)$, i. e., S is a finitely presentable object of $\mathcal{A}$. By the point (1), the F-coalgebra $\hat{\Sigma}S$ is obtained as the filtered colimit of the F-coalgebras carried by T^nS in the chain (4), $n = 0, 1, 2, \ldots$, which all lie in $\mathsf{Coalg}_{\mathsf{f}}(F)$. Thus, $\hat{\Sigma}S$ lies in $\mathsf{Coalg}_{\mathsf{lfp}}(F)$, and we have a restriction $\hat{\Sigma} : \mathsf{Coalg}_{\mathsf{f}}(F) \to \mathsf{Coalg}_{\mathsf{lfp}}(F)$. Since $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ is the Ind-completion of $\mathsf{Coalg}_{\mathsf{f}}(F)$, there is (up to equivalence) a unique extension of $\hat{\Sigma}$ to an endofunctor on $\mathsf{Coalg}_{\mathsf{lfp}}(F)$. $\square$

Since $\mu\Sigma = \hat{\Sigma}0$, it follows that $\mu\Sigma$ carries *some* F-coalgebra structure that turns it into a locally finitely presentable coalgebra. It remains to show that the coalgebra structure on $\mu\Sigma$ provided by the previous theorem is indeed the structure $c : \mu\Sigma \to F(\mu\Sigma)$ of the operational model from the previous section:

Theorem 3.4 *The operational model of λ is a locally finitely presentable coalgebra.*

Before we proceed to the proof of the theorem let us make a couple of technical remarks.

Remark 3.5 Recall that the operational model c is uniquely determined by the commutativity of Diagram (2). Actually, c is obtained by using the initiality of $\mu\Sigma$ to obtain a unique Σ-algebra homomorphism from $(\mu\Sigma, \iota)$ to the Σ-algebra

$$\Sigma(F(\mu\Sigma) \times \mu\Sigma) \xrightarrow{\langle \lambda, \Sigma\pi_1 \rangle} F(\Sigma(\mu\Sigma) + \mu\Sigma) \times \Sigma(\mu\Sigma) \xrightarrow{F[\iota, id] \times \iota} F(\mu\Sigma) \times \mu\Sigma \qquad (6)$$

It is then easy to prove that this homomorphism must be of the form

$$\langle c, id \rangle : \mu\Sigma \to F(\mu\Sigma) \times \mu\Sigma$$

so that c is uniquely determined by the commutativity of (2).

Remark 3.6 (1) In the case of an initial object $X = 0$ the free algebra chain (4) yields the chain $\Sigma^n 0$ with connecting morphisms $t_{n,n+1} = \Sigma^n u$, where $u : 0 \to \Sigma 0$ is unique. We will denote the colimit injections by $t_n : \Sigma^n 0 \to \mu\Sigma$. Notice also that $T^n(0, u)$ from Lemma 3.2 takes the form $f_n : \Sigma^n 0 \to F(\Sigma^n 0)$.

(2) Any Σ-algebra (A, α) induces a canonical cocone $\alpha_n : \Sigma^n 0 \to A$ on this chain: $\alpha_0 : 0 \to A$ is uniquely determined and

$$\alpha_{n+1} = \Sigma^{n+1} 0 = \Sigma(\Sigma^n 0) \xrightarrow{\Sigma\alpha_n} \Sigma A \xrightarrow{\alpha} A.$$

Furthermore, for every Σ-algebra homomorphism $h : (A, \alpha) \to (B, \beta)$ we have

$$h \cdot \alpha_n = \beta_n \qquad \text{for every } n < \omega. \tag{7}$$

(3) Notice that the unique Σ-algebra homomorphism $h : (\mu\Sigma, \iota) \to (A, \alpha)$ arises as the unique morphism from the colimit $\mu\Sigma$ induced by the canonical cocone, i.e. h is the unique morphism such that the following triangles commute:

$$\begin{tikzcd}
\Sigma^n 0 \arrow[d, "t_n"'] \arrow[dr, "\alpha_n"] & \\
\mu\Sigma \arrow[r, "h"'] & A
\end{tikzcd} \tag{8}$$

Proof (Theorem 3.4) Consider the operational model $c : \mu\Sigma \to F(\mu\Sigma)$ uniquely determined by the commutativity of Diagram (2). To prove the theorem we must show that c is the coalgebra structure on the colimit $\mu\Sigma$ induced by the coalgebra structures $f_n : \Sigma^n 0 \to F(\Sigma^n 0)$. To this end we will show that for every n the outside of the following square commutes:

$$\begin{tikzcd}
\Sigma^n 0 \arrow[r, "f_n"] \arrow[d, "t_n"'] \arrow[dr, "\pi_1 \cdot \alpha_n"] & F(\Sigma^n 0) \arrow[d, "Ft_n"] \\
\mu\Sigma \arrow[r, "c"'] & F(\mu\Sigma)
\end{tikzcd} \tag{9}$$

It then follows that $(\mu\Sigma, c)$ is the filtered colimit of the chain of coalgebras $T^n(0, u) = (\Sigma^n 0, f_n)$ (c.f. the proof of Theorem 3.3), which all have finitely presentable carrier since Σ is strongly finitary. Thus, $(\mu\Sigma, c)$ lies in $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ as desired.

To see that (9) commutes, let $\alpha : \Sigma(F(\mu\Sigma) \times \mu\Sigma) \to F(\mu\Sigma \times \mu\Sigma)$ be the algebra in (6) and consider its canonical cocone $\alpha_n : \mu\Sigma \to F(\mu\Sigma) \times \mu\Sigma$, $n < \omega$. We will prove that the two inner triangles in Diagram (9) commute, where π_1 is the left-hand product projection. Indeed, the lower left-hand triangle follows from (8) with $h = \langle c, id \rangle$ (cf. Remark 3.5). To show the commutativity of the upper right-hand triangle we will now prove that

$$\alpha_n = (\Sigma^n 0 \xrightarrow{\langle f_n, id \rangle} F\Sigma^n 0 \times \Sigma^n \xrightarrow{Ft_n \times t_n} F(\mu\Sigma) \times \mu\Sigma). \tag{10}$$

Let us first consider the right-hand product component. First it is easy to see that $\pi_2 : (F(\mu\Sigma) \times \mu\Sigma, \alpha) \to (\mu\Sigma, \iota)$ is a Σ-algebra homomorphism. Thus, we see that

$$\pi_2 \cdot \alpha_n = \iota_n = t_n,$$

where the two equations hold by (7) and (8), respectively.

We now proceed by induction on n to prove (10). The base case $n = 0$ is obvious, and for the induction step we consider the following diagram (we need only consider the left-hand product component of (10)):

$$\begin{tikzcd}
\Sigma(\Sigma^n 0) \arrow[r, "\Sigma\langle f_n, id\rangle"] \arrow[d, "\Sigma\alpha_n"'] \arrow[rrr, bend left, "f_{n+1}"] & \Sigma(F\Sigma^n 0 \times \Sigma^n 0) \arrow[r, "\lambda"] \arrow[dl, "\Sigma(Ft_n \times t_n)"] & F(\Sigma\Sigma^n 0 + \Sigma^n 0) \arrow[r, "{F[id, t_{n,n+1}]}"] \arrow[d, "F(\Sigma t_n + t_n)"'] & F\Sigma\Sigma^n 0 \arrow[d, "Ft_{n+1}"'] \\
\Sigma(F(\mu\Sigma) \times \mu\Sigma) \arrow[rr, "\lambda"] \arrow[rrr, bend right, "\pi_1 \cdot \alpha"'] & & F(\Sigma(\mu\Sigma) + \mu\Sigma) \arrow[r, "{F[\iota, id]}"] & F(\mu\Sigma)
\end{tikzcd}$$

This diagram commutes: the upper part is the coalgebra structure f_{n+1} from Lemma 3.2 in the special case where $S = 0$, for the left-hand part remove Σ and use the induction hypothesis the middle part commutes by naturality of λ, and for the right-hand part remove F and consider the components of the coproduct in the upper left-hand corner separately (both clearly commute). This completes the proof. □

4 Finite dependency

With Theorem 3.4 we have the main ingredient for generalizing Aceto's theorem for simple GSOS specifications. However, notice that our restriction to strongly finitary functors Σ means that Theorem 3.4 only generalizes Aceto's theorem for the special case of transition system specifications over a *finite* signature of specified operations. Aceto's theorem instead was proved for transition system specifications having *finite dependency*. In this section we briefly recall that concept. Then we generalize finite dependency to bipointed specifications, and we prove that our previous results hold for bipointed specifications having finite dependency.

