A New Foundation for Finitary Corecursion

The Locally Finite Fixpoint and its Properties

Stefan Milius[1,⋆], Dirk Pattinson[2], and Thorsten Wißmann[1,⋆]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg
[2] The Australian National University

Abstract. This paper contributes to a theory of the behaviour of "finite-state" systems that is generic in the system type. We propose that such systems are modeled as coalgebras with a finitely generated carrier for an endofunctor on a locally finitely presentable category. Their behaviour gives rise to a new fixpoint of the coalgebraic type functor called *locally finite fixpoint* (LFF). We prove that if the given endofunctor preserves monomorphisms then the LFF always exists and is a subcoalgebra of the final coalgebra (unlike the rational fixpoint previously studied by Adámek, Milius and Velebil). Moreover, we show that the LFF is characterized by two universal properties: 1. as the final locally finitely generated coalgebra, and 2. as the initial fg-iterative algebra. As instances of the LFF we first obtain the known instances of the rational fixpoint, e.g. regular languages, rational streams and formal power-series, regular trees etc. And we obtain a number of new examples, e.g. (realtime deterministic resp. non-deterministic) context-free languages, constructively S-algebraic formal power-series (and any other instance of the generalized powerset construction by Silva, Bonchi, Bonsangue, and Rutten) and the monad of Courcelle's algebraic trees.

1 Introduction

Coalgebras capture many types of state based system within a uniform and mathematically rich framework [39]. One outstanding feature of the general theory is *final semantics* which gives a fully abstract account of system behaviour. For example, coalgebraic modelling of deterministic automata (without a finiteness restriction on state sets) yields the set of all formal languages as a final model, and restricting to *finite* automata one precisely obtains the regular languages [38]. This correspondence has been generalized to locally finitely presentable categories [8, 20], where *finitely presentable* objects play the role of finite sets, leading to the notion of *rational fixpoint* that provides final semantics to all models with finitely presentable carrier [31]. It is known that the rational fixpoint is fully abstract (identifies all behaviourally equivalent states) as long as finitely presentable objects agree with finitely generated objects in the base category [12, Proposition 3.12]. While this is the case in some categories (e.g. sets, posets, graphs, vector spaces, commutative monoids), it is currently unknown in other base categories that are used in the construction of system models, for example in idempotent semirings (used in the treatment of context-free grammars [43]), in algebras for the stack monad

⋆ Supported by Deutsche Forschungsgemeinschaft (DFG) under project MI 717/5-1

(used for modelling configurations of stack machines [23]); or it even fails, for example in the category of finitary monads on sets (used in the categorical study of algebraic trees [7]), or Eilenberg-Moore categories for a monad in general (the target category of generalized determinization [41], in which the above examples live). Coalgebras over a category of Eilenberg-Moore algebras over Set in particular provide a paradigmatic setting: automata that describe languages beyond the regular languages consist of a finite state set, but their transitions produce side effects such as the manipulation of a stack. These can be described by a monad, so that the (infinite) set of system states (machine states plus stack content) is described by a free algebra (for that monad) that is generated by the finite set of machine states. This is formalized by the generalized powerset construction [41] and interacts nicely with the coalgebraic framework we present.

Technically, the shortcoming of the rational fixpoint is due to the fact that finitely presentable objects are not closed under quotients, so that the rational fixpoint itself may fail to be a subcoalgebra of the final coalgebra and so identifies too little behaviour. The main conceptual contribution of this paper is the insight that also in cases where finitely presentable and finitely generated do not agree, the *locally finite fixpoint* provides a fully abstract model of finitely generated behaviour. We give a construction of the locally finite fixpoint, and support our claim both by general results and concrete examples: we show that under mild assumptions, the locally finite fixpoint always exists, and is indeed a subcoalgebra of the final coalgebra. Moreover, we give a characterization of the locally finite fixpoint as the initial iterative algebra. We then instantiate our results to several scenarios studied in the literature.

First, we show that the locally finite fixpoint is universal (and fully abstract) for the class of systems produced by the generalized powerset construction over Set: every determinized finite-state system induces a unique homomorphism to the locally finite fixpoint, and the latter contains precisely the finite-state behaviours.

Applied to the coalgebraic treatment of context-free languages, we show that the locally finite fixpoint yields precisely the context-free languages, and real-time deterministic context-free languages, respectively, when modelled using algebras for the stack monad of [23]. For context-free languages weighted in a semiring S, or equivalently for constructively S-algebraic power series [36], the locally finite fixpoint comprises precisely those, by phrasing the results of Winter et al. [44] in terms of the generalized powerset construction. Our last example shows the applicability of our results beyond categories of Eilenberg-Moore algebras over Set, and we characterize the monad of Courcelle's algebraic trees over a signature [16, 7] as the locally finite fixpoint of an associated functor (on a category of monads), solving an open problem of [7].

The work presented here is based on the third author's master thesis in [45]. Most proofs are omitted; they can be found in the appendix.

2 Preliminaries and Notation

Locally finitely presentable categories. A *filtered colimit* is the colimit of a diagram $\mathcal{D} \to \mathcal{C}$ where $\mathcal{D}$ is filtered (every finite subdiagram has a cocone in $\mathcal{D}$) and *directed* if $\mathcal{D}$ is additionally a poset. *Finitary functors* preserve filtered (equivalently directed) colimits. Objects $C \in \mathcal{C}$ are *finitely presentable* (fp) if the hom-functor $\mathcal{C}(C, -)$ preserves filtered (equivalently directed) colimits, and *finitely generated* (fg) if $\mathcal{C}(C,-)$ preserves directed colimits of monos (i.e. colimits of directed diagrams where all connecting morphisms are monic). Clearly any fp object is fg, but not vice versa. Also, fg objects are closed under strong epis (quotients) which fails for fp objects in general. A cocomplete category is *locally finitely presentable* (lfp) if the full subcategory $\mathcal{C}_{\mathsf{fp}}$ of finitely presentable objects is essentially small, i.e. is up to isomorphism only a set, and every object $C \in \mathcal{C}$ is a filtered colimit of a diagram in $\mathcal{C}_{\mathsf{fp}}$. We refer to [20, 8] for further details.

It is well known that the categories of sets, posets and graphs are lfp with finitely presentable objects precisely the finite sets, posets, graphs, respectively. The category of vector spaces is lfp with finite-dimensional spaces being fp. Every finitary variety is lfp (i.e. an equational class of algebras induced by finite-arity operations or equivalently the Eilenberg-Moore category for a finitary Set-Monad, see Section 4.1 later). The finitely generated objects are the finitely generated algebras, and finitely presentable objects are algebras specified by finitely many generators and relations. This includes the categories of groups, monoids, (idempotent) semirings, semi-modules, etc. Every lfp category has mono/strong epi factorization [8, Proposition 1.16], i.e. every f factors as $f = m \cdot e$ with m mono (denoted by $\rightarrowtail$), e strong epi (denoted by $\twoheadrightarrow$), and we call the domain $\mathsf{Im}(f)$ of e the *image* of f. Any strong epi e has the diagonal fill-in property, i.e. $m \cdot g = h \cdot e$ with m mono and e strong epi gives a unique d such that $m \cdot d = h$ and $g = d \cdot e$.

Coalgebras. If $H : \mathcal{C} \to \mathcal{C}$ is an endofunctor, *H-coalgebras* are pairs (C, c) with $c : C \to HC$, and C is the *carrier* of (C, c). Homomorphisms $f : (C, c) \to (D, d)$ are maps $f : C \to D$ such that $Hf \cdot c = d \cdot f$. This gives a category denoted by $\mathsf{Coalg}H$. If its final object exists then this final H-coalgebra $(\nu H, \tau)$ represents a canonical domain of behaviours of H-typed systems, and induces for each (C, c) a unique homomorphism, denoted by $c^\dagger$, giving semantics to the system (C, c). The final coalgebra always exists provided $\mathcal{C}$ is lfp and H is finitary. The forgetful functor $\mathsf{Coalg}H \to \mathcal{C}$ creates colimits and reflects monos and epis. A morphism f in $\mathsf{Coalg}H$ is *mono-carried* (resp. *epi-carried*) if the underlying morphism in $\mathcal{C}$ is monic (resp. epic). Strong epi/mono factorizations lift from $\mathcal{C}$ to $\mathsf{Coalg}H$ whenever H preserves monos yielding epi-carried/mono-carried factorizations. A *directed union of coalgebras* is the colimit of a directed diagram in $\mathsf{Coalg}H$ where all connecting morphisms are mono-carried.

The Rational Fixpoint. For $\mathcal{C}$ lfp and $H : \mathcal{C} \to \mathcal{C}$ finitary let $\mathsf{Coalg}_{\mathsf{fp}}H$ denote the full subcategory of $\mathsf{Coalg}H$ of coalgebras with fp carrier, and $\mathsf{Coalg}_{\mathsf{lfp}}H$ the full subcategory of $\mathsf{Coalg}H$ of coalgebras that arise as filtered colimits of coalgebras with fp carrier [31, Corollary III.13]. The coalgebras in $\mathsf{Coalg}_{\mathsf{lfp}}H$ are called *lfp coalgebras* and for $\mathcal{C} = \mathsf{Set}$ those are precisely the locally finite coalgebras (i.e. those coalgebras where every element is contained in a finite subcoalgebra). The final lfp coalgebra exists and is the colimit of the inclusion $\mathsf{Coalg}_{\mathsf{fp}}H \hookrightarrow \mathsf{Coalg}H$, and it is a fixpoint of H (see [6]) called the *rational fixpoint* of H. Here are some examples: the rational fixpoint of a polynomial set functor associated to a finitary signature Σ is the set of rational Σ-trees [6], i.e. finite and infinite Σ-trees having, up to isomorphism, finitely many subtrees only, and one obtains rational weighted languages for Noetherian semirings S for a functor on the category of S-modules [12], and rational λ-trees for a functor on the category of presheaves on finite sets [2] or for a related functor on nominal sets [34]. If the classes of fp and fg objects coincide in $\mathcal{C}$, then the rational fixpoint is a subcoalgebra of the final coalgebra [12, Theorem 3.12]. This is the case in the above examples, but not in general, see [12, Example 3.15] for a concrete example where the rational fixpoint does not identify behaviourally equivalent states. Conversely, even if the classes differ, the rational fixpoint can be a subcoalgebra, e.g. for any constant functor.

Iterative Algebras. If $H : \mathcal{C} \to \mathcal{C}$ is an endofunctor, an H-algebra $(A, a : HA \to A)$ is *iterative* if every *flat equation morphism* $e : X \to HX + A$ where X is an fp object has a unique *solution*, i.e. if there exists a unique $e^\dagger : X \to A$ such that $e^\dagger = [a, \mathrm{id}_A] \cdot (He^\dagger + \mathrm{id}_A) \cdot e$. The rational fixpoint is also characterized as the initial iterative algebra [6] and is the starting point of the coalgebraic approach to Elgot's iterative theories [18] and to the iteration theories of Bloom and Ésik [11, 6, 3, 4].

3 The Locally Finite Fixpoint

The locally finite fixpoint can be characterized similarly to the rational fixpoint, but with respect to coalgebras with finitely generated (not finitely presentable) carrier. We show that the locally finite fixpoint always exists, and is a subcoalgebra of the final coalgebra, i.e. identifies all behaviourally equivalent states. As a consequence, the locally finite fixpoint provides a fully abstract notion of finitely generated behaviour. From now on, we rely on the following:

Assumption 3.1. *Throughout the rest of the paper we assume that $\mathcal{C}$ is an lfp category and that $H : \mathcal{C} \to \mathcal{C}$ is finitary and preserves monomorphisms.*

As for the rational fixpoint, we denote the full subcategory of $\mathsf{Coalg}\,H$ comprising all coalgebras with finitely generated carrier by $\mathsf{Coalg}_{\mathsf{fg}}H$ and have the following notion of locally finitely generated coalgebra.

Definition 3.2. *A coalgebra $X \xrightarrow{x} HX$ is called* locally finitely generated (lfg) *if for all $f : S \to X$ with S finitely generated, there exist a coalgebra $p : P \to HP$ in $\mathsf{Coalg}_{\mathsf{fg}}H$, a coalgebra morphism $h : (P, p) \to (X, x)$ and some $f' : S \to P$ such that $h \cdot f' = f$. $\mathsf{Coalg}_{\mathsf{lfg}}H \subseteq \mathsf{Coalg}\,H$ denotes the full subcategory of lfg coalgebras.*

Equivalently, one can characterize lfg coalgebras in terms of subobjects and subcoalgebras, making it a generalization of of *local finiteness* in Set, i.e. the property of a coalgebra that every element is contained in a finite subcoalgebra.

Lemma 3.3. *$X \xrightarrow{x} HX$ is an lfg coalgebra iff for all fg subobjects $S \overset{f}{\rightarrowtail} X$, there exist a subcoalgebra $h : (P, p) \rightarrowtail (X, x)$ and a mono $f' : S \rightarrowtail P$ with $h \cdot f' = f$, i.e. S is a subobject of P.*

Proof. ($\Rightarrow$) Given some mono $f : S \rightarrowtail X$, factor the induced h into some strong epi-carried and mono-carried homomorphisms and use that fg objects are closed under strong epis. ($\Leftarrow$) Factor $f : S \to X$ into an epi and a mono $g : \mathsf{Im}(f) \rightarrowtail X$ and use the diagonal fill-in property for g. □

Evidently all coalgebras with finitely generated carriers are lfg. Moreover, lfg coalgebras are precisely the filtered colimits of coalgebras from $\mathsf{Coalg}_{\mathsf{fg}}H$.

Proposition 3.4. *Every filtered colimit of coalgebras from* $\mathsf{Coalg}_{\mathsf{fg}}H$ *is lfg.*

Proof (Sketch; for the full proof see the appendix). One first proves that directed unions of coalgebras from $\mathsf{Coalg}_{\mathsf{fg}}H$ are lfg. Now given a filtered colimit $c_i : X_i \to C$ where X_i are coalgebras in $\mathsf{Coalg}_{\mathsf{fg}}H$, one epi-mono factorizes every colimit injection: $c_i = (X_i \xrightarrow{e_i} T_i \xrightarrow{m_i} C)$. Using the diagonalization of the factorization one sees that the T_i form a directed diagram of subobjects of C. Furthermore C is the directed union of the T_i and therefore an lfg coalgebra as desired. □

Proposition 3.5. *Every lfg coalgebra* (X, x) *is a directed colimit of its subcoalgebras from* $\mathsf{Coalg}_{\mathsf{fg}}H$.

Proof. Recall from [8, Proof I of Theorem 1.70] that X is the colimit of the diagram of all its finitely generated subobjects. Now the subdiagram given by all subcoalgebras of X is cofinal. Indeed, this follows directly from the fact that (X, x) is an lfg coalgebra: for every subobject $S \rightarrowtail X$, S fg, we have a subcoalgebra of (X, x) in $\mathsf{Coalg}_{\mathsf{fg}}H$ containing S. □

Corollary 3.6. *The lfg coalgebras are precisely the filtered colimits, or equivalently directed unions, of coalgebras with fg carrier.*

As a consequence, a coalgebra is final in $\mathsf{Coalg}_{\mathsf{lfg}}F$ if there is a unique morphism from every coalgebra with finitely generated carrier.

Proposition 3.7. *An lfg coalgebra* L *is final in* $\mathsf{Coalg}_{\mathsf{lfg}}H$ *iff for every for every coalgebra* X *in* $\mathsf{Coalg}_{\mathsf{fg}}H$ *there exists a unique coalgebra morphism from* X *to* L.

The proof is analogous to [31, Theorem 3.14]; the full argument can be found in the Appendix. Cocompleteness of $\mathcal{C}$ ensures that the final lfg coalgebra always exists.

Theorem 3.8. *The category* $\mathsf{Coalg}_{\mathsf{lfg}}H$ *has a final object, and the final lfg coalgebra is the colimit of the inclusion* $\mathsf{Coalg}_{\mathsf{fg}}H \hookrightarrow \mathsf{Coalg}_{\mathsf{lfg}}H$.

Proof. By Corollary 3.6, the colimit of the inclusion $\mathsf{Coalg}_{\mathsf{fg}}H \hookrightarrow \mathsf{Coalg}_{\mathsf{lfg}}H$ is the same as the colimit of the entire $\mathsf{Coalg}_{\mathsf{lfg}}H$. And the latter is clearly the final object of $\mathsf{Coalg}_{\mathsf{lfg}}H$. □

This theorem provides a construction of the final lfg coalgebra collecting precisely the behaviours of the coalgebras with fg carriers. In the following we shall show that this construction does indeed identify precisely behaviourally equivalent states, i.e. the final lfg coalgebra is always a subcoalgebra of the final coalgebra. Just like fg objects are closed under quotients – in contrast to fp objects – we have a similar property of lfg coalgebras:

Lemma 3.9. *Lfg coalgebras are closed under strong quotients, i.e. for every strong epi carried coalgebra homomorphisms* $X \twoheadrightarrow Y$, *if* X *is lfg then so is* Y.

The failure of this property for lfp coalgebras is the reason why the rational fixpoint is not necessarily a subcoalgebra of the final coalgebra and in particular the rational fixpoint in [12, Example 3.15] is an lfp coalgebra for which the property fails.

Theorem 3.10. *The final lfg H-coalgebra is a subcoalgebra of the final H-coalgebra.*

Proof. Let (L, ℓ) be the final lfg coalgebra. Consider the unique coalgebra morphism $L \to \nu H$ and take its factorization:

$$\begin{tikzcd} (L,\ell) \arrow[loop left, dashed, "\mathrm{id}"] \arrow[r, two heads, shift left, "e"] & (I,i) \arrow[l, dashed, shift left, "i^\dagger"] \arrow[r, tail, "m"] & (\nu H, \tau) \end{tikzcd}, \quad \text{with } e \text{ strong epi in } \mathcal{C}.$$

By Lemma 3.9, I is an lfg coalgebra and so by finality of L we have the coalgebra morphism $i^\dagger$ such that $\mathrm{id}_L = i^\dagger \cdot e$. It follows that e is monic and therefore an iso. □

In other words, the final lfg H-coalgebra collects precisely the finitely generated behaviours from the final H-coalgebra. We now show that the final lfg coalgebra is a fixpoint of H which hinges on the following:

Lemma 3.11. *For any lfg coalgebra $C \xrightarrow{c} HC$, the coalgebra $HC \xrightarrow{Hc} HHC$ is lfg.*

Proof. Consider $f : S \to HC$ with S finitely generated. As $\mathcal{C}$ is lfp we know that HC is the colimit of its fg subobjects, and so $f : S \to HC$ factors through some subobject $\mathsf{in}_q : Q \rightarrowtail HC$ with Q fg and $f = \mathsf{in}_q \cdot f'$. On the other hand, (C, c) is lfg, i.e. the directed union of its subcoalgebras from $\mathsf{Coalg}_{\mathsf{fg}} H$. Then, since H is finitary and mono-preserving, $HC \xrightarrow{c} HHC$ is also a directed union and the morphism $\mathsf{in}_q : Q \to HC$ factors through some $HP \xrightarrow{Hp} HHP$ with $(P, p) \in \mathsf{Coalg}_{\mathsf{fg}} H$ via $\mathsf{in}_p : (P, p) \rightarrowtail (C, c)$, i.e. $H\mathsf{in}_p \cdot q = \mathsf{in}_q$. Finally, we can construct a coalgebra with fg carrier

$$Q + P \xrightarrow{[q,p]} HP \xrightarrow{H\mathsf{inr}} H(Q + P)$$

and a coalgebra homomorphism $H\mathsf{in}_p \cdot [q, p] : Q + P \to HC$. In the diagram

$$\begin{tikzcd}
S \arrow[r, "f"] \arrow[dd, "f'"'] & HC \arrow[rr, "Hc"] & & HHC \\
& HP \arrow[u, "H\mathsf{in}_p"'] \arrow[rr, "Hp"] \arrow[dr, equal] & & HHP \arrow[u, "HH\mathsf{in}_p"'] \\
Q \arrow[uur, "\mathsf{in}_q"] \arrow[ur, "q"] \arrow[r, "\mathsf{inl}"] & Q+P \arrow[u, "{[q,p]}"] \arrow[r, "{[q,p]}"] & HP \arrow[r, "H\mathsf{inr}"] & H(Q+P) \arrow[u, "{H[q,p]}"'] \arrow[uu, bend right=60, "H(H\mathsf{in}_p\cdot[q,p])"']
\end{tikzcd}$$

every part trivially commutes, so $H\mathsf{in}_p \cdot [q, p]$ is the desired homomorphism. □

So with a proof in virtue to Lambek's Lemma [28, Lemma 2.2], we obtain the desired fixpoint:

Theorem 3.12. *The carrier of the final lfg H-coalgebra is a fixpoint of H.*

We denote the above fixpoint by $(\vartheta H, \ell)$ and call it the *locally finite fixpoint* (LFF) of H. In particular, the LFF always exists under Assumption 3.1, providing a finitary corecursion principle.

3.1 Iterative Algebras

Recall from [6, 31] that the rational fixpoint of a functor H has a universal property both as a coalgebra and as an algebra for H. This situation is completely analogous for the LFF. We already established its universal property as a coalgebra in Theorem 3.8. Now we turn to study the LFF as an algebra for H.

Definition 3.13. *An* equation morphism e *in an object A is a morphism $X \to HX + A$, where X is a finitely generated object. If A is the carrier of an algebra $\alpha : HA \to A$, we call the $\mathcal{C}$-morphism $e^\dagger : X \to A$ a* solution *of e if $[\alpha, \mathrm{id}_A] \cdot He^\dagger + \mathrm{id}_A \cdot e = e^\dagger$. An H-algebra A is called* fg-iterative *if every equation morphism in A has a unique solution.*

Example 3.14 (see [30, Example 2.5 (iii)]). The final H-coalgebra (considered as an algebra for H) is fg-iterative. In fact, in this algebra even morphisms $X \to HX + \nu H$ where X is not necessarily an fg object have a unique solution.

Definition 3.15. *For fg-iterative algebras A and C, an equation morphism $e : X \to HX + A$ and a morphism $h : A \to C$ of $\mathcal{C}$ define an equation morphism $h \bullet e$ in C as $X \xrightarrow{e} HX + A \xrightarrow{HX+h} HX + C$. We say that h preserves the solution $e^\dagger$ of e if $h \cdot e^\dagger = (h \bullet e)^\dagger$. The morphism h is called* solution preserving *if it preserves the solution of any equation morphism e.*

Similarly to [6], the algebra homomorphisms are precisely the solution preserving morphisms between iterative algebras, the proof is also very similar.

Proposition 3.16. *The locally finite fixpoint is fg-iterative.*

Theorem 3.17. *For an fg-iterative algebra $\alpha : HA \to A$ and an lfg coalgebra $e : X \to HX$ there is a unique $\mathcal{C}$-morphism $u_e : X \to A$ such that $u_e = \alpha \cdot Hu_e \cdot e$.*

Corollary 3.18. *The locally finite fixpoint is the initial fg-iterative algebra.*

3.2 Relation to the Rational Fixpoint

There are examples, where the rational fixpoint is not a subcoalgebra of the final coalgebra. In categories, where fp and fg objects coincide, the rational fixpoint and the LFF coincide as well (cf. the respective colimit-construction in Section 2 and Theorem 3.8). In this section we will see, under slightly stronger assumptions, that fg-carried coalgebras are quotients of fp-carried coalgebras, and in particular the locally finite fixpoint is a quotient of the rational fixpoint: namely its image in the final coalgebra.

Assumption 3.19. *In addition to Assumption 3.1, assume that in the base category $\mathcal{C}$, every finitely presentable object is a strong quotient of a finitely presentable strong epi projective object and that the endofunctor H also preserves strong epis.*

The condition that every fg object is the strong quotient of a strong epi projective often is phrased as *having enough strong epi projectives* [14]. This assumption is apparently very strong but still is met in many situations:

Example 3.20.
- In categories in which all (strong) epis are split, every object is projective and any endofunctor preserves epis, e.g. in Set or Vec_K.
- In the category of finitary endofunctors $\mathsf{Fun_f}(\mathsf{Set})$, all polynomial functors are projective. The finitely presentable functors are quotients of polynomial functors H_Σ, where Σ is a finite signature.
- In the Eilenberg-Moore category Set^T for a finitary monad T, strong epis are surjective T-algebra homomorphisms, and thus preserved by any endofunctor. In Set^T, every free algebra TX is projective; this is easy to see using the projectivity of X in Set. Every finitely generated object of Set^T is a strong quotient of some free algebra TX with X finite. For more precise definitions, see Section 4.1 later.