4.1 GSOS specifications having finite dependency. Let $\mathcal{T}$ be a transition system specification in the GSOS format defining operations in the signature Σ (see [2] and Section 2.3). *Operator dependency* is the smallest transitive relation on Σ which contains a pair (f, g) of operations if there is a rule in $\mathcal{T}$ of the form (1) where g occurs in the term t. We say that $\mathcal{T}$ has *finite dependency* if each operation f of Σ only depends on finitely many other operations.

The *positive trigger* of a rule (1) is the sequence $\langle A_1, \ldots, A_n\rangle$ where each $A_i \subseteq A$ consists of those labels a_j with $i_j = i$, i.e. $x_i \xrightarrow{a_j} y_j$ occurs in the premise of the rule. An operation f is called *bounded* if for every positive trigger there are only finitely many rules with f on the left-hand side of the conclusion. In the following theorem, by the associated transition system of $\mathcal{T}$ we mean the (operational) term model given by the initial Σ-algebra. Regularity means that from every state there are only finitely many other states reachable by a sequence of transitions.

Theorem 4.1 ([2, Theorem 5.28]) *Let $\mathcal{T}$ be a transition system specification in simple GSOS format having finite dependency, where every operation is bounded. Then the associated transition system of $\mathcal{T}$ is regular.*

Example 4.2 A simple example of a transition system specification is given by the prefixing operation for an infinite label alphabet A; the infinite rule set in (11) obviously has finite dependency.

$$\frac{}{a.P \xrightarrow{a} P} \quad (a \in A) \tag{11}$$

4.2 Bipointed specifications having finite dependency. Transition system specifications in simple GSOS format for which every operation f is bounded are in 1-1-correspondence with bipointed specifications $\Sigma(\mathcal{P}_{\mathsf{f}}(A \times Id) \times Id) \to \mathcal{P}_{\mathsf{f}}(A \times (\Sigma + Id))$; in fact, the functor $\mathcal{P}_{\mathsf{f}}$ in the codomain of the bipointed specification models the finitely many transitions specified for f for each positive trigger.

Now we will analyze how finite dependency can be captured on the level of bipointed specifications. Let $\mathcal{T}$ be a transition system specification satisfying the conditions in Theorem 4.1 and let $\lambda : \Sigma(F \times Id) \to F(\Sigma + Id)$ be the corresponding bipointed specification (where Σ is a polynomial endofunctor on Set). Suppose that Γ is a subfunctor of Σ that corresponds to a subsignature that is closed under operator dependency in Σ and let $\mathsf{in}_\Gamma : \Gamma \to \Sigma$ be the corresponding inclusion map. Then there exists a bipointed specification $\lambda_\Gamma : \Gamma(F \times Id) \to F(\Gamma + Id)$ such that in_Γ is a *morphism of bipointed specifications*, i.e. the square on the right commutes. Also every inclusion $m : \Gamma \to \Gamma'$ between closed subsignatures of Σ is a morphism of bipointed specifications; one has $F(m+Id)\cdot\lambda_\Gamma = \lambda_{\Gamma'} \cdot m(F \times Id)$. Recall from Example 2.9(1) that a polynomial functor Γ is strongly finitary iff its associated signature is finite.

$$\begin{tikzcd} \Gamma(F \times Id) \arrow[r, "\lambda_\Gamma"] \arrow[d, "\mathsf{in}_\Gamma(F \times Id)"'] & F(\Gamma + Id) \arrow[d, "F(\mathsf{in}_\Gamma + Id)"] \\ \Sigma(F \times Id) \arrow[r, "\lambda"'] & F(\Sigma + Id) \end{tikzcd} \tag{12}$$

Proposition 4.3 *Let $\mathcal{T}$ be a transition system specification as in Theorem 4.1 and let $\lambda : \Sigma(F \times Id) \to F(\Sigma + Id)$ be its corresponding bipointed specification. Then Σ is the directed union of a diagram of strongly finitary polynomial functors Γ such that there exist λ_Γ as in (12).*

Remark 4.4 Recall the notion of a *closure operator* on a poset $(P, \leq)$. This is a monotone map $x \mapsto \overline{x}$ on P satisfying $x \leq \overline{x}$ and $\overline{\overline{x}} = \overline{x}$. An element $x \in P$ is called *closed* if $x = \overline{x}$.

Proof (Proposition 4.3) We will abuse notation and denote by Σ the signature of the operation symbols specified by $\mathcal{T}$ as well as the associated polynomial functor. For any subsignature Γ of Σ let

$$\overline{\Gamma} = \{f \mid f \text{ depends on some } g \text{ in } \Gamma\}.$$

Then $\Gamma \mapsto \overline{\Gamma}$ is a *closure operator* on the set of subsignatures of Σ. Notice that due to finite dependency the closure $\overline{\Gamma}$ of a finite subsignature is finite. It follows that Σ is the directed union of all its finite closed subsignatures Γ; for Σ is the directed union of all its finite subsignatures and every finite subsignature is contained in a closed finite subsignature. Now the desired result follows because for a closed subsignature Γ of Σ we easily see that there is λ_Γ as in (12). □

The previous proposition states that λ is the directed union of the λ_Γ. In the following definition we consider the colimit of a filtered diagram of bipointed specifications $\lambda_\Gamma : \Gamma(F \times Id) \to F(\Gamma + Id)$, i.e. the bipointed specification for the colimit Σ of all functors Γ from the diagram uniquely determined by the commutativity of the squares (12).

Definition 4.5 Let $F : \mathcal{A} \to \mathcal{A}$ be finitary and $\Sigma : \mathcal{A} \to \mathcal{A}$. A bipointed specification $\lambda : \Sigma(F \times Id) \to F(\Sigma + Id)$ has *finite dependency* if it is the filtered colimit of a diagram of bipointed specifications $\lambda_\Gamma : \Gamma(F \times Id) \to F(\Gamma + Id)$ where each Γ is a strongly finitary functor.

Remark 4.6 (1) One common instance of the above definition is when Σ can be decomposed into a (not necessarily finite) coproduct $\Sigma = \coprod_{i \in I} \Sigma_i$ such that there are bipointed specifications $\lambda_i : \Sigma_i(F \times Id) \to F(\Sigma_i + Id)$ such that (12) commutes with Γ replaced by Σ_i for each $i \in I$. Indeed, Σ is then the filtered colimit of all $\Sigma_J = \coprod_{i \in J} \Sigma_i$, where J ranges over all finite subsets of I with λ_J formed by the obvious "copairing" involving those λ_i with $i \in J$. For a concrete example consider $FX = \mathbb{R} \times X$ and the behavioural differential equation (see [28]) $\hat{r} = r : \hat{r}$ for every $r \in \mathbb{R}$. Then one has $I = \mathbb{R}$ and Σ_i is constant on 1 for all i.