Proposition 3.21. *Every coalgebra in* $\mathsf{Coalg_{fg}}H$ *is a strong quotient of a coalgebra with finitely presentable carrier.*

Theorem 3.22. ϑH *is the image of the rational fixpoint* ϱH *in the final coalgebra.*

Proof. Consider the factorization $(\varrho H, r) \overset{e}{\twoheadrightarrow} (B, b) \overset{m}{\rightarrowtail} (\nu H, \tau)$. Since ϱH is the colimit of all fp carried H-coalgebras it is an lfg coalgebra by Proposition 3.4 using that fp objects are also fg. Hence, by Lemma 3.9 the coalgebra B is lfg, too. By Proposition 3.7 it now suffices to show that from every $(X, x) \in \mathsf{Coalg_{fg}}H$ there exists a unique coalgebra morphism into (B, b). Given (X, x) in $\mathsf{Coalg_{fg}}H$, it is the quotient $q : (P, p) \twoheadrightarrow (X, x)$ of an fp-carried coalgebra by Proposition 3.21. Hence, we obtain a unique coalgebra morphism $p^\dagger : (P, p) \to (\varrho H, r)$. By finality of νH, we have $m \cdot e \cdot p^\dagger = x^\dagger \cdot q$ (with $x^\dagger : (X, x) \to (\nu H, \tau)$). So the diagonal fill-in property induces a homomorphism $(X, x) \to (B, b)$, being the only homomorphism $(X, x) \to (B, b)$ by the finality of νH and because m is monic. □

4 Instances of the Locally Finite Fixpoint

We will now present a number of instances of the LFF. First note, that all the known instances of the rational fixpoint (see e.g. [6, 31, 12] are also instances of the locally finite fixpoint, because in all those cases the fp and fg objects coincide. For example, the class of regular languages is the rational fixpoint of $2 \times (-)^\Sigma$ on Set. In this section, we will study further instances of the LFF that are most likely not instances of the rational fixpoint and which – to the best of our knowledge – have not been characterized by a universal property yet:

1. Behaviours of finite-state machines with side-effects as considered by the generalized powerset construction (cf. Section 4.1),particularly the following.
   (a) Deterministic and ordinary context-free languages obtained as the behaviours of deterministic and non-deterministic stack-machines, respectively.
   (b) Constructively S-algebraic formal power series, i.e. the "context-free" subclass of weighted languages with weights from a semiring S, yielded from weighted context-free grammars.
2. The monad of Courcelle's algebraic trees.

4.1 Generalized Powerset Construction

The determinization of a non-deterministic automaton using the powerset construction is an instance of a more general framework, described by Silva, Bonchi, Bonsangue, and Rutten [41] based on an observation by Bartels [10] (see also Jacobs [26]). In that *generalized powerset construction*, an automaton with side-effects is turned into an ordinary automaton by internalizing the side-effects in the states. The LFF interacts well with this construction, because it precisely captures the behaviours of finite-state automata with side effects. The notion of side-effect is formalized by a monad, which induces the category, in which the LFF is considered.

In the following we assume that readers are familiar with monads and Eilenberg-Moore algebras (see e.g. [29] for an introduction). For a monad T on $\mathcal{C}$ we denote by $\mathcal{C}^T$ the category of Eilenberg-Moore algebras. Recall from [8, Corollary 2.75] that if $\mathcal{C}$ is lfp (in most of our examples $\mathcal{C}$ is Set) and T is finitary then $\mathcal{C}^T$ is lfp, too, and for every fp object X the free Eilenberg-Moore algebra TX is fp in $\mathcal{C}^T$. In all the examples we consider below, the classes of fp and fg objects either provably differ or it is still unknown whether these classes coincide.

Example 4.1. In Sections 4.4 and 4.5 we are going to make use of Moggi's exception monad transformer (see e.g. [15]). Let us recall that for a fixed object E, the finitary functor $(-) + E$ together with the unit $\eta_X = \mathsf{inl} : X \to X + E$ and multiplication $\mu_X = \mathrm{id}_X + [\mathrm{id}_E, \mathrm{id}_E] : X + E + E \to X + E$ form a finitary monad, the *exception monad*. Its algebras are E-pointed objects, i.e. objects X, together with a morphism $E \to X$, and homomorphisms are morphisms preserving the pointing. So the induced Eilenberg-Moore category is just the slice category $\mathcal{C}^{(-)+E} \cong E/\mathcal{C}$.

Now, given any monad T we obtain a new monad $T(- + E)$ with obvious unit and multiplication. An Eilenberg-Moore algebra for $T(- + E)$ consists of an Eilenberg-Moore algebra for T and an E-pointing, and homomorphisms are T-algebra homomorphisms preserving the pointing [25].

Now an automaton with side-effects is modelled as an HT-coalgebra, where T is a finitary monad on $\mathcal{C}$ providing the type of side-effect. For example, for $HX = 2 \times X^\Sigma$, where Σ is an input alphabet, $2 = \{0, 1\}$ and T the finite powerset monad on Set, HT-coalgebras are non-deterministic automata. However, the coalgebraic semantics using the final HT-coalgebra does not yield the usual language semantics of non-deterministic automata. To obtain this one considers the final coalgebra of a lifting of H to $\mathcal{C}^T$. Denote by $U : \mathcal{C}^T \to \mathcal{C}$ the canonical forgetful functor.

Definition 4.2. *For a functor $H : \mathcal{C} \to \mathcal{C}$ and a monad $T : \mathcal{C} \to \mathcal{C}$, a* lifting *of H is a functor $H^T : \mathcal{C}^T \to \mathcal{C}^T$ such that $H \cdot U = U \cdot H^T$.*

If such a (not necessarily unique) lifting exists, the generalized powerset construction transforms an HT-coalgebra into a H^T-coalgebra on $\mathcal{C}^T$: For a coalgebra $x : X \to HTX$, HTX carries an Eilenberg-Moore algebra, and one uses freeness of the Eilenberg-Moore algebra TX to obtain a canonical T-algebra homomorphism $x^\sharp : (TX, \mu^T) \to H^T(TX, \mu^T)$. The *coalgebraic language semantics* of (X, x) is then given by $X \xrightarrow{\eta_X} TX \xrightarrow{x^{\sharp\dagger}} \nu H^T$, i.e. by composing the unique coalgebra morphism induced by $x^\sharp$ with η_X. This construction yields a functor $T' : \mathsf{Coalg}(HT) \to \mathsf{Coalg} H^T$ mapping coalgebras

$X \xrightarrow{x} HTX$ to $x^\sharp$ and homomorphisms f to Tf (see e.g. [12, Proof of Lemma 3.27] for a proof).

Now our aim is to show that the LFF of H^T characterizes precisely the coalgebraic language semantics of all fp-carried HT-coalgebras. As the right adjoint U preserves monos and is faithful, we know that H^T preserves monos, and as T is finitary, filtered colimits in $\mathcal{C}^T$ are created by the forgetful functor to $\mathcal{C}$, and we therefore see that H^T is finitary. Thus, by Theorem 3.8, ϑH^T exists and is a subcoalgebra of νH^T. By [37] and [10, Corollary 3.4.19], we know that νH^T is carried by νH equipped with a canonical algebra structure.

Now let us turn to the desired characterization of ϑH^T. Formally, the coalgebraic language semantics of all fp-carried HT-coalgebras is collected by forming the colimit $k : K \to HK$ of the diagram $\mathsf{Coalg}_{\mathsf{fg}} HT \xrightarrow{T'} \mathsf{Coalg} H^T \xrightarrow{U} \mathsf{Coalg} H$. This coalgebra K is not yet a subcoalgebra of νH (for $\mathcal{C} = \mathsf{Set}$ that means, not all behaviourally equivalent states are identified in K), but taking its image in νH we obtain the LFF:

Proposition 4.3. *The image (I, i) of the unique coalgebra morphism $k^\dagger : K \to \nu H^T$ is precisely the locally finite fixpoint of the lifting H^T.*

One can also directly take the union of all desired behaviours, for $\mathcal{C} = \mathsf{Set}$:

Theorem 4.4. *The locally finite fixpoint of the lifting H^T comprises precisely the images of determinized HT-coalgebras:*

$$\vartheta H^T = \bigcup_{\substack{x: X \to HTX \\ X \text{ finite}}} x^{\sharp\dagger}[TX] = \bigcup_{\substack{x: X \to HTX \\ X \text{ finite}}} x^{\sharp\dagger} \cdot \eta_X^T[X] \subseteq \nu H^T. \tag{1}$$

This result suggests that the locally finite fixpoint is the right object to consider in order to represent finite behaviour. We now instantiate the general theory with examples from the literature to characterize several well-known notions as LFF.

4.2 The Languages of Non-deterministic Automata

Let us start with a simple standard example. We already mentioned that non-deterministic automata are coalgebras for the functor $X \mapsto 2 \times \mathcal{P}_\mathsf{f}(X)^\Sigma$. Hence they are HT-coalgebras for $H = 2 \times (-)^\Sigma$ and $T = \mathcal{P}_\mathsf{f}$ the finite powerset monad on Set. The above generalized powerset construction then instantiates as the usual powerset construction that assigns to a given non-deterministic automaton its determinization.

Now note that the final coalgebra for H is carried by the set $\mathcal{L} = \mathcal{P}(\Sigma^*)$ of all formal languages over Σ with the coalgebra structure given by $o : \mathcal{L} \to 2$ with $o(L) = 1$ iff L contains the empty word and $t : \mathcal{L} \to \mathcal{L}^\Sigma$ with $t(L)(s) = \{w \mid sw \in L\}$ the left language derivative. The functor H has a canonical lifting H^T on the Eilenberg-Moore category of $\mathcal{P}_\mathsf{f}$, viz. the category of join semi-lattices. The final coalgebra νH^T is carried by all formal languages with the join semi-lattice structure given by union and $\emptyset$ and with the above coalgebra structure. Furthermore, the coalgebraic language semantics of $x : X \to HTX$ assigns to every state of the non-deterministic automaton X the language it accepts. Observe that join semi-lattices form a so-called *locally finite variety*, i.e. the finitely presentable algebras are precisely the finite ones. Hence, Theorem 4.4 states that the LFF of H^T is precisely the subcoalgebra of νH^T formed by all languages accepted by finite NFA, i.e. regular languages.

Note that in this example the LFF and the rational fixpoint coincide since both fp and fg join semi-lattices are simply the finite ones. Similar characterizations of the coalgebraic language semantics of finite coalgebras follow from Theorem 4.4 in other instances of the generalized powerset construction from [41] (cf. e.g. the treatment of the behaviour of finite weighted automata in [12]).

We now turn to examples that, to the best of our knowledge, cannot be treated using the rational fixpoint.

4.3 The Behaviour of Stack Machines

Push-down automata are finite state machines with infinitely many configurations. It is well-known that deterministic and non-deterministic pushdown automata recognize different classes of context-free languages. We will characterize both as instances of the locally finite fixpoint, using the results from [23] on stack machines, which can push or read *multiple* elements to or from the stack in a single transition, respectively.

That is, a transition of a stack machine in a certain state consists of reading an input character, going to a successor state based on the stack's topmost elements and of modifying the topmost elements of the stack. These stack operations are captured by the stack monad.

Definition 4.5 (Stack monad, [22, Proposition 5]). *For a finite set of stack symbols Γ, the* stack monad *is the submonad T of the store monad $(- \times \Gamma^*)^{\Gamma^*}$ for which the elements $\langle r, t\rangle$ of $TX \subseteq (X \times \Gamma^*)^{\Gamma^*} \cong X^{\Gamma^*} \times (\Gamma^*)^{\Gamma^*}$ satisfy the following restriction: there exists k depending on r, t such that for every $w \in \Gamma^k$ and $u \in \Gamma^*$, $r(wu) = r(w)$ and $t(wu) = t(w)u$.*

Note that the parameter k gives a bound on how may of the topmost stack cells the machine can access in one step.

Using the stack monad, stack machines are HT-coalgebras, where $H = B \times (-)^\Sigma$ is the Moore automata functor for the finite input alphabet Σ and the set B of all predicates mapping (initial) stack configurations to output values from 2, taking only the topmost k elements into account: $B = \{p \in 2^{\Gamma^*} \mid \exists k \in \mathbb{N}_0 : \forall w, u \in \Gamma^*, |w| \geq k : p(wu) = p(w)\} \subseteq 2^{\Gamma^*}$.

The final coalgebra νH is carried by B^{Σ^*} which is (modulo power laws) a set of predicates, mapping stack configurations to formal languages. Goncharov et al. [23] show that H lifts to Set^T and conclude that finite-state HT-coalgebras match the intuition of *deterministic* pushdown automata without spontaneous transitions. The languages accepted by those automata are precisely the *real-time deterministic context-free languages*; this notion goes back to Harrison and Havel [24]. We obtain the following, with γ_0 playing the role of an initial symbol on the stack:

Theorem 4.6. *The locally finite fixpoint of H^T is carried by the set of all maps $f \in B^{\Sigma^*}$ such that for any fixed $\gamma_0 \in \Gamma$, $\{w \in \Sigma^* \mid f(w)(\gamma_0) = 1\}$ is a real-time deterministic context-free language.*

Proof. By [23, Theorem 5.5], a language L is a real-time deterministic context-free language iff there exists some $x : X \to HTX$, X finite, with its determinization $x^\sharp : TX \to HTX$ and there exist $s \in X$ and $\gamma_0 \in \Gamma$ such that $f = x^{\sharp\dagger} \cdot \eta_X^T(s) \in B^{\Sigma^*}$ and $f(w)(\gamma_0) = 1$ for all $w \in \Sigma^*$. The rest follows by (1). $\square$

Just as for pushdown automata, the expressiveness of stack machines increases when equipping them with non-determinism. Technically, this is done by considering the *non-deterministic stack monad* T', i.e. T' denotes a submonad of the non-deterministic store monad $\mathcal{P}_{\mathsf{f}}(- \times \Gamma^*)^{\Gamma^*}$, as described in [23, Section 6]. In the non-deterministic setting, a similar property holds, namely that the determinized HT'-coalgebras with finite carrier describe precisely the context-free languages [23, Theorem 6.5]. Combine this with (1):

Theorem 4.7. *The locally finite fixpoint of $H^{T'}$ is carried by the set of all maps $f \in B^{\Sigma^*}$ such that for any fixed $\gamma_0 \in \Gamma$, $\{w \in \Sigma^* \mid f(w)(\gamma_0) = 1\}$ is a context-free language.*

4.4 Context-Free Languages and Constructively S-Algebraic Power Series

One generalizes from formal (resp. context-free) languages to weighted formal (resp. context-free) languages by assigning to each word a weight from a fixed semiring. More formally, a weighted language – a.k.a. *formal power series* – over an input alphabet X is defined as a map $X^* \to S$, where S is a semiring. The set of all formal power series is denoted by $S\langle\langle X\rangle\rangle$. Ordinary formal languages are formal power series over the boolean semiring $\mathbb{B} = \{0, 1\}$, i.e. maps $X^* \to \{0, 1\}$.

An important class of formal power series is that of *constructively S-algebraic* formal power series. We show that this class arises precisely as the LFF of the standard functor for deterministic Moore automata $H = S \times (-)^{\Sigma}$, but on an Eilenberg-Moore category of a Set monad. As a special case, constructively $\mathbb{B}$-algebraic series are the context-free weighted languages and are precisely the LFF of the automata functor in the category of idempotent semirings.

The original definition of constructively S-algebraic formal power series goes back to Fliess [19], see also [17]. Here, we use the equivalent coalgebraic characterization by Winter et al. [44].

Let $S\langle X\rangle \subseteq S\langle\langle X\rangle\rangle$ the subset of those maps, that are 0 for all but finitely many $w \in X^*$. If S is commutative, then $S\langle -\rangle$ yields a finitary monad and thus also $T = S\langle - + \Sigma\rangle$ by Example 4.1. The algebras for $S\langle -\rangle$ are associative S-algebras (over the commutative semiring S), i.e. S-modules together with a monoid structure that is a module morphism in both arguments. The algebras for T are Σ-pointed S-algebras. The following notions are special instances of S-algebras.

Example 4.8. For $S = \mathbb{B} = \{0, 1\}$, one obtains idempotent semirings as $\mathbb{B}$-algebras, for $S = \mathbb{N}$ semirings, and for $S = \mathbb{Z}$ ordinary rings.

Winter et al. [44, Proposition 4] show that the final H-coalgebra is carried by $S\langle\langle\Sigma\rangle\rangle$ and that constructively S-algebraic series are precisely those elements of $S\langle\langle\Sigma\rangle\rangle$ that arise as the behaviours of those coalgebra $c : X \to HS\langle X\rangle$ with finite X, after determinizing them to some $c^\sharp : S\langle X\rangle \to HS\langle X\rangle$ (see [44, Theorem 23]).

However, this determinization is not directly an instance of the generalized powerset construction. We shall show that the same behaviours can be obtained by using the standard generalized powerset construction with an appropriate lifting of H to T-algebras. Having an S-algebra structure on A and a Σ-pointing $j : \Sigma \to A$ we need to define another S-algebra structure and Σ-pointing on $HA = S \times A^{\Sigma}$. While the S-module structure is just point-wise, we need to take care when multiplying two elements from HA. To this end we first we define the operation $[-,-] : S \times A^\Sigma \to A$ by

$$[o, \delta] := i(o) + \sum_{b \in \Sigma} \big(j(b) \cdot \delta(b)\big),$$

where $i : S \to A$ is the canonical map with $i(s) = s \cdot 1$ with 1 the neutral element of the monoid on A. The idea is that $[o, \delta]$ acts like a state with output o and derivation δ. The multiplication on $HA = S \times A^\Sigma$ is then defined by

$$(o_1, \delta_1) * (o_2, \delta_2) := \big(o_1 \cdot o_2, a \mapsto \delta_1(a) \cdot [o_2, \delta_2] + i(o_1) \cdot \delta_2(a)\big). \qquad (2)$$

The Σ-pointing is the obvious: $a \mapsto (0, \varrho_a)$ where $\varrho_a(a) = 1$ and $\varrho_a(b) = 0$ for $a \neq b$.

Lemma 4.9. *For any $w \in A$ in Set^T and any H^T-coalgebra structure $c : A \to H^T A$, w and $[c(w)]$ are behaviourally equivalent in* Set.

Given a coalgebra $c : X \to HS\langle X\rangle$, Winter et al. [44, Proposition 14] determinize c to some $\hat{c} = \langle \hat{o}, \hat{\delta}\rangle : S\langle X\rangle \to HS\langle X\rangle$ with the property that for any $v, w \in S\langle X\rangle$,

$$\hat{o}(v * w) = \hat{o}(v) \cdot \hat{o}(w) \quad \text{and} \quad \hat{\delta}(v * w, a) = \hat{\delta}(v, a) * w + \hat{o}(v) * \hat{\delta}(w, a), \qquad (3)$$

and such that $\hat{c}$ is a S-module homomorphism. However, the generalized powerset construction w.r.t. T yields a coalgebra $c^\sharp : S\langle X + \Sigma\rangle \to HS\langle X + \Sigma\rangle$. The above property, together with Lemma 4.9 and (2) implies that $\hat{c}$ and $c^\sharp$ are essentially the same coalgebra structures:

Lemma 4.10. *In* Set*, $u \in (S\langle X\rangle, \hat{c})$ and $S\langle \mathsf{inl}\rangle(u) \in (S\langle X+\Sigma\rangle, c^\sharp)$ are behaviourally equivalent.*

It follows that $\hat{c}^\dagger = c^{\sharp\dagger} \cdot S\langle \mathsf{inl}\rangle$ and thus their images in νH are identical. Hence, a formal power series is constructively S-algebraic iff it is in the image of some $c^{\sharp\dagger} \cdot S\langle \mathsf{inl}\rangle$, and by (1), iff it is in the locally finite fixpoint of H^T.

4.5 Courcelle's Algebraic Trees

For a fixed signature Σ of so called *givens*, a *recursive program scheme* (or *rps*, for short) contains mutually recursive definitions of new operations $\varphi_1, \dots, \varphi_k$ (with respective arities $n_1, \dots, n_k$). The recursive definition of φ_i may involve symbols from Σ, operations $\varphi_1, \dots, \varphi_k$ and n_i variables $x_1, \dots, x_{n_i}$. The (uninterpreted) solution of an rps is obtained by unravelling these recursive definitions, producing a possibly infinite Σ-tree over $x_1, \dots, x_{n_i}$ for each operation φ_i. Figure 1 shows an rps over the signature $\Sigma = \{\star/0, \times/2, +/2\}$ and its solution. In general, an *algebraic Σ-tree* is a Σ-tree which is definable by an rps over Σ (see Courcelle [16]). Generalizing from a signature to a finitary endofunctor $H : \mathcal{C} \to \mathcal{C}$ on an lfp category, Adámek et al. [7] describe an rps as a coalgebra for a functor $\mathscr{H}_\mathsf{f}$ on $H/\mathsf{Mnd}_\mathsf{f}(\mathcal{C})$, in which objects are finitary H-pointed monads on $\mathcal{C}$, i.e. finitary monads M together with a natural transformation $H \to M$. They introduce the *context-free* monad C^H of H, which is an H-pointed monad that is a subcoalgebra

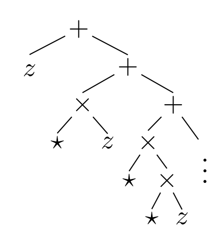

Fig. 1. Solution of $\varphi(z) = z + \varphi(\star \times z)$

of the final coalgebra for $\mathscr{H}_{\mathsf{f}}$ and which is the monad of Courcelle's algebraic Σ-trees in the special case where $\mathcal{C} = \mathsf{Set}$ and H is a polynomial functor associated to a signature Σ. We will prove that this monad is the LFF of $\mathscr{H}_{\mathsf{f}}$, and thereby we characterize it by a universal property – solving the open problem in [7].

The setting is again an instance of the generalized powerset construction, but this time with $\mathsf{Fun_f}(\mathcal{C})$ as the base category in lieu of Set. Let $\mathcal{C}$ be an lfp category in which the coproduct injections are monic and consider a finitary, mono-preserving endofunctor $H : \mathcal{C} \to \mathcal{C}$. Denote by $\mathsf{Fun_f}(\mathcal{C})$ the category of finitary endofunctors on $\mathcal{C}$. Then H induces an endofunctor $H \cdot (-) + \mathrm{Id}$ on $\mathsf{Fun_f}(\mathcal{C})$, denoted $\dot{H}$ and mapping an endofunctor V to the functor $X \mapsto HVX + X$. This functor $\dot{H}$ gets precomposed with a monad on $\mathsf{Fun_f}(\mathcal{C})$ as we now explain.

Proposition 4.11 (Free monad, [5, 9]). *For a finitary endofunctor H, free H-algebras $\varphi_X : HF^H X \to F^H X$ exist for all $X \in \mathcal{C}$. F^H itself is a finitary monad on $\mathcal{C}$, more specifically it is the* free monad *on H.*

For example, if H is a polynomial functor associated to a signature Σ, then $F^H X$ is the usual term algebra that contains all finite Σ-trees over the set of generators X. Proposition 4.11 implies that $H \mapsto F^H$ is the object assignment of a monad on $\mathsf{Fun_f}(\mathcal{C})$. The Eilenberg-Moore category of $F^{(-)}$ is easily seen to be $\mathsf{Mnd_f}(\mathcal{C})$, the category of finitary monads on $\mathcal{C}$. Here, fp and fg objects differ, see [45, Section 5.4.1] for a proof.

Similarly as in the case of context-free languages, we will work with the monad $E^{(-)} = F^{H+(-)}$, so we get H-pointed finitary monads as the $E^{(-)}$-algebras. This category is equivalent to a slice category: the universal property induced by $F^{(-)}$ states, that for any finitary monad B the natural transformations $H \to B$ are in one-to-one correspondence with monad morphisms $F^H \to B$; so the category $H/\mathsf{Mnd_f}(\mathcal{C})$ of finitary H-pointed monads on $\mathcal{C}$ is isomorphic to the slice category $F^H/\mathsf{Mnd_f}(\mathcal{C})$. This finishes the description of the base category and we now lift the functor $\dot{H}$ to this category.

Consider an H-pointed monad $(B, \beta : H \to \mathcal{C}) \in H/\mathsf{Mnd_f}(\mathcal{C})$. By [21], the endofunctor $H \cdot B + \mathrm{Id}$ carries a canonical monad structure. Furthermore, we have an obvious pointing $\mathsf{inl} \cdot H\eta^B : H \to H \cdot B + \mathrm{Id}$. By [33], this defines an endofunctor on H-pointed monads, $\mathscr{H}_{\mathsf{f}} : H/\mathsf{Mnd_f}(\mathcal{C}) \to H/\mathsf{Mnd_f}(\mathcal{C})$, which is a lifting of $\dot{H}$. In order to verify that $\mathscr{H}_{\mathsf{f}}$ is finitary, we first need to know how filtered colimits look in $H/\mathsf{Mnd_f}(\mathcal{C})$.