(2) That filtered colimits are necessary in Definition 4.5 is demonstrated by the usual definition of constants in the stream calculus [28]: $[r] = r : [0]$. All constants $[r]$ depend on $[0]$, and therefore the signature can not be decomposed into a coproduct. In the context of simple GSOS rules on transition systems, a similar example can be found by defining infinitely many constants c_n, $n < \omega$, by the axioms $c_{n+1} \xrightarrow{a} c_n$, for some $a \in A$. This specification cannot be decomposed into finite independent parts as in point (1) above.

The following proposition is related to results of Lenisa, Power and Watanabe [22, Section 5] for distributive laws of monads over copointed endofunctors. Indeed, notice that a bipointed specification can equivalently be presented as a distributive law of the free pointed functor $\Sigma + Id$ over the cofree copointed functor $F \times Id$, and the latter gives rise to a distributive law of the free monad on Σ over $F \times Id$. Lenisa, Power and Watanabe show how to combine distributive laws using coproduct; here we consider filtered colimits.

Proposition 4.7 *Let λ be a bipointed specification having finite dependency, and let $(\lambda_\Gamma)_{\Gamma \in \mathcal{D}}$ be as in Definition 4.5. Then, for each Γ, the denotational models $\alpha : \Sigma(\nu F) \to \nu F$ and $\alpha_\Gamma : \Gamma(\nu F) \to \nu F$ of λ and λ_Γ, respectively, satisfy*

$$\alpha_\Gamma = (\Gamma(\nu F) \xrightarrow{\mathsf{in}_\Gamma} \Sigma(\nu F) \xrightarrow{\alpha} \nu F).$$

Proof Recall that for every $\Gamma \in \mathcal{D}$ the denotational model $\alpha_\Gamma : \Gamma(\nu F) \to \nu F$ is uniquely determined by the commutativity of the square

$$\begin{tikzcd}
\Gamma(\nu F) \arrow[r, "\Gamma\langle t,id\rangle"] \arrow[d, "\alpha_\Gamma"'] & \Gamma(F(\nu F) \times \nu F) \arrow[r, "\lambda"] & F(\Gamma(\nu F) + \nu F) \arrow[d, "{F[\alpha_\Gamma,id]}"] \\
\nu F \arrow[rr, "t"'] & & F(\nu F)
\end{tikzcd}$$

and similarly for $\alpha : \Sigma(\nu F) \to \nu F$. So by precomposing the diagram for α by a colimit injection $\mathsf{in}_\Gamma : \Gamma(\nu F) \to \Sigma(\nu F)$ we obtain the commutative diagram

$$\begin{tikzcd}
\Gamma(\nu F) \arrow[r, "\Gamma\langle t,id\rangle"] \arrow[d, "\mathsf{in}_\Gamma"] \arrow[dd, bend right=60, "\alpha_\Gamma"'] & \Gamma(F(\nu F) \times \nu F) \arrow[r, "\lambda_\Gamma"] \arrow[d, "\mathsf{in}_\Gamma(F\times id)"'] & F(\Gamma(\nu F) + \nu F) \arrow[d, "F(\mathsf{in}_\Gamma+id)"'] \arrow[dd, bend left=60, "{F[\alpha_\Gamma,id]}"] \\
\Sigma(\nu F) \arrow[r, "\Sigma\langle t,id\rangle"] \arrow[d, "\alpha"] & \Sigma(F(\nu F) \times \nu F) \arrow[r, "\lambda"] & F(\Sigma(\nu F) + \nu F) \arrow[d, "{F[\alpha,id]}"'] \\
\nu F \arrow[rr, "t"'] & & F(\nu F)
\end{tikzcd}$$

So the desired equation holds by the unicity of α_Γ. □

The following result extends the main result from [14] from the bipointed specifications considered in Section 2.3 to those with finite dependency.

Corollary 4.8 *Let λ be a bipointed specification having finite dependency. Then (a) there is a unique Σ-algebra structure $\beta : \Sigma(\rho F) \to \rho F$ such that the diagram (3) commutes, and (b) the unique F-coalgebra homomorphism $(\rho F, r) \to (\nu F, t)$ is a Σ-algebra homomorphism from $(\rho F, \beta)$ to $(\nu F, \alpha)$.*

Proof Let $(\lambda_\Gamma)_{\Gamma \in \mathcal{D}}$ be as in Definition 4.5.

Ad (a). By Proposition 4.7, we have $\alpha \cdot \mathsf{in}_\Gamma = \alpha_\Gamma$ for each $\Gamma \in \mathcal{D}$ for the denotational models of λ and λ_Γ, respectively. So by Theorem 2.11 we have a unique $\beta_\Gamma : \Gamma(\rho F) \to \rho F$ such that the diagram

$$\begin{tikzcd}
\Gamma(\rho F) \arrow[r, "\Sigma\langle r,id\rangle"] \arrow[d, "\beta_\Gamma"'] & \Gamma(F(\rho F) \times \rho F) \arrow[r, "\lambda_\Gamma"] & F(\Gamma(\rho F) + \rho F) \arrow[d, "{F[\beta_\Gamma,id]}"] \\
\rho F \arrow[rr, "r"] & & F(\rho F)
\end{tikzcd}$$

commutes for every $\Gamma \in \mathcal{D}$. Now recall that the colimit $\Sigma = \mathrm{colim}_{\Gamma \in \mathcal{D}} \Gamma$ of functors is formed objectwise, and so $\Sigma(\rho F)$ is a filtered colimit of the $\Gamma(\rho F)$. It is not difficult to see that the denotational models $\beta_\Gamma : \Gamma(\rho F) \to \rho F$ form a cocone; indeed, to see this let $m : \Gamma \to \Gamma'$ be a connecting natural transformation in $\mathcal{D}$ and consider the following diagram

$$\begin{tikzcd}
\Gamma(\rho F) \arrow[r, "{\Gamma\langle r,id\rangle}"] \arrow[d, "m"] \arrow[dd, bend right=60, "\beta_\Gamma"'] & \Gamma(F(\rho F)\times\rho F) \arrow[r, "\lambda_\Gamma"] \arrow[d, "m(F\times id)"'] & F(\Gamma(\rho F)+\rho F) \arrow[d, "F(m+id)"'] \arrow[dd, bend left=60, "{F[\beta_\Gamma,id]}"] \\
\Gamma'(\rho F) \arrow[r, "{\Gamma'\langle r,id\rangle}"] \arrow[d, "\beta_{\Gamma'}"] & \Gamma'(F(\rho F)\times\rho F) \arrow[r, "\lambda_{\Gamma'}"] & F(\Gamma'(\rho F)+\rho F) \arrow[d, "{F[\beta_{\Gamma'},id]}"'] \\
\rho F \arrow[rr, "r"'] & & F(\rho F)
\end{tikzcd}$$

Its upper left-hand square commutes by the naturality of m, the upper-right-hand square commutes since m is a morphism of bipointed specifications and the lower part as well as the outside of the diagram commute by Theorem 2.11. Thus, the desired equation $\beta_{\Gamma'} \cdot m = \beta_\Gamma$ follows from the unicity of β_Γ. This implies that there exists a unique $\beta : \Sigma(\rho F) \to \rho F$ satisfying $\beta \cdot \mathsf{in}_\Gamma = \beta_\Gamma$. To prove that β is uniquely determined by the commutativity of the diagram in the statement of the Corollary we consider the diagram obtained from the one in the proof of Proposition 4.7 by replacing $(\nu F, t)$ by $(\rho F, r)$ and α by β:

$$\begin{tikzcd}
\Gamma(\rho F) \arrow[r, "{\Gamma\langle r,id\rangle}"] \arrow[d, "\mathsf{in}_\Gamma"] \arrow[dd, bend right=60, "\beta_\Gamma"'] & \Gamma(F(\rho F)\times\rho F) \arrow[r, "\lambda_\Gamma"] \arrow[d, "\mathsf{in}_\Gamma(F\times id)"'] & F(\Gamma(\rho F)+\rho F) \arrow[d, "F(\mathsf{in}_\Gamma+id)"'] \arrow[dd, bend left=60, "{F[\beta_\Gamma,id]}"] \\
\Sigma(\rho F) \arrow[r, "{\Sigma\langle r,id\rangle}"] \arrow[d, "\beta"] & \Sigma(F(\rho F)\times\rho F) \arrow[r, "\lambda"] & F(\Sigma(\rho F)+\rho F) \arrow[d, "{F[\beta,id]}"'] \\
\rho F \arrow[rr, "r"'] & & F(\rho F)
\end{tikzcd}$$

Now we see that the desired lower square commutes when extended by any colimit injection in_Γ since all other parts and the outside commute. For the uniqueness assume that β is given such that the lower part commutes. Then we see that $\beta \cdot \mathsf{in}_\Gamma = \beta_\Gamma$ by the uniqueness of β_Γ in Theorem 2.11.