Lemma 4.12. *The forgetful $U : \mathsf{Mnd_f}(\mathcal{C}) \to \mathsf{Fun_f}(\mathcal{C})$ creates filtered colimits.*

Clearly, the canonical projection functor $H/\mathsf{Mnd_f}(\mathcal{C}) \to \mathsf{Mnd_f}(\mathcal{C})$ creates filtered colimits, too. Therefore, filtered colimits in the slice category $H/\mathsf{Mnd_f}(\mathcal{C})$ are formed on the level of $\mathsf{Fun_f}(\mathcal{C})$, i.e. object-wise. The functor $\dot{H}$ is finitary on $\mathsf{Fun_f}(\mathcal{C})$ and thus also its lifting $\mathscr{H}_{\mathsf{f}}$ is finitary. So all requirements from Assumption 3.1 are met: we have a finitary endofunctor $\mathscr{H}_{\mathsf{f}}$ on the lfp category $H/\mathsf{Mnd_f}(\mathcal{C})$, and by [7, Corollary 2.20] $\mathscr{H}_{\mathsf{f}}$ preserves monos since H does. By Theorem 3.8, $\mathscr{H}_{\mathsf{f}}$ has a locally finite fixpoint.

Remark 4.13. The final $\mathscr{H}_{\mathsf{f}}$-coalgebra is not of much interest, but that of a related functor. $\mathscr{H}_{\mathsf{f}}$ generalizes to a functor $\mathscr{H} : H/\mathsf{Mnd_c}(\mathcal{C}) \to H/\mathsf{Mnd_c}(\mathcal{C})$ on H-pointed countably accessible[3] monads. For any object $X \in \mathcal{C}$, the finitary endofunctor $H(-) + X$ has a

[3] A colimit is *countably filtered* if its diagram has for every countable subcategory a cocone. A functor is *countably accessible* if it preserves countably filtered colimits.

final coalgebra; call the carrier TX. Then T is a monad [1], is countably accessible [7] and is the final $\mathcal{H}$-coalgebra [33].

Adámek et al. [7] characterize a (guarded) recursive program scheme as a natural transformation $V \to H \cdot E^V + \mathrm{Id}$ with V fp (in $\mathsf{Fun_f}(\mathcal{C})$), or equivalently, via the generalized powerset construction w.r.t. the monad $E^{(-)}$ as an $\mathcal{H}_\mathsf{f}$-coalgebra on the carrier E^V (in $\mathsf{Mnd_f}(\mathcal{C})$). These $\mathcal{H}_\mathsf{f}$-coalgebras on carriers E^V where $V \in \mathsf{Fun_f}(\mathcal{C})$ is fp form the full subcategory $\mathsf{EQ} \subseteq \mathsf{Coalg}\mathcal{H}_\mathsf{f}$. They show two equivalent ways of constructing the monad of Courcelle's algebraic trees for the case $\mathcal{C} = \mathsf{Set}$: as the image of $\operatorname{colim} \mathsf{EQ}$ in the final coalgebra T of Remark 4.13, and as the colimit of EQ_2, where EQ_2 is the closure of EQ under strong quotients. We now provide a third characterization, and show that the monad of Courcelle's algebraic trees is the locally finite fixpoint of $\mathcal{H}_\mathsf{f}$.

To this end it suffices to show that EQ_2 is precisely the diagram of $\mathcal{H}_\mathsf{f}$-coalgebras with an fg carrier. This is established with the help of the following two technical lemmas. We now assume that $\mathcal{C} = \mathsf{Set}$.

Lemma 4.14. *$\mathcal{H}_\mathsf{f}$ maps strong epis to morphisms carried by strong epi natural transformations.*

We have the following variation of Proposition 3.21:

Lemma 4.15. *Any $\mathcal{H}_\mathsf{f}$-coalgebra $b : (B, \beta) \to \mathcal{H}_\mathsf{f}(B, \beta)$, with B fg, is the strong quotient of a coalgebra from* EQ.

The proof of Lemma 4.15 makes use of Lemma 4.14 as well as the following properties:

- The fp objects in $\mathsf{Fun_f}(\mathsf{Set})$ are the quotients of polynomial functors.
- The polynomial functors are projective. That means that for a polynomial functor P and any natural transformation $n : K \to L$ with surjective components we have the following property: for every $f : P \to L$ there exists $f' : P \to K$ with $n \cdot f' = f$.
- Any fg object in $H/\mathsf{Mnd_f}(\mathsf{Set})$ is the quotient of some E^V with V fp in $\mathsf{Fun_f}(\mathsf{Set})$ and thus also of some E^P with P a polynomial functor.

Note that the last property holds because $H/\mathsf{Mnd_f}(\mathsf{Set})$ is an Eilenberg-Moore category and E^V is the free Eilenberg-Moore algebra on the fp object V. It follows from Lemma 4.15 that $\mathsf{Coalg_{fg}}\mathcal{H}_\mathsf{f}$ is the same category as EQ_2; thus their colimits in $\mathsf{Coalg}\mathcal{H}_\mathsf{f}$ are isomorphic and we conclude:

Theorem 4.16. *The locally finite fixpoint of $\mathcal{H}_\mathsf{f} : H_\Sigma/\mathsf{Mnd_f}(\mathsf{Set}) \to H_\Sigma/\mathsf{Mnd_f}(\mathsf{Set})$ is the monad of Courcelle's algebraic trees, sending a set to the algebraic Σ-trees over it.*

5 Conclusions and Future Work

We have introduced the locally finite fixpoint of a finitary mono-preserving endofunctor on an lfp category. We proved that this fixpoint is characterized by two universal properties: it is the final lfg coalgebra and the initial fg-iterative algebra for the given endofunctor. And we have seen many instances where the LFF is the domain of behaviour of finite-state and finite-equation systems. In particular all previously known instances of the rational fixpoint are also instances of the LFF, and we have obtained a number of interesting further instances not captured by the rational fixpoint.

On a more technical level, the LFF solves a problem that sometimes makes the rational fixpoint hard to apply. The latter identifies behaviourally equivalent states (i.e. is a subcoalgebra of the final coalgebra) if the classes of fp and fg objects coincide. This condition, however, may be false or unknown (and sometimes non-trivial to establish) in a given lfp category. But the LFF always identifies behaviourally equivalent states.

There are a number of interesting topics for future work concerning the LFF. First, it should be interesting to obtain further instances of the LFF, e.g. analyzing the behaviour of tape machines [23] may perhaps lead to a description of the recursively enumerable languages by the LFF. Second, syntactic descriptions of the LFF are of interest. In works such as [42, 40, 12, 35] Kleene type theorems and axiomatizations of the behaviour of finite systems are studied. Completeness of an axiomatization is then established by proving that expressions modulo axioms form the rational fixpoint. It is an interesting question whether the theory of the LFF we presented here may be of help as a tool for syntactic descriptions and axiomatizations of further system types.

As we have mentioned already the rational fixpoint is the starting point for the coalgebraic study of iterative and iteration theories. A similar path could be followed based on the LFF and this should lead to new coalgebraic iteration/recursion principles, in particular in instances such as context-free languages or constructively S-algebraic formal power series.

Another approach to more powerful recursive definition principles are abstract operational rules (see [27] for an overview). It has been shown that certain rule formats define operations on the rational fixpoint [13, 32], and it should be investigated whether a similar theory can be developed based on the LFF.

Finally, in the special setting of Eilenberg-Moore categories one could base the study of finite systems on *free* finitely generated algebras (rather than all fp or all fg algebras). Does this give a third fixpoint capturing behaviour of finite state systems with side effects besides the rational fixpoint and the LFF? And what is then the relation between the three fixpoints? Also the parallelism in the technical development between rational fixpoint and LFF indicates that there should be a general theory that is parametric in a class of "finite objects" and that allows to obtain results about the rational fixpoint, the LFF and other possible "finite behaviour domains" as instances.

References

1. Aczel, P., Adámek, J., Milius, S., Velebil, J.: Infinite trees and completely iterative theories: A coalgebraic view. Theoret. Comput. Sci. 300, 1–45 (2003)
2. Adámek, J., Milius, S., Velebil, J.: Semantics of higher-order recursion schemes. Log. Methods Comput. Sci. 7(1) (2011)
3. Adámek, J., Milius, S., Velebil, J.: Equational properties of iterative monads. Inform. and Comput. 208, 1306–1348 (2010)
4. Adámek, J., Milius, S., Velebil, J.: Elgot theories: a new perspective of the equational properties of iteration. Math. Structures Comput. Sci. 21(2), 417–480 (2011)
5. Adámek, J.: Free algebras and automata realizations in the language of categories. Comment. Math. Univ. Carolin. 015(4), 589–602 (1974)
6. Adámek, J., Milius, S., Velebil, J.: Iterative Algebras at Work. Math. Structures Comput. Sci. 16(6), 1085–1131 (2006)
7. Adámek, J., Milius, S., Velebil, J.: On second-order iterative monads. Theoret. Comput. Sci. 412(38), 4969–4988 (2011)
8. Adámek, J., Rosický, J.: Locally Presentable and Accessible Categories. Cambridge University Press (1994)

9. Barr, M.: Coequalizers and free triples. Math. Z. 116, 307–322 (1970)
10. Bartels, F.: On generalized coinduction and probabilistic specification formats: Distributive laws in coalgebraic modelling. Ph.D. thesis, Vrije Universiteit Amsterdam (2004)
11. Bloom, S.L., Ésik, Z.: Iteration Theories: the equational logic of iterative processes. EATCS Monographs on Theoretical Computer Science, Springer (1993)
12. Bonsangue, M., Milius, S., Silva, A.: Sound and complete axiomatizations of coalgebraic language equivalence. ACM Trans. Comput. Log. 14(1:7), 52 pp. (2013)
13. Bonsangue, M.M., Milius, S., Rot, J.: On the specification of operations on the rational behaviour of systems. In: Luttik, B., Reniers, M.A. (eds.) Proc. Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS'12). Electron. Proc. Theoret. Comput. Sci., vol. 89, pp. 3–18 (2012)
14. Borceux, F.: Handbook of Categorical Algebra: Volume 1, Basic Category Theory. Encyclopedia of Mathematics and its Applications, Cambridge University Press (1994)
15. Cenciarelli, P., Moggi, E.: A syntactic approach to modularity in denotational semantic. In: Proc. 5th CTCS. CWI Technical Report (1993)
16. Courcelle, B.: Fundamental properties of infinite trees. Theoret. Comput. Sci. 25, 95–169 (1983)
17. Droste, M., Kuich, W., Vogler, H.: Handbook of Weighted Automata. Springer Publishing Company, Incorporated, 1st edn. (2009)
18. Elgot, C.: Monadic computation and iterative algebraic theories. In: Rose, H.E., Sheperdson, J.C. (eds.) Logic Colloquium '73. vol. 80, pp. 175–230. North-Holland Publishers, Amsterdam (1975)
19. Fliess, M.: Sur divers produits de séries formelles. Bulletin de la Société Mathématique de France 102, 181–191 (1974)
20. Gabriel, P., Ulmer, F.: Lokal präsentierbare Kategorien, Lecture Notes Math., vol. 221. Springer-Verlag (1971)
21. Ghani, N., Lüth, C., Marchi, F.D.: Monads of coalgebras: Rational terms and term graphs. Math. Structures Comput. Sci. 15, 433–451 (2005)
22. Goncharov, S.: Trace semantics via generic observations. In: Heckel, R., Milius, S. (eds.) Algebra and Coalgebra in Computer Science (CALCO 2013). Lecture Notes Comput. Sci., vol. 8089, pp. 158–174. Springer (2013)
23. Goncharov, S., Milius, S., Silva, A.: Towards a coalgebraic Chomsky hierarchy. In: Diaz, J., Lanese, I., Sangiorgi, D. (eds.) Proc. 8th IFIP TC 1/WG 2.2 International Conference on Theoretical Computer Science (TCS'14). Lecture Notes Comput. Sci., vol. 8705, pp. 265–280. Springer (2014)
24. Harrison, M.A., Havel, I.M.: Real-time strict deterministic languages. SIAM J. Comput. 1(4), 333–349 (1972)
25. Hyland, M., Plotkin, G., Power, J.: Combining effects: Sum and tensor. Theoret. Comput. Sci. 357(1–3), 70–99 (2006)
26. Jacobs, B.: A bialgebraic review of regular expressions, deterministic automata and languages. In: et al., K.F. (ed.) Goguen Festschrift. Lecture Notes Comput. Sci., vol. 4060, pp. 375–404 (2006)
27. Klin, B.: Bialgebras for structural operational semantics: An introduction. Theoret. Comput. Sci. 412(38), 5043–5069 (2011)
28. Lambek, J.: A fixpoint theorem for complete categories. Math. Z. 103, 151–161 (1968)
29. MacLane, S.: Categories for the Working Mathematician. Graduate Texts in Mathematics, Springer New York, 2nd edn. (1998)
30. Milius, S.: Completely iterative algebras and completely iterative monads. Inform. and Comput. 196, 1–41 (2005)
31. Milius, S.: A Sound and Complete Calculus for finite Stream Circuits. In: Proc. 25th Annual Symposium on Logic in Computer Science (LICS'10). pp. 449–458 (2010)

32. Milius, S., Bonsangue, M.M., Myers, R.S., Rot, J.: Rational operation models. In: Mislove, M. (ed.) Proc. 29th conference on Mathematical Foundations of Programming Science (MFPS XXIX). Electron. Notes Theor. Comput. Sci., vol. 298, pp. 257–282 (2013)
33. Milius, S., Moss, L.S.: The category theoretic solution of recursive program schemes. Theoret. Comput. Sci. 366, 3–59 (2006)
34. Milius, S., Wißmann, T.: Finitary corecursion for the infinitary lambda calculus. In: Proc. 6th Conference on Algebra and Coalgebra in Computer Science, CALCO 2015. Leibniz International Proceedings in Informatics (2015)
35. Myers, R.: Rational coalgebraic machines in varieties: Languages, completeness and automatic proofs. Ph.D. thesis, Imperial College London, Department of Computing (2011)
36. Petre, I., Salomaa, A.: Algebraic systems and pushdown automata. In: Handbook of Weighted Automata, pp. 257–289. Springer (2009)
37. Plotkin, G., Turi, D.: Towards a mathematical operational semantics. In: Proc. 12th LICS Conf. pp. 280–291. IEEE, Computer Society Press (1997)
38. Rutten, J.: Automata and coinduction (an exercise in coalgebra). In: Sangiorgi, D., de Simone, R. (eds.) Proc. CONCUR 1998. LNCS, vol. 1466, pp. 194–218. Springer (1998)
39. Rutten, J.: Universal Coalgebra: A theory of systems. Theor. Comp. Sci. 249(1), 3–80 (2000)
40. Silva, A., Bonchi, F., Bonsangue, M.M., Rutten, J.J.M.M.: Quantitative Kleene Coalgebras. Inform. and Comput. 209(5), 822–849 (2011)
41. Silva, A., Bonchi, F., Bonsangue, M.M., Rutten, J.J.M.M.: Generalizing determinization from automata to coalgebras. Logical Methods in Computer Science 9(1) (2013)
42. Silva, A., Bonsangue, M.M., Rutten, J.J.M.M.: Non-deterministic Kleene coalgebras. Log. Methods Comput. Sci. 6(3:23), 39 pp. (2010)
43. Winter, J., Bonsangue, M., Rutten, J.: Coalgebraic Characterizations of Context-Free Languages. Log. Methods Comput. Sci. 9(3) (September 2013)
44. Winter, J., Bonsangue, M.M., Rutten, J.J.: Context-free coalgebras. J. Comput. System Sci. 81(5), 911 – 939 (2015)
45. Wißmann, T.: The Locally Finite Fixpoint and its properties. Master's thesis, Friedrich-Alexander Universität Erlangen-Nürnberg (April 2015), available at `http://thorsten-wissmann.de/theses/ma-wissmann.pdf`

A Omitted Proofs and Results

Technical Lemmas for Proposition 3.4

We first show directed unions of fg-carried coalgebras are lfg.

Lemma A.1. *Every directed union of coalgebras from* $\mathsf{Coalg}_{\mathsf{fg}}H$ *is an lfg coalgebra.*

Proof. Let $D : (I, \leq) \to \mathsf{Coalg}H, (D_i, d_i) := Di$ be a directed diagram of coalgebras from $\mathsf{Coalg}_{\mathsf{fg}}H$ and of mono-carried morphisms. Name the colimit cocone $c_i : (D_i, d_i) \to (A, a)$ in $\mathsf{Coalg}H$. To check Definition 3.2, let S be a finitely generated object with $f : S \to A$ in $\mathcal{C}$. As colimits in $\mathsf{Coalg}H$ are created by the forgetful functor $U : \mathsf{Coalg}H \to \mathcal{C}$, and because $U \cdot D$ is a directed diagram of monos and S is an fg object, we obtain some factorization as shown below:

$$\begin{tikzcd} S \arrow[rr, "f"] \arrow[dr, "f'"'] & & U(A,a) = A \\ & UDi = D_i \arrow[ur, "Uc_i"'] & \end{tikzcd}$$

Note that because U creates the colimits, we know that the colimit injection for UDi in $\mathcal{C}$ is precisely Uc_i. ☐

Next follow two easy technical lemmas on directed colimits.

Lemma A.2. *For a directed diagram* $D : \mathcal{D} \to \mathcal{C}$ *of subobjects* $m_i : C_i \rightarrowtail C$ *of* C*, the colimit* $(d_i : C_i \to \operatorname{colim} D)_{i \in \mathcal{D}}$ *is obtained by taking the (strong epi,mono)-factorization of* $\coprod C_i \xrightarrow{[m_i]} C$.

Proof. At first, the $(m_i)_{i\in\mathcal{D}}$ form a cocone, so we have a unique $m : \operatorname{colim} D \to C$ with $m \cdot d_i = m_i$, and d_i is monic. As $\mathcal{C}$ is lfp and both d_i and m_i are monic, [8, Proposition 1.62 (ii)] gives us that m is monic, too. The copair of a family of jointly strongly epic family $[d_i] : \coprod C_i \to \operatorname{colim} D$ is a strong epi and therefore we have the factorization:

$$\begin{tikzcd} \coprod C_i \arrow[rr, "{[m_i]}"] \arrow[dr, two heads, "{[d_i]}"'] & & C \\ & \operatorname{colim} D \arrow[ur, tail, "m"'] & \end{tikzcd}$$

☐

Lemma A.3. *Images of colimits in* $\mathsf{Coalg}H$ *are directed unions of images. More precisely, for a diagram* $D : \mathcal{D} \to \mathsf{Coalg}H$*, given a colimit cocone* $(c_i : Di \to C)_{i\in\mathcal{D}}$ *and a morphism* $f : C \to B$*, define* A_i *as* $\mathsf{Im}(f \cdot c_i)$*. Then* $\mathsf{Im}(f)$ *is the directed union of the* A_i *together with the induced monomorphisms:*

$$\begin{tikzcd} Di \arrow[rr, two heads, "e_i"] \arrow[d, "c_i"'] & & A_i \arrow[d, tail, "m_i"] \arrow[dl, dashed, "d_i"'] \\ C \arrow[r, two heads, "e"] \arrow[rr, bend right, "f"'] & \mathsf{Im}(f) \arrow[r, tail, "m"] & B \end{tikzcd} \tag{4}$$

Proof. As colimits in $\mathsf{Coalg}H$ are created by the forgetful functor $U : \mathsf{Coalg}H \to \mathcal{C}$, we consider only the objects first. Take the (strong epi-carried,mono-carried)-factorizations $f \cdot c_i = m_i \cdot e_i$ for each $i \in \mathcal{D}$, and $f = m \cdot e$. Then (4) where d_i is induced by the strong epi e_i. Notice that by $m \cdot d_i = m_i$, d_i is a mono as well. For any morphism $g : Di \to Dj$ we get a mono in $\bar{g} : A_i \rightarrowtail A_j$ by the strong epi e_i:

$$\begin{tikzcd}
Di \arrow[r, two heads, "e_i"] \arrow[d, "g"'] & A_i \arrow[d, tail, dashed, "\bar{g}"'] \arrow[dr, tail, "d_i"] & \\
Dj \arrow[r, two heads, "e_j"] & A_j \arrow[r, tail, "d_j"'] & \mathsf{Im}(f)
\end{tikzcd}$$

By $d_j \cdot \bar{g} = d_i$, we know that $\bar{g}$ is a mono as well. The d_i also ensure that between each pair of objects A_i, A_j there is at most one morphism. With this relation to the D_i, we also inherit the existence of upper bounds in A_i, which can be summarized in: the A_i form a directed diagram of monos in $\mathcal{C}$, i.e. a directed union in $\mathsf{Coalg}H$.

To see that $\mathsf{Im}(f)$ is indeed its colimit, consider

$$\begin{tikzcd}
\coprod_i Di \arrow[r, two heads, "{[c_i]}"] \arrow[d, two heads, "\coprod e_i"'] & C \arrow[d, two heads, "e"] \\
\coprod A_i \arrow[r, "{[d_i]}"'] & \mathsf{Im}(f)
\end{tikzcd}$$

which commutes, because (4) did for every $i \in \mathcal{D}$. The copair of strong epis $[c_i]$ itself is a strong epi and so $e \cdot [c_i]$ and $[d_i] \cdot \coprod e_i$ as well. So $[d_i]$ is a strong epi and $[m_i]$ factors into m and $[d_i]$, and by Lemma A.2 $\mathsf{Im}(f)$, is the colimit.

$$\begin{tikzcd}
\coprod A_i \arrow[r, two heads, "{[d_i]}"] \arrow[rr, bend left, "{[m_i]}"] & \mathsf{Im}(f) \arrow[r, tail, "m"] & B
\end{tikzcd}$$

□

Proof of Proposition 3.4

Proof. Let $c_i : (X_i, x_i) \to (X, x)$ be a colimit cocone of a filtered diagram with (X_i, x_i) from $\mathsf{Coalg}_{\mathsf{fg}}H$. Take the (strong epi,mono)-factorizations

$$c_i \equiv (\ X_i \xrightarrow{e_i} T_i \xrightarrow{m_i} X\ )$$

to get the subcoalgebras (T_i, t_i) of (X, x). By Lemma A.3 with $f = \mathrm{id}_X : X \to X$, $\mathsf{Im}(f) = X$ is the directed union of the T_i. These T_i are in $\mathsf{Coalg}_{\mathsf{fg}}H$ since strong quotients of finitely generated objects are finitely generated. This diagram of the T_i is a directed union with colimit (X, x), both in $\mathcal{B}$ and in $\mathsf{Coalg}H$, so according to Lemma A.1, (X, x) is lfg.

Proof of Proposition 3.7

Proof. The direction from left to right is clear, as $\mathsf{Coalg}_{\mathsf{fg}} \subseteq \mathsf{Coalg}_{\mathsf{lfg}}$. For the other one, let (S, s) be some lfg coalgebra. By Proposition 3.5, it is the directed union of all its subcoalgebras with finitely generated carrier. For each subcoalgebra $\mathsf{in}_p : (P, p) \to (S, s)$, there is a unique homomorphism $p^\dagger : (P, p) \to (L, \ell)$. By the uniqueness of $p^\dagger$ it follows that L together with the $p^\dagger$ is a cocone. Hence there is a unique morphism $\exists! u : (S, s) \to (L, \ell)$ with $u \cdot \mathsf{in}_p = p^\dagger$ for each appropriate (P, p). For any other morphism $\bar{u} : (S, s) \to (L, \ell)$ the equation $\bar{u} \cdot \mathsf{in}_p = p^\dagger$ must hold as well, because $p^\dagger$ is unique. As the in_p are jointly epic, one gets $\bar{u} = u$. □

Proof of Lemma 3.9

Proof. Consider some strong quotient $q : (X, x) \to (Y, y)$ where (X, x) is lfg. As (X, x) is the directed colimit of its subcoalgebras with fg carrier, we have that (Y, y) – the codomain of the strong epi-carried q – is the union of the images of these subcoalgebras by Lemma A.3. The images themselves have a finitely generated carrier – more precisely the factorization in $\mathsf{Coalg}H$ exist because H preserves monos, by factorization. So (Y, y) is the union of these images and thus is lfg. □

Technical Lemmas for Proposition 3.16

The first task is to show that $\bar{e}$ is lfg. So essentially for each $f : S \to X + \vartheta H$ where f is fg we have to find a coalgebra through which f factors, as required by Definition 3.2. Roughly this is done in two steps: firstly we construct the fg image of e in ϑH, secondly the fg image of f in ϑH, for the union P of these images, we construct a coalgebra structure on $X + P$ through which f factors. In order to get this kind of image factorization of f and e from the property of X being finitely generated, ϑH has to be expressed as a directed colimit of monos. This is done with the following lemmas before going into the detail of the proof of the theorem.