Ad (b). The second statement easily follows from the fact that the unique F-coalgebra homomorphism $h : (\rho F, r) \to (\nu F, t)$ is a Γ-algebra homomorphism from $(\rho F, \beta_\Gamma)$ to $(\nu F, \alpha_\Gamma)$ for every $\Gamma \in \mathcal{D}$ (recall the discussion after Theorem 2.11). Indeed, we have the diagram

$$\begin{tikzcd}
\Gamma(\rho F) \arrow[r, "\mathsf{in}_\Gamma"] \arrow[d, "\Gamma h"'] & \Sigma(\rho F) \arrow[r, "\beta"] \arrow[d, "\Sigma h"'] & \rho F \arrow[d, "h"] \\
\Gamma(\nu F) \arrow[r, "\mathsf{in}_\Gamma"'] & \Sigma(\nu F) \arrow[r, "\alpha"'] & \nu F
\end{tikzcd}$$

where the left-hand square commutes by the naturality of in_Γ. So the right-hand square commutes when precomposed with every in_Γ; now use that the colimit injections in_Γ form an epimorphic family. $\square$

Remark 4.9 Bipointed specifications that do not have finite dependency will, in general, not yield a "restriction" of the denotational model $\alpha : \Sigma(\nu F) \to \nu F$ to $\beta : \Sigma(\rho F) \to \rho F$. To see this consider the following example for streams (cf. [14, Example 3.5]), i.e. $FX = \mathbb{R} \times X$ on Set. Let $\Sigma X = \mathbb{N}$ be given by constants c_n, $n \in \mathbb{N}$ and operational rules

$$\frac{}{c_n \overset{n}{\to} c_{n+1}}.$$

This induces a natural transformation

$$\ell_X : \Sigma F X = \mathbb{N} \to \mathbb{R} \times \mathbb{N} = F \Sigma X \quad \text{with} \quad n \mapsto (n, n+1),$$

and we get a bipointed specification $\lambda = (\Sigma(F \times Id) \xrightarrow{\Sigma\pi_0} \Sigma F \xrightarrow{\ell} F\Sigma \xrightarrow{F\mathsf{inl}} F(\Sigma + Id)$. The corresponding operational model $\alpha : \mathbb{N} \to \mathbb{R}^\omega$ interprets the constants c_n as the streams $\alpha(n) = (n, n+1, n+2, \dots)$ which clearly are not eventually periodic.

We are now ready to state the main result of this paper, the generalization of Theorem 4.1 to bipointed specifications.

Theorem 4.10 *Let λ be a bipointed specification having finite dependency. Then the lifted functor $\hat{\Sigma} : \mathsf{Coalg}(F) \to \mathsf{Coalg}(F)$ restricts to $\mathsf{Coalg}_{\mathsf{lfp}}(F)$.*

Proof Let $(\lambda_\Gamma)_{\Gamma \in \mathcal{D}}$ be as in Definition 4.5. For the proof of the theorem we proceed in two steps.

(1) First notice that Σ is a finitary endofunctor being the filtered colimit of the (strongly) finitary functors $\Gamma \in \mathcal{D}$. The coproduct injections $\mathsf{in}_\Gamma : \Gamma \to \Sigma$ extend to monad morphisms

$$\hat{\mathsf{in}}_\Gamma : \hat{\Gamma} \to \hat{\Sigma},$$

which are colimit injections exhibiting the free monad $\hat{\Sigma}$ as a filtered colimit of the free monads $\hat{\Gamma}$. By Theorem 3.3 every $\hat{\Gamma}$ lifts to a functor on $\mathsf{Coalg}(F)$ and so does $\hat{\Sigma}$, by the same argument as in the proof of Theorem 3.3. Since filtered colimits of monads are formed objectwise in $\mathcal{A}$ and since the forgetful functor $\mathsf{Coalg}(F) \to \mathcal{A}$ creates all colimits it follows that the lifting of $\hat{\Sigma}$ is a filtered colimit of the liftings of $\hat{\Gamma}$ to $\mathsf{Coalg}(F)$. Now let (S, f) be an F-coalgebra. Using the constructions of $\hat{\Sigma}S$ and $\hat{\Gamma}S$ (see (4)) one can easily prove by induction on n that for each $\Gamma \in \mathcal{D}$, $(\hat{\mathsf{in}}_\Gamma)_S$ is an F-coalgebra homomorphism from $\hat{\Gamma}(S, f)$ to $\hat{\Sigma}(S, f)$.

(2) Every $\Gamma \in \mathcal{D}$ is strongly finitary. From Theorem 3.3 we then know that $\hat{\Gamma}$ restricts to $\mathsf{Coalg}_{\mathsf{lfp}}(F)$. It then follows that $\hat{\Sigma}$ restricts to $\mathsf{Coalg}_{\mathsf{lfp}}(F)$: for every coalgebra (S, f) in $\mathsf{Coalg}(S, f)$, $\hat{\Sigma}(S, f)$ is a filtered colimit of the coalgebras $\hat{\Gamma}(S, f)$, $\Gamma \in \mathcal{D}$. So since all $\hat{\Gamma}(S, f)$ are in $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ and $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ has filtered colimits we see that $\hat{\Sigma}(S, f)$ lies in $\mathsf{Coalg}_{\mathsf{lfp}}(F)$ as desired. $\square$

In other words, for every locally finitely presentable coalgebra (S, f) the free Σ-algebra $\hat{\Sigma}S$ carries a canonical locally finitely presentable coalgebra. So finally, we obtain the desired generalization of Aceto's theorem. Notice that the following theorem is not just a trivial corollary of Theorem 4.10; as for Theorem 3.4 we still need to prove that the canonical F-coalgebra structure arising on $\mu\Sigma = \hat{\Sigma}0$ coincides with the operational model $c : \mu\Sigma \to F(\mu\Sigma)$.

Theorem 4.11 *Let λ be a bipointed specification having finite dependency. Then the operational model of λ is a locally finitely presentable coalgebra.*

Remark 4.12 Let $\mathsf{in}_\Gamma : \Gamma \to \Sigma$ and $\hat{\mathsf{in}}_\Gamma : \hat{\Gamma} \to \hat{\Sigma}$ denote the colimit injections from the proof of Theorem 4.10. The natural transformation in_Γ induces natural transformations from the free-algebra chain for $\hat{\Gamma}X$ to the one for $\hat{\Sigma}X$, for every object X (see (4)). We only need the case $X = 0$ here; we denote the components of the corresponding natural transformation by $h_n^\Gamma : \Gamma^n 0 \to \Sigma^n 0$. They are defined by $h_0^\Gamma = id_0$ and

$$h_{n+1}^\Gamma = (\Gamma^{n+1}0 = \Gamma(\Gamma^n 0) \xrightarrow{\Gamma h_n^\Gamma} \Gamma(\Sigma^n 0) \xrightarrow{\mathsf{in}_\Gamma} \Sigma(\Sigma^n 0) = \Sigma^{n+1}0).$$

This natural transformation induces the morphism $\hat{h}^\Gamma : \mu\Gamma \to \mu\Sigma$ on the colimits of the chains, i. e., $\hat{h}^\Gamma$ is unique such that $\hat{h}^\Gamma \cdot t_n^\Gamma = t_n \cdot h_n^\Gamma$, where $t_n^\Gamma : \Gamma^n 0 \to \mu\Gamma$ and $t_n : \Sigma^n 0 \to \mu\Sigma$ are the chain colimit injections (cf. the proof of Theorem 3.4).