Lemma A.4. *Let* $\mathsf{Coalg}'_{\mathsf{fg}}$ *be the full subdiagram of* $\mathsf{Coalg}_{\mathsf{fg}}$ *consisting of those coalgebras* (A, a) *where* $a^\dagger : A \to \vartheta H$ *is a monomorphism. Then the forgetful functor* $U' : \mathsf{Coalg}'_{\mathsf{fg}} \to \mathcal{C}$ *is a directed diagram of monos and filtered.*

Proof. At first, let us show that

$$\text{for } (A, a) \text{ in } \mathsf{Coalg}_{\mathsf{fg}} \text{ there exists } (A', a') \text{ in } \mathsf{Coalg}'_{\mathsf{fg}} \text{ with } h : (A, a) \to (A', a'). \tag{5}$$

This follows directly from the (strong epi,mono) factorization which lifts from $\mathcal{C}$ to $\mathsf{Coalg}_{\mathsf{fg}}$. So $a^\dagger : A \to \vartheta H$ factors into $h : A \twoheadrightarrow A'$ and $a'^\dagger : A' \rightarrowtail \vartheta H$. The strong epi h induces the structure $a' : A' \to HA'$ and proves that both h and $a'^\dagger$ are coalgebra homomorphisms. For the existence of upper bounds, which is required by the directedness, observe that coproducts exists in $\mathsf{Coalg}_{\mathsf{fg}}$, inducing upper bounds in $\mathsf{Coalg}'_{\mathsf{fg}}$ by (5).

For any homomorphisms $g, h : (A_1, a_1) \to (A_2, a_2)$ we have $a_2^\dagger \cdot g = a_1^\dagger = a_2^\dagger \cdot h$. As $a_2^\dagger$ is monic, $g = h$, i.e. there is at most one arrow in each hom set of $\mathsf{Coalg}'_{\mathsf{fg}}$, which means that U' is essentially small, a poset, and thus directed. As $a_1^\dagger$ is a mono, h is a mono as well, so U' is a directed diagram of monos. □

Lemma A.5. *ϑH is the colimit of $U' : \mathsf{Coalg}'_{\mathsf{fg}} H \to \mathcal{C}$.*

Proof. As (5) proves, the inclusion functor $V : \mathsf{Coalg}'_{\mathsf{fg}} H \to \mathsf{Coalg}_{\mathsf{fg}} H$ is a cofinal subdiagram. ϑH is the colimit of the forgetful functor $U : \mathsf{Coalg}_{\mathsf{fg}} H \to \mathcal{C}$, so $\operatorname{colim} U = \operatorname{colim} UV = \operatorname{colim} U'$. □

Proof of Proposition 3.16

Proof. Let $e : X \to HX + \vartheta H$ be an equation morphism with X fg. In the following we prove that e has a unique solution in ϑH. The codomain $HX + \vartheta H$ is the colimit of the following directed diagram of monos:

- The diagram scheme $\mathcal{D}$ is the product category containing pairs $(T \overset{t}{\rightarrowtail} HX, V \xrightarrow{v} HV)$ consisting of an fg subobject of HX and $(V, v) \in \mathsf{Coalg}'_{\mathsf{fg}} H$. $\mathcal{D}$ is directed, because both the fg subobjects of HX and $\mathsf{Coalg}'_{\mathsf{fg}} H$ are.
- The diagram $D : \mathcal{D} \to \mathcal{C}$ is defined by

$$D(t, v) = \mathsf{Im}(t + v^\dagger : T + V \to HX + \vartheta H)$$

  on objects and by diagonalizaton on morphisms. By mono laws, all connecting morphisms are monic.

That $HX + \vartheta H$ is indeed the colimit of D follows from Lemma A.3 applied with $f = \mathrm{id}$. Because X is fg, the morphism e factors through one of the colimit injections, i.e. we obtain an $m : W \rightarrowtail HX + \vartheta H$, W fg, and e such that $m \cdot e' = e$. Furthermore, choose some $t : T \rightarrowtail HX$ and $v : V \to HV$ from $\mathcal{D}$ such that $W = D(t, v)$ as shown in the diagram below:

$$\begin{tikzcd}
X \arrow[r, "e"] \arrow[dr, dashed, "e'"'] & HX + \vartheta H \\
& W \arrow[u, tail, "m"] \\
& T + V \arrow[u, two heads, "{[e_T, e_V]}"] \arrow[uu, bend right=60, "t+v^\dagger"']
\end{tikzcd}$$

Since $T + V$ is fg, so is its strong quotient W. The intermediate object W carries a coalgebra structure by diagonalization:

$$\begin{tikzcd}
& HX + \vartheta H \arrow[rr, "{[He, H\mathsf{inr}\cdot\ell]}"] & & H(HX + \vartheta H) & \\
W \arrow[ur, tail, "m"] & & & & HW \arrow[ul, tail, "Hm"'] \\
& T + V \arrow[ul, two heads, "{[e_T, e_V]}"] \arrow[uu, "t+v^\dagger"] \arrow[rr, "t+v"'] & & HX + HV \arrow[uu, "{[He, H\mathsf{inr}\cdot Hv^\dagger]}"] \arrow[ur, "{[He', He_V]}"'] &
\end{tikzcd}$$

The inner square commutes on the left component trivially, and on the right component because $v^\dagger$ is a H-coalgebra homomorphism, and the two triangles by the previous diagram. This induces a morphism $w : W \to HW$ making m and e_V coalgebra homomorphisms. Since m is independent of the choice of t and v and since Hm is monic, w is independent of the choice of t and v. We have the following commuting diagram:

$$
\begin{tikzcd}
W \arrow[r, tail, "m"] & HX+\vartheta H \arrow[rr, "He'+\ell"] \arrow[dr, phantom, "\text{$v^\dagger$ coalgebra homomorphism}"] & & HW+H\vartheta H \arrow[ddl, bend left, "{[Hw^\dagger, H\vartheta H]}"] \\
T+V \arrow[u, two heads, "{[e_T,e_V]}"] \arrow[ur, "t+v^\dagger"] \arrow[r, "t+v"] \arrow[dd, two heads, "{[e_T,e_V]}"'] \arrow[dr, phantom, "\text{Definition of } w"] & HX+HV \arrow[urr, "He'+Hv^\dagger"] \arrow[r, "He'+He_V"] \arrow[d, "{[He',He_V]}"] & HW+HW \arrow[ur, "HW+Hw^\dagger"] \arrow[d, "{[Hw^\dagger,Hw^\dagger]}"] \arrow[u, phantom, "\text{finality of } \vartheta H"] & \\
 & HW \arrow[r, "Hw^\dagger"] \arrow[d, phantom, "\text{$w^\dagger$ coalgebra homomorphism}"] & H\vartheta H \arrow[dl, bend left, "\ell^{-1}"] & \\
W \arrow[ur, "w"] \arrow[r, "w^\dagger"'] & \vartheta H & &
\end{tikzcd}
$$

Since $[e_T, e_V]$ is an epimorphism, we therefore have

$$w^\dagger = \ell^{-1} \cdot [Hw^\dagger \cdot He', \ell] \cdot m = [\ell^{-1} \cdot Hw^\dagger \cdot He', \mathrm{id}_{\vartheta H}] \cdot m$$

and so $w^\dagger \cdot e'$ is a solution of e in $(\vartheta H, \ell^{-1})$:

$$
\begin{tikzcd}
X \arrow[r, "e'"] \arrow[d, "e"'] & W \arrow[r, "w^\dagger"] \arrow[dl, "m"'] & \vartheta H \\
HX+\vartheta H \arrow[r, "He'+\vartheta H"] & HW+\vartheta H \arrow[r, "Hw^\dagger+\vartheta H"] & H\vartheta H+\vartheta H \arrow[u, "{[\ell^{-1},\vartheta H]}"']
\end{tikzcd}
$$

To verify that this solution is unique, let $s : X \to \vartheta H$ be any solution of e, i.e. we have

$$s = [\ell^{-1} \cdot Hs, \mathrm{id}_{\vartheta H}] \cdot e. \tag{6}$$

This defines a coalgebra homomorphism from (W, w) to ϑH:

$$
\begin{tikzcd}
W \arrow[r, "m"] \arrow[d, "w"'] \arrow[dr, phantom, "\text{Definition of } w"] & HX+\vartheta H \arrow[r, "{[\ell^{-1}\cdot Hs,\vartheta H]}"] \arrow[d, "{[He,H\mathsf{inr}\cdot\ell]}"'] \arrow[dr, "{[Hs,\ell]}"] \arrow[dr, phantom, "(6)"'] & \vartheta H \arrow[d, "\ell"] \\
HW \arrow[r, "Hm"] & H(HX+\vartheta H) \arrow[r, "{H[\ell^{-1}\cdot Hs,\vartheta H]}"] & H\vartheta H
\end{tikzcd}
$$

Hence $[\ell^{-1} \cdot Hs, \mathrm{id}_{\vartheta H}] \cdot m = w^\dagger$ and so

$$w^\dagger \cdot e' = [\ell^{-1} \cdot Hs, \mathrm{id}_{\vartheta H}] \cdot m \cdot e' = [\ell^{-1} \cdot Hs, \mathrm{id}_{\vartheta H}] \cdot e, = s$$

which completes the proof. □

Technical Lemma for Theorem 3.17

Lemma A.6. *For an fg-iterative algebra* $(A, \alpha : HA \to A)$ *and a coalgebra* $e : X \to HX$ *from* $\mathsf{Coalg}_{\mathsf{fg}}$ *there is a unique* $\mathcal{C}$*-morphism* $u_e : X \to A$ *such that* $u_e = \alpha \cdot Hu_e \cdot e$.

$$\begin{tikzcd} X \arrow[r, dashed, "\exists! u_e"] \arrow[d, "e"'] \arrow[dr, phantom, "\circlearrowleft"] & A \\ HX \arrow[r, "Hu_e"] & HA \arrow[u, "\alpha"'] \end{tikzcd}$$

Proof. Consider the equation morphism $\mathsf{inl} \cdot e : X \to HX + A$. For an arbitrary morphism $s : X \to A$, consider the following diagram:

$$\begin{tikzcd} X \arrow[rr, "s"] \arrow[d, "e"] & & A \\ HX \arrow[r, "\mathsf{inl}"] \arrow[rr, bend right, "Hs"'] \arrow[dr, phantom, "\circlearrowleft"] & HX + A \arrow[r, "Hs+A"] & HA + A \arrow[u, "{[\alpha, A]}"'] \arrow[r, phantom, "\circlearrowleft"] & {} \\ & {} & HA \arrow[u, "\mathsf{inl}"'] \arrow[uu, bend right=90, "\alpha"'] \end{tikzcd}$$

The lower part and the right-hand part always commute. But for the commutativity of the whole diagram consider the following sequence of equivalences:

s is a solution of $\mathsf{inl} \cdot e$ in A.
$\Leftrightarrow$ The upper square commutes.
$\Leftrightarrow$ $s = [\alpha, \mathrm{id}_A] \cdot \mathsf{inl} \cdot Hs \cdot e$
$\Leftrightarrow$ $s = \alpha \cdot Hs \cdot e$

So by the existence and the uniqueness of a solution of $\mathsf{inl} \cdot e$ in the fg-iterative algebra A, we get the desired morphism $u_e : X \to A$ with $u_e = \alpha \cdot Hu_e \cdot e$ and its uniqueness, by reading the equivalences from top or from bottom respectively. □

Proof of Theorem 3.17

Proof. By Proposition 3.5, $e : X \to HX$ is the union of the diagram D of its subcoalgebras $s : S \to HS$ with S finitely generated. Denote the corresponding colimit injections by $\mathsf{in}_s : (S, s) \to (X, e)$. Each such s induces a unique morphism $u_s : S \to A$ with

$$u_s = \alpha \cdot Hu_s \cdot s. \tag{7}$$

For any coalgebra homomorphism $h : (R, r) \to (S, s)$ in $\mathsf{Coalg}_{\mathsf{fg}}$ the diagram

$$\begin{tikzcd} R \arrow[r, "h"] \arrow[d, "r"'] & S \arrow[r, "u_s"] \arrow[d, "s"'] & A \\ HR \arrow[r, "Hh"'] & HS \arrow[r, "Hu_s"'] & HA \arrow[u, "\alpha"'] \end{tikzcd}$$

commutes, because h is a coalgebra homomorphism and because of the property of u_s. So $u_r = u_s \cdot h$. In other words, A together with the morphisms $(u_s : S \to A)_{s:S\to HS \text{ lfg}}$ form a cocone for D in $\mathcal{C}$. This induces a unique morphism $u_e : X \to A$.