Proof (Theorem 4.11) Let $\Sigma = \operatorname{colim}_{\Gamma\in\mathcal{D}} \Gamma$ as in Definition 4.5. From Theorem 4.10, we know that the F-coalgebra structure on $\mu\Sigma = \hat{\Sigma}0$ is uniquely induced on this colimit by the coalgebra structures on $\mu\Gamma = \hat{\Gamma}0$ that we have for each $\Gamma \in \mathcal{D}$. From the proof of Theorem 3.4 we know that the latter coalgebra structures are the operational models $c_\Gamma : \mu\Gamma \to F(\mu\Gamma)$ of λ_Γ (see (12)). So all we need to prove is that the following squares commute:

$$\begin{tikzcd} \mu\Gamma \arrow[r, "c_\Gamma"] \arrow[d, "\hat{h}^\Gamma"'] & F(\mu\Gamma) \arrow[d, "F\hat{h}^\Gamma"] \\ \mu\Sigma \arrow[r, "c"'] & F(\mu\Sigma) \end{tikzcd} \qquad \text{for each } \Gamma \in \mathcal{D}.$$

For this we will use that the coalgebra $(\mu\Gamma, c_\Gamma)$ is a colimit of the chain of coalgebras $(\Gamma^n 0, f_n^\Gamma)$ and that $(\mu\Sigma, c)$ is a colimit of the chain of coalgebras given by $(\Sigma^n 0, f_n)$ (cf. the proof of Theorem 3.4). So we consider the following square:

$$\begin{tikzcd} (\Gamma^n 0, f_n^\Gamma) \arrow[r, "t_n^\Gamma"] \arrow[d, "h_n^\Gamma"'] & (\mu\Gamma, c_\Gamma) \arrow[d, "\hat{h}^\Gamma"] \\ (\Sigma^n 0, f_n) \arrow[r, "t_n"'] & (\mu\Sigma, c) \end{tikzcd}$$

On the levels of the carriers of the displayed F-coalgebras the square commutes. So in order to prove that the right-hand arrow is an F-coalgebra homomorphism as indicated it suffices to show that the composite $\hat{h}^\Gamma \cdot t_n^\Gamma = t_n \cdot h_n^\Gamma$ is one (since we already know that t_n^Γ also is one and the t_n^Γ, $n < \omega$, form a jointly epimorphic family). Furthermore, because we know that t_n is a F-coalgebra homomorphism it only remains to prove that h_n^Γ is one. This is done by induction on n. The base case $n = 0$ is trivial. For the induction step we consider the following diagram:

$$\begin{tikzcd}
\Gamma^{n+1}0 \arrow[rr, "h^\Gamma_{n+1}"] \arrow[d, equal] & & \Sigma^{n+1}0 \arrow[d, equal] \\
\Gamma(\Gamma^n 0) \arrow[r, "\Gamma h^\Gamma_n"] \arrow[d, "{\Gamma\langle f^\Gamma_n, id\rangle}"'] & \Gamma(\Sigma^n 0) \arrow[r, "\mathsf{in}_\Gamma"] \arrow[d, "{\Gamma\langle f_n, id\rangle}"] & \Sigma(\Sigma^n 0) \arrow[d, "{\Sigma\langle f_n, id\rangle}"'] \\
\Gamma(F(\Gamma^n 0) \times \Gamma^n 0) \arrow[r, "\Gamma(F(h^\Gamma_n)\times h^\Gamma_n)"] \arrow[d, "\lambda_\Gamma"'] & \Gamma(F(\Sigma^n 0) \times \Sigma^n 0) \arrow[r, "\mathsf{in}_\Gamma"] \arrow[d, "\lambda_\Gamma"'] & \Sigma(F(\Sigma^n 0) \times \Sigma^n 0) \arrow[d, "\lambda"'] \\
F(\Gamma(\Gamma^n 0) + \Gamma^n 0) \arrow[r, "F(\Gamma h^\Gamma_n + h^\Gamma_n)"] \arrow[d, "{F[id, f^\Gamma_{n,n+1}]}"'] & F(\Gamma(\Sigma^n 0) + \Sigma^n 0) \arrow[r, "F(\mathsf{in}_\Gamma + id)"] & F(\Sigma(\Sigma^n 0) + \Sigma^n 0) \arrow[d, "{F[id, f_{n,n+1}]}"'] \\
F(\Gamma^{n+1}0) \arrow[rr, "F h^\Gamma_{n+1}"'] & & F(\Sigma^{n+1}0)
\end{tikzcd}$$

This diagram commutes: the upper part commutes by the definition of h^Γ_{n+1}, and we next consider the four small inner squares. For the commutativity of the upper left-hand square remove Γ and consider the product components separately: the left-hand one commutes using the induction hypothesis and the right-hand one is trivial. The upper right-hand and lower left-hand and square commute by the naturality of in_Γ and λ_Γ, respectively, and the lower right-hand square commutes by (12). Finally, to see that the lowest part commutes, remove F and consider the coproduct components separately: the left-hand component commutes by the definition of h^Γ_{n+1} again, and the right-hand one by naturality of h^Γ (recall that this is a natural transformation from the chain $(\Gamma^n 0)_{n<\omega}$ to the chain $(\Sigma^n 0)_{n<\omega}$). So since the left-hand and right-hand edges of the diagram above are the coalgebras $(\Gamma^{n+1}0, f^\Gamma_{n+1})$ and $(\Sigma^{n+1}0, f_{n+1})$ we are done. ☐

Remark 4.13 We chose to present all our results for bipointed specifications because in applications it is easier to find concrete rule formats corresponding to them. But we believe that all of our results can be proved more generally for so-called *coGSOS laws* $\Sigma\bar{F} \to F(\Sigma + Id)$, where $\bar{F}$ denotes the cofree comonad on F (see Klin [20, Section 6.4]).

5 A rule format for operations on regular languages

In [14] there are a number of examples of concrete formats and operations corresponding to bipointed specifications. All of these examples are on Set. However, for example in the case of deterministic automata, bipointed specifications on Set are rather limited; standard operations like concatenation, Kleene star or the shuffle product of languages cannot be specified by bipointed specifications for $FX = 2 \times X^A$ on Set.

Moving from Set to the category Jsl, bipointed specifications allows for different and more powerful specification formats. Recall that the functor $F = 2 \times \mathrm{Id}^A$ lifts to the functor $\bar{F} = 2 \times \mathrm{Id}^A$ on Jsl, the category of join semilattices, where $2 = \{0, 1\}$ is the join semilattice where 0 is bottom and the join is the usual "or" operation on bits. Recall from Example 2.9(2) that the rational fixpoint of $\bar{F}$ is carried by the set of regular languages as well. In this section we exploit this fact to derive a concrete format for operations on regular languages from bipointed specifications for $\bar{F}$. This format is more expressive than bipointed specifications for F, as the join semilattice structure allows to express non-determinism in the conclusion of rules.

Before we present a concrete rule format we will analyze (certain) bipointed specifications for $\bar{F}$. In the sequel let $U : \mathsf{Jsl} \mathrel{\substack{\longleftarrow \\ \perp \\ \longrightarrow}} \mathsf{Set} : \Phi$ denote the free and forgetful functor, respectively. We also denote by $J : \mathsf{FJsl} \to \mathsf{Jsl}$ the inclusion of the full subcategory given by free join semilattices. We are interested in functors $\Sigma : \mathsf{Jsl} \to \mathsf{Jsl}$ of the form $\Phi P_\Gamma U$, where $P_\Gamma : \mathsf{Set} \to \mathsf{Set}$ is a polynomial functor associated to the signature Γ. The reason for this is that Σ-algebras are precisely join semilattices A equipped with a function of the type $P_\Gamma UA \to UA$, i. e., for every operation symbol $\gamma \in \Gamma$ a (not necessarily join preserving) operation $A^{ar(\gamma)} \to A$.

Lemma 5.1 *Families of natural transformations*

$$\hat{\gamma} : (FUJ \times UJ)^{ar(\gamma)} \Rightarrow FU(\Sigma J + J) \qquad \gamma \in \Gamma \tag{13}$$

are in one-to-one correspondence with bipointed specifications of $\Sigma = \Phi P_\Gamma U$ over the functor $\bar{F}$.

The proof of the lemma makes use of the fact that any variety is the completion of its subcategory of free finitely generated algebras under sifted colimits. We recall the necessary notions and prove a technical lemma.

First recall (e. g. from Adámek and Rosicky [8]) that a colimit of a diagram with domain (or diagram scheme $\mathcal{D}$) is called *sifted* if $\mathcal{D}$-colimits commute with all finite products in Set. For example, filtered colimits and reflexive coequalizers are sifted colimits.

Let $\mathcal{V}$ be a finitary variety. Recall from Example 2.2(2) that the finitely presentable objects $A \in \mathcal{V}$ are precisely those algebras presented by finitely many generators and finitely many relations. For example, in Jsl they are precisely the finite algebras because the finitely generated free algebras Φn are finite. One can define a notion of *strongly finitely presentable* object $A \in \mathcal{V}$; these are precisely the retracts of finitely generated free algebras [9]. In Set and $\mathsf{Vect}(\mathbb{F})$ they coincide with the finitely presentable objects. But this fails in Jsl: for example, the 3 element chain is a retract of the four element algebra $\Phi 2$. Let $\mathcal{V}_{sfp}$ be the full subcategory of $\mathcal{V}$ given by strongly finitely presentable objects.

Remark 5.2 Note that sifted colimit preserving functors between varieties are equivalently (1) finitary functors which preserve reflexive coequalisers (see [8]), or (2) equationally presentable functors i.e. those with a presentation by rank-1 equations [21, Theorem 4.9].

Lemma 5.3 *Let $\mathcal{V}$, $\mathcal{W}$ be finitary varieties and $\mathcal{V}_0$ be the full subcategory of $\mathcal{V}$ given by finitely generated free algebras with the inclusion functor $J : \mathcal{V}_0 \hookrightarrow \mathcal{V}$. Then the* *functor category* $[\mathcal{V}_0, \mathcal{W}]$ *is equivalent to the category of sifted colimit preserving functors from* $\mathcal{V}$ *to* $\mathcal{W}$*. More precisely, we have an equivalence of categories*

$$[\mathcal{V}, \mathcal{W}]_{sift} \cong [\mathcal{V}_0, \mathcal{W}]$$

given by restricting by composition with J*; in symbols,* $G \mapsto GJ$*.*

Proof It is known that $[\mathcal{V}_{sfp}, \mathcal{W}] \cong [\mathcal{V}, \mathcal{W}]_{sift}$ for any finitary variety $\mathcal{V}$ and $\mathcal{W}$ (see [8, Definition 2.2 and Theorem 3.10]).

Now it suffices to show that $[\mathcal{V}_0, \mathcal{W}] \cong [\mathcal{V}_{sfp}, \mathcal{W}]$. This follows from the fact that $\mathcal{V}_{sfp}$ is equivalent to the Cauchy-completion of $\mathcal{V}_0$. To see this, note that $\mathcal{V}$ is cocomplete and $\mathcal{V}_0$ is a full subcategory of it. In this case the Cauchy completion is equivalent to the closure of $\mathcal{V}_0$ under retracts in $\mathcal{V}$, viz. $\mathcal{V}_{sfp}$. The desired equivalence now follows from the universal property of the Cauchy completion. ☐

Proof (Lemma 5.1) Let Γ be a signature with associated polynomial set functor $P_\Gamma : \mathsf{Set} \to \mathsf{Set}$, let $\Sigma = \Phi P_\Gamma U : \mathsf{Jsl} \to \mathsf{Jsl}$. A bipointed specification in this case is a natural transformation

$$\lambda : \Phi P_\Gamma U(\bar{F} \times \mathrm{Id}) \Rightarrow \bar{F}(\Phi P_\Gamma U + \mathrm{Id}).$$

Notice that its components are Jsl homomorphisms. Since Φ preserves colimits and U preserves limits, such bipointed specifications are in one-to-one correspondence with families

$$\Phi((U\bar{F} \times U)^{ar(\gamma)}) \Rightarrow \bar{F}(\Phi P_\Gamma U + \mathrm{Id}) \qquad (\gamma \in \Gamma)$$

of natural transformations. These are, by virtue of the adjunction $\Phi \dashv U$ and the fact that $\bar{F}$ lifts F, in one-to-one correspondence to families of natural transformations

$$(FU \times U)^{ar(\gamma)} \Rightarrow FU(\Phi P_\Gamma U + \mathrm{Id}) \qquad (\gamma \in \Gamma) \tag{14}$$

whose components are just functions.

Such natural transformations are in fact uniquely determined by their components at free algebras, i.e., they are equivalently given by families as in (13). To see this notice that the two functors $(FU \times U)^{ar(\gamma)}$ and $FU(\Phi P_\Gamma U + \mathrm{Id})$ are finitary and clearly preserve reflexive coequalizers. This implies that they preserve sifted colimits (see [8]). The desired result is now an application of the Lemma 5.3 to $\mathcal{V} = \mathcal{W} = \mathsf{Jsl}$ with $\mathcal{V}_0$ being the finite free algebras. ☐

We proceed to move from free join semilattices to plain sets and consider natural transformations

$$\bar{\gamma} : (F \times \mathrm{Id})^{ar(\gamma)} \Rightarrow FU\Phi(P_\Gamma U\Phi + \mathrm{Id}) \qquad \gamma \in \Gamma \tag{15}$$

Such families of natural transformations induce bipointed specifications, but the converse does not hold.

Lemma 5.4 *Every* $\bar{\gamma}$ *as in (15) induces a* $\hat{\gamma}$ *as in (13), and consequently such a collection induces a bipointed specification.*

Proof Let $\bar{\gamma} : (F \times \mathrm{Id})^{ar(\gamma)} \Rightarrow FU\Phi(P_\Gamma U\Phi + \mathrm{Id})$ be a natural transformation. Let $J : \mathsf{FJsl} \to \mathsf{Jsl}$ again be the inclusion of free join semilattices. Then $\bar{\gamma}$ induces a natural transformation $\bar{\gamma}UJ : (FUJ \times UJ)^{ar(\gamma)} \Rightarrow FU\Phi(P_\Gamma U\Phi UJ + UJ)$ simply by instantiating it to the carriers of the free join semilattices. Since Φ preserves coproducts this is equivalent to a natural transformation

$$\bar{\gamma}'UJ : (FUJ \times UJ)^{ar(\gamma)} \Rightarrow FU(\Phi P_\Gamma U\Phi UJ + \Phi UJ)\,.$$

By composing the counit $\epsilon : \Phi U \to Id$ of the adjunction $\Phi \dashv U : \mathsf{Jsl} \to \mathsf{Set}$ we obtain a natural transformation

$$\hat{\gamma} \overset{\mathsf{def}}{=} FU(\Phi P_\Gamma U\epsilon J + \epsilon J) \cdot \bar{\gamma}'UJ : (FUJ \times UJ)^{ar(\gamma)} \Rightarrow FU(\Phi P_\Gamma UJ + J)$$

which is of type (13) as desired. □

Remark 5.5 The above treatment of bipointed specifications on Jsl does not depend on the specific properties of join semilattices, but works similarly for any locally finite variety.

We are now ready to define a concrete syntactic rule format, inducing the above families of natural transformations $\hat{\gamma}$.

5.1 A concrete format for deterministic automata on Jsl. In the remainder of this section let Σ be a finitary signature. A *transition rule* and an *output rule* are of the form

$$\frac{\{x_i{\downarrow}\}_{i\in I} \qquad \{x_i{\uparrow}\}_{i\in J}}{\sigma(x_1,\ldots,x_n) \xrightarrow{a} t} \qquad \text{and} \qquad \frac{\{x_i{\downarrow}\}_{i\in I} \qquad \{x_i{\uparrow}\}_{i\in J}}{\sigma(x_1,\ldots,x_n){\downarrow}}$$

respectively, where $x_1,\ldots,x_n$ is a collection of pairwise distinct variables, σ an n-ary operator of Σ; further $I, J \subseteq \{1,2,\ldots,n\}$ and t is a term over the grammar

$$t ::= \bot \mid t \oplus t \mid \tau(u_1,\ldots,u_{ar(\tau)}) \mid x \qquad\qquad u ::= \bot \mid u \oplus u \mid x, \tag{16}$$

where τ ranges over the operators of Σ, x ranges over the least collection of variables V such that $x_i \in V$ for all i, and for each alphabet letter $a \in A$ and index $i \leq n$ there is a distinct variable $x_i^a \in V$. Intuitively, $x_i{\uparrow}$ and $x_i{\downarrow}$ represent states that must be non-final and final, respectively, and x_i^a represents the unique state reached by x_i after an a-transition[6]. A *(bipointed) DFA (SOS) specification* is a set of transition rules and output rules such that for every operator σ of Σ, every alphabet symbol $a \in A$ and all possible sets of premises $\{x_i{\downarrow}\}_{i\in I}$ and $\{x_i{\uparrow}\}_{i\in J}$ only finitely many rules with a conclusion of the form $\sigma(x_1,\ldots,x_n) \xrightarrow{a} t$ exist. (Notice that this finiteness property corresponds to boundedness of GSOS specifications.)

Operator dependency on Σ and finite dependency of a DFA specification is defined in exactly the same way as for GSOS specifications (see Section 4.1).

---

[6] In analogy with standard SOS we will denote x_i^a by a variable y by writing a transition $x_i \xrightarrow{a} y$ in the premise of the rule.

Proposition 5.6 *Any DFA specification (having finite dependency) induces a bipointed specification (having finite dependency).*

Proof (1) First we see that every DFA SOS specification corresponds precisely to a family of functions

$$f_\sigma : 2^{ar(\sigma)} \to 2 \times \mathcal{L}^A,$$

where $\mathcal{L}$ is the set of terms defined by the grammar in (16) on the set of variables $V = \{x_i, x_i^a \mid a \in A, i = 1, \dots, ar(\sigma)\}$. Indeed, given $\sigma \in \Sigma$ define $\pi_1 \cdot f_\sigma(s) = 1$ iff there is an output rule for σ with

$$s(i) = 1 \iff i \in I \qquad \text{and} \qquad s(i) = 0 \iff i \in J. \tag{17}$$

And we define

$$\pi_2 \cdot f_\sigma(s)(a) = \bigoplus \{t \mid \sigma(x_1, \dots, x_n) \xrightarrow{a} t \text{ conclusion of a transition rule with (17)}\}.$$

This is a well-defined term in $\mathcal{L}$ since the join above is formed over a finite set because we assume that there are only finitely many rules with conclusion $\sigma(x_1, \dots x_n) \xrightarrow{a} t$.

Next observe that $\mathcal{L}$ is precisely the set $\Phi(P_\Sigma U \Phi V + V)$. So the f_σ form a family of functions

$$f_\sigma : 2^{ar(\sigma)} \to FU\Phi(P_\Sigma U \Phi V + V).$$

Now for every $\sigma \in \Sigma$ the function f_σ induces a natural transformation

$$\mathsf{Set}(V, -) \to FU\Phi(P_\Sigma U \Phi + \mathrm{Id})^{2^{ar(\sigma)}} \tag{18}$$

by an application of the Yoneda Lemma. Finally, observe that the set V of variables is isomorphic to $ar(\sigma) \times (A + 1)$. Thus, $\mathsf{Set}(V, -)$ is (isomorphic to) the functor $(\mathrm{Id}^A \times \mathrm{Id})^{ar(\sigma)}$. Then the natural transformation in (18) corresponds precisely to a natural transformation

$$(2 \times \mathrm{Id}^A \times \mathrm{Id})^{ar(\sigma)} \to FU\Phi(P_\Sigma U \Phi + \mathrm{Id}),$$

i.e. a natural transformation as in (15). So we obtain a bipointed specification λ according to Lemma 5.4.

(2) It remains to show that finite dependency of the given DFA specification entails finite dependency of the induced bipointed specification. As in the proof of Proposition 4.3 we see that Σ is the directed union of all its closed subsignatures Γ, which are all finite. For a closed subsignature Γ of Σ we see that each f_σ, $\sigma \in \Gamma$, above restricts to

$$f_\sigma : 2^{ar(\sigma)} \to 2 \times {\mathcal{L}_\Gamma}^A,$$

where $\mathcal{L}_\Gamma$ is the subset of $\mathcal{L}$ given by the terms using only operators from Γ. Now it is not difficult to prove that by following the same steps as in point (1) we get bipointed specifications λ_Γ for each closed subsignature Γ of Σ whose directed union is λ as obtained in (1); this shows that λ has finite dependency (see Definition 4.5). We leave the details to the reader. □

Thus by Corollary 4.8 the rational fixpoint, i.e., the set of regular languages, is closed under any operations defined by a DFA specification having finite dependency. And by Theorem 4.11 the operational model is locally finite. We proceed to show several examples.

Given two words w and v, the *shuffle* of w and v, denoted $w \bowtie v$, is the set of words obtained by arbitrary interleavings of w and v [30]. For example, $ab \bowtie c = \{abc, acb, cab\}$. The shuffle of two languages L_1 and L_2 is the pointwise extension: $L_1 \bowtie L_2 = \bigcup_{w \in L_1, v \in L_2} w \bowtie v$. The shuffle operator can be defined in terms of a DFA specification as follows:

$$\frac{x \xrightarrow{a} x'}{x \bowtie y \xrightarrow{a} x' \bowtie y} \qquad \frac{y \xrightarrow{a} y'}{x \bowtie y \xrightarrow{a} x \bowtie y'} \qquad \frac{x{\downarrow} \quad y{\downarrow}}{(x \bowtie y){\downarrow}}$$

By Corollary 4.8 the set of regular languages is closed under shuffle.

Concatenation, Kleene star, a single alphabet letter and the neutral element $1 = \{\varepsilon\}$ w.r.t. concatenation, are defined as follows (the Kleene star is defined using an additional binary operation f, such that intuitively $f(L_1, L_2) = L_1 \cdot L_2^*$):

$$\frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y} \qquad \frac{x{\downarrow} \qquad y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'} \qquad \frac{x{\downarrow} \qquad y{\downarrow}}{(x \cdot y){\downarrow}} \qquad\qquad \frac{}{a \xrightarrow{a} 1} \quad a \in A$$

$$\frac{x \xrightarrow{a} x'}{f(x,y) \xrightarrow{a} f(x',y)} \qquad \frac{x{\downarrow} \qquad y \xrightarrow{a} y'}{f(x,y) \xrightarrow{a} f(y',y)} \qquad \frac{x{\downarrow}}{f(x,y){\downarrow}} \qquad\qquad \frac{}{1{\downarrow}}$$

For the corresponding signature Γ the functor $\Sigma = FP_\Gamma U$ on Jsl thus represents syntactically the above operations, in addition to the join semilattices operations. Thus the initial algebra of Σ consists of *regular expressions* (with a binary Kleene star) modulo the join semilattice equations. So the *operational model* is precisely the coalgebra of regular expressions; by Theorem 4.11 this is locally finite. As such, we obtain for free that the number of derivatives of a regular expression is finite modulo the join semilattice equations (cf. [16]).

Interestingly, Proposition 5.6 works for any DFA specification having finite dependency, thus also when considering an infinite signature. Consider, for example, the (obviously infinite) signature containing all regular languages $L \subseteq A^*$ as constant, together with the following DFA specification:

$$\frac{}{L \xrightarrow{a} L_a} \qquad\qquad \frac{}{L{\downarrow}} \quad \text{if } \varepsilon \in L$$

where L_a is the a-derivative of L given by $\{w \mid aw \in L\}$. Because every regular languages has finitely many different derivatives [16], the above DFA specification has finite dependency, and thus by Theorem 4.11 the operational model is locally finite (it coincides, in fact, with the rational fixpoint, with, as carrier, the set of all regular languages).

6 Conclusions and future work

We have generalized Aceto's theorem on the regularity of the operational model of a transition system specification from process algebra to the realm of mathematical operational semantics of Turi and Plotkin. In previous work [14] it was already shown that bipointed specifications for $F = \mathcal{P}_\mathsf{f}(A \times Id)$ generalize Aceto's simple GSOS format, and it was proved that for general bipointed specifications of a strongly finitary functor Σ over a finitary one F a canonical Σ-algebra structure is induced on the rational fixpoint of F "restricting" the denotational model on the final coalgebra for F. Here we have extended this result to finitary functors Σ that are not necessarily strongly finitary. The key to our extension is an abstract formulation of the notion of finite dependency for bipointed specifications that captures Aceto's more concrete notion for simple GSOS specifications as a special instance. This then allows us to prove our generalisation of Aceto's result in Theorem 4.11: the operational model of such a specification is a locally finitely presentable coalgebra. The latter property is interesting for possible tool development, as in any locally finite variety it implies decidability of bisimilarity: there are only finitely many states to check. Moreover, recent results on up to context techniques [26] may lead to a generic and efficient construction of a bisimulation witness of the desired equivalence.

Our second contribution is the new rule format of DFA specifications for operations on formal languages. These specifications are obtained by instantiating bipointed specifications for functors of the form $\Sigma = \Phi P_\Gamma U$ on the category of join semilattices. From our results we then conclude that regular languages are closed under operations specified by DFA specification, and as a corollary we also obtain the well-known result that regular expressions have only finitely many derivatives modulo the axioms of join semilattices.

Many interesting directions are still to be explored. The process described in Section 5.1 can easily be adapted to other locally finite varieties, allowing to derive more expressive concrete formats based on adding equations. In order to treat rational power series and even context-free ones, one needs to move to other algebraic categories, such as vector spaces and idempotent semirings. Furthermore, we plan to investigate the extension of bipointed specification to coGSOS laws [20] to allow arbitrary lookahead in premises of rules.

References

[1] Aceto, L., *GSOS and finite labelled transition systems*, Theoret. Comput. Sci. **131** (1994), pp. 181–195.

[2] Aceto, L., W. Fokkink and C. Verhoef, *Structural operational semantics*, in: *Handbook of Process Algebra*, Elsevier Science, 2001 pp. 197–292.

[3] Adámek, J., *Free algebras and automata realizations in the language of categories*, Comment. Math. Univ. Carolin. **15** (1974), pp. 589–602.

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

[5] Adámek, J., S. Milius and J. Velebil, *Iterative algebras at work*, Math. Structures Comput. Sci. **16** (2006), pp. 1085–1131.

[6] Adámek, J. and H.-E. Porst, *On tree coalgebras and coalgebras presentations*, Theoret. Comput. Sci. **311** (2004), pp. 257–283.

[7] Adámek, J. and J. Rosický, "Locally presentable and accessible categories," Cambridge Univ. Press, 1994.

[8] Adámek, J. and J. Rosický, *On sifted colimits and generalized varieties*, Theory Appl. Categ. **8** (2001), pp. 33–53.

[9] Adámek, J., J. Rosický and E. Vitale, "Algebraic Theories," Cambridge Univ. Press, 2011.

[10] Adámek, J. and V. Trnková, "Automata and Algebras in Categories," Mathematics and its Applications **37**, Kluwer Academic Publishers, 1990.

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

[12] Bartels, F., "On generalised coinduction and probabilistic specification formats," Ph.D. thesis, CWI, Amsterdam (2004).

[13] Bloom, B., S. Istrail and A. Meyer, *Bisimulation can't be traced*, J. ACM **42** (1995), pp. 232–268.

[14] Bonsangue, M., S. Milius and J. Rot, *On the specification of operations on the rational behaviour of systems*, in: *EXPRESS/SOS'12*, Elect. Proc. of Theoret. Comput. Sci. **89**, 2012, pp. 3–18.

[15] Bonsangue, M., S. Milius and A. Silva, *Sound and complete axiomatizations of coalgebraic language equivalence*, ACM Trans. Comput. Log. **14** (2013).

[16] Brzozowski, J. A., *Derivatives of regular expressions*, J. ACM **11** (1964), pp. 481–494.

[17] Gabriel, P. and F. Ulmer, "Lokal präsentierbare Kategorien," Lecture Notes Math. **221**, Springer, 1971.

[18] Johnstone, P., "Stone Spaces," Cambridge University Press, 1986.

[19] Klin, B., *Structural operational semantics for weighted transition systems*, in: *Semantics and Algebraic Specification*, LNCS **5700** (2009).

[20] Klin, B., *Bialgebras for structural operational semantics: An introduction*, Theoret. Comput. Sci. **412** (2011), pp. 5043–5069.

[21] Kurz, A. and J. Rosický, *Strongly complete logic for coalgebras*, Log. Meth. in Comput. Sci. **8** (2012), pp. 3–14.

[22] Lenisa, M., A. J. Power and H. Watanabe, *Category theory for operational semantics*, Theoret. Comput. Sci. **327** (2004).

[23] Makkai, M. and R. Paré, "Accessible categories: the foundation of categorical model theory," Contemporary Math. **104**, Amer. Math. Soc., Providence, RI, 1989.

[24] Milius, S., *A sound and complete calculus for finite stream circuits*, in: *Proc. of LICS 2010* (2010), pp. 421–430.

[25] Milner, R., "Communication and Concurrency," Prentice Hall, 1989.

[26] Rot, J. Bonsangue, M., Rutten, J., *Coalgebraic Bisimulation-Up-To*, in Proc. of SOFSEM 2013, Lecture Notes Comput. Sci. 7741 , Springer (2013) pp. 369–381.

[27] Rutten, J., *Universal coalgebra: a theory of systems*, Theoret. Comput. Sci. **249** (2000).

[28] Rutten, J., *A coinductive calculus of streams*, Math. Structures Comput. Sci. **15** (2005), pp. 93–147.

[29] Rutten, J., *Rational streams coalgebraically*, Log. Meth. in Comput. Sci. **4** (2008), pp. 1–22.

[30] Shallit, J., "A Second Course in Formal Languages and Automata Theory," Cambridge Univ. Press, 2008.

[31] Turi, D. and G. Plotkin, *Towards a mathematical operational semantics*, in: *Proc. of LICS 1997* (1997), pp. 280–291.