For each $s : S \to HS$, $\mathsf{in}_s : S \to X$ is a coalgebra homomorphism. Furthermore, we have is $u_s = u_e \cdot \mathsf{in}_s$ in $\mathcal{C}$ by the universal property of X. So every part except possibly (ii) of the diagram

$$\begin{tikzcd}
S \arrow[r, "\mathsf{in}_s"] \arrow[d, "s"'] \arrow[rr, bend left, "u_s"] \arrow[dr, phantom, "\text{(i)}\circlearrowleft"] & X \arrow[r, "u_e"] \arrow[d, "e"'] \arrow[dr, phantom, "\text{(ii)}"] & A \\
HS \arrow[r, "H\mathsf{in}_s"'] \arrow[rr, bend right, "Hu_s"'] & HX \arrow[r, "Hu_e"'] & HA \arrow[u, "\alpha"']
\end{tikzcd}$$

commutes, as indicated. In particular the outer square square commutes which gives

$$\alpha \cdot Hu_e \cdot e \cdot \mathsf{in}_s = u_e \cdot \mathsf{in}_s \text{ for every fg subcoalgebra } (S, s) \text{ of } (X, e).$$

As the colimit injections in_s are jointly epic, (ii) commutes.

Conversely every $\mathcal{C}$-morphism $\tilde{u}_e : X \to A$ making (ii) commute, makes the bigger square (i)+(ii) commute and defines a family of morphisms $\tilde{u}_e \cdot \mathsf{in}_s : S \to A$ having the property (7) each. So by the uniqueness of the $u_s : S \to A$, we get $u_s = \tilde{u}_e \cdot \mathsf{in}_s$. Using again that the in_s are jointly epic, reduces the equation

$$u_e \cdot \mathsf{in}_s = u_s = \tilde{u}_e \cdot \mathsf{in}_s$$

to the desired uniqueness of u_e, namely $u_e = \tilde{u}_e$. □

Proof of Proposition 3.21

Proof. Take a coalgebra (X, x) with finitely generated carrier, which is the strong quotient of some fp object X' via $q : X' \twoheadrightarrow X$. By assumption, X' is the strong quotient of a projective fp object X'' via $q' : X'' \to X'$. As H preserves strong epis, the projectivity of X'' induces the coalgebra structure x'':

$$\begin{tikzcd}
X'' \arrow[r, dashed, "x''"] \arrow[d, two heads, "q'"'] & HX'' \arrow[d, two heads, "Hq'"] \\
X' \arrow[d, two heads, "q"'] & HX' \arrow[d, two heads, "Hq"] \\
X \arrow[r, "x"] & HX
\end{tikzcd}$$

□

Proof of Proposition 4.3

Proof. First of all, $(\vartheta H^T, \ell)$ is final for all $(TX, x^\sharp)$, with X finite, so it is a competing cocone for (K, k):

$$\begin{tikzcd}
(TX, x^\sharp) \arrow[r, "\mathsf{in}_X"] \arrow[dr, "Ux^{\sharp\dagger}"'] & (K, k) \arrow[r, two heads, "e"] \arrow[d, dashed, "w"] \arrow[dr, bend left=60, "k^\dagger"] & (I, i) \arrow[d, tail, "m"] \\
& (U\vartheta H^T, U\ell) \arrow[r, tail, "n"] & (\nu H, \tau)
\end{tikzcd}$$

Hence, w is induced making the triangle commute. Any (G, g) in $\mathsf{Coalg}_{\mathsf{fg}} H^T$ is the quotient of some $(TX, x^\sharp)$. And on the other hand, the $g^\dagger : (G, g) \to (\vartheta H^T, \ell)$ are jointly epic. Hence, the $x^{\sharp\dagger}$ are jointly epic as well, and so the $Ux^{\sharp\dagger}$, too. Hence also w is epic, and – as we are in Set – even a strong epimorphism. In other words, $(U\vartheta H^T, U\ell)$ is the (unique) image of (K, k) in $(\nu H, \tau)$. □

Proof of Theorem 4.4

Proof. Combining the previous Proposition 4.3 together with the Lemma A.3 proves the first equality. For the second equality, consider any element $t \in TX$ and define a new coalgebra on $X + 1$ by

$$(Y, y) \equiv \big(\; X + 1 \xrightarrow{[x, x^\sharp(t)]} HTX \xrightarrow{HT\mathsf{inl}} HT(X+1) \; \big).$$

Clearly, $[\mathrm{id}_{TX}, t] : Y \to X$ is a H^T-coalgebra homomorphism, and $t \in y^{\sharp\dagger} \cdot \eta^T_Y[Y]$. □

Definition of the Lifting of $S \times (-)^{\Sigma}$ to S-algebras

The $S\langle - + \Sigma \rangle$-algebra structure – S-module structure, monoid structure, Σ-pointing – on $S \times A^\Sigma$ can be defined using the $S\langle - + \Sigma \rangle$-structure on A as follows:

Structure	Connective	in S	in A^Σ
S-Module	0	0_S	$a \mapsto 0_A$
	$(o_1, \delta_1) + (o_2, \delta_2)$	$o_1 + o_2$	$a \mapsto \delta_1(a) + \delta_2(a)$
	$s \cdot (o_1, \delta_1)$	$s \cdot o_1$	$a \mapsto s \cdot \delta_1(a)$
Monoid	1	1_S	$a \mapsto 0_A$
	$(o_1, \delta_1) * (o_2, \delta_2)$	$o_1 \cdot o_2$	$a \mapsto \delta_1(a) \cdot [o_2, \delta_2] + i(o_1) \cdot \delta_2(a)$
Σ-pointing	$b \in \Sigma$	0_S	$b \mapsto 1_A, \quad a \mapsto 0_A, b \neq a$

The defined connectives only makes use of connectives from S (seen as a S-algebra) and from the S-algebra A, so H maps any $S\langle - + \Sigma \rangle$-algebra homomorphism $h : A \to B$ to again a homomorphism $Hh : S \times A^\Sigma \to S \times B^\Sigma$. In total, we have a lifting $H^T : \mathsf{Set}^T \to \mathsf{Set}^T$ of H, as soon as we have checked the S-algebra axioms for $HA = S \times A^\Sigma$.

S-algebra connective preserving $[-]: S \times A^{\Sigma} \to A$

In order to show that $S \times A^{\Sigma}$ is indeed an S-algebra, it comes handy to establish some identities for $[-]: S \times A^{\Sigma} \to A$ first, namely, that it preserves the proposed S-algebra structure in the expected manner. It preserves the S-module connectives zero

$$[0_S, a \mapsto 0_A] = i(0_S) + \sum_{b \in \Sigma} j(b) \cdot 0_A = 0_A$$

addition,

$$\begin{aligned}
&[o_1 + o_2, a \mapsto \delta_1(a) + \delta_2(a)] \\
&= i(o_1 + o_2) + \sum_{b \in \Sigma} \big(j(b) \cdot (\delta_1(a) + \delta_2(a))\big) \\
&= i(o_1) + i(o_2) + \sum_{b \in \Sigma} \big(j(b) \cdot \delta_1(a)\big) + \sum_{b \in \Sigma} \big(j(b) \cdot \delta_2(a)\big) \\
&= [o_1, \delta_1] + [o_2, \delta_2]
\end{aligned}$$

and scalar multiplication:

$$\begin{aligned}
[s \cdot o, a \mapsto s \cdot \delta(a)] &= i(s \cdot o) + \sum_{b \in \Sigma} \big(j(b) \cdot (s \cdot \delta(b))\big) = s \cdot i(o) + \sum_{b \in \Sigma} s \cdot \big(j(b) \cdot \delta(b)\big) \\
&= s \cdot \left(i(o) + \sum_{b \in \Sigma} \big(j(b) \cdot \delta(b)\big) \right) = s \cdot [o, \delta]
\end{aligned}$$

The monoid connectives are preserved as well:

$$[1_S, a \mapsto 0_A] = i(1_S) + \sum_{b \in \Sigma} j(b) \cdot 0_A = i(1_S) = 1_A$$

$$\begin{aligned}
[o_1, \delta_1] \cdot [o_2, \delta_2] &= \left(i(o_1) + \sum_{b \in \Sigma} j(b) \cdot \delta_1(b) \right) \cdot [o_2, \delta_2] \\
&= i(o_1) \cdot [o_2, \delta_2] + \sum_{b \in \Sigma} j(b) \cdot \delta_1(b) \cdot [o_2, \delta_2] \\
&= i(o_1) \cdot \left(i(o_2) + \sum_{b \in \Sigma} j(b) \cdot \delta_2(b) \right) + \sum_{b \in \Sigma} j(b) \cdot \delta_1(b) \cdot [o_2, \delta_2] \\
&= i(o_1 \cdot o_2) + \sum_{b \in \Sigma} j(b) \cdot i(o_1) \cdot \delta_2(b) + \sum_{b \in \Sigma} j(b) \cdot \delta_1(b) \cdot [o_2, \delta_2] \\
&= i(o_1 \cdot o_2) + \sum_{b \in \Sigma} j(b) \cdot \big(i(o_1) \cdot \delta_2(b) + \delta_1(b) \cdot [o_2, \delta_2] \big) \\
&= \big[o_1 \cdot o_2, a \mapsto i(o_1) \cdot \delta_2(a) + \delta_1(a) \cdot [o_2, \delta_2] \big] = \big[(o_1, \delta_1) * (o_2, \delta_2) \big]
\end{aligned}$$

S-Algebra axioms

Firstly, note that $(HA, 0, +, \cdot)$ fulfills all the S-Module axioms, because $0, +, \cdot$ are defined point-wise in A. Secondly, $(HA, 1, *)$ is a monoid:

$$\begin{aligned}(1_S, a \mapsto 0_A) * (o, \delta) &= \big(1_S \cdot o, a \mapsto 0_A \cdot [o, \delta] + i(1_S) \cdot \delta(a)\big) = \big(o, a \mapsto \delta(a)\big)\\ (o, \delta) * (1_S, a \mapsto 0_A) &= \big(o \cdot 1_S, a \mapsto \delta(a) \cdot [1_S, a \mapsto 0_A] + i(o) \cdot 0_A\big)\\ &= \big(o, a \mapsto \delta(a) \cdot 1_A + 0_A\big) = (o, \delta)\end{aligned}$$

$$\begin{aligned}&\big((o_1, \delta_1) * (o_2, \delta_2)\big) * (o_3, \delta_3)\\ &= \big(o_1 \cdot o_2, a \mapsto \delta_1(a) \cdot [o_2, \delta_2] + i(o_1) \cdot \delta_2(a)\big) * (o_3, \delta_3)\\ &= \big(o_1 \cdot o_2 \cdot o_3, a \mapsto \big(\delta_1(a) \cdot [o_2, \delta_2] + i(o_1) \cdot \delta_2(a)\big) \cdot [o_3, \delta_3] + i(o_1 \cdot o_2) \cdot \delta_3(a)\big)\\ &= \big(o_1 \cdot o_2 \cdot o_3, a \mapsto \delta_1(a) \cdot [o_2, \delta_2] \cdot [o_3, \delta_3] + i(o_1) \cdot \delta_2(a) \cdot [o_3, \delta_3] + i(o_1 \cdot o_2) \cdot \delta_3(a)\big)\\ &= \big(o_1 \cdot o_2 \cdot o_3, a \mapsto \delta_1(a) \cdot \big[(o_2, \delta_2) * (o_3, \delta_3)\big] + i(o_1) \cdot \big(\delta_2(a) \cdot [o_3, \delta_3] + i(o_2) \cdot \delta_3(a)\big)\big)\\ &= (o_1, \delta_1) * \big(o_2 \cdot o_3, a \mapsto \delta_2(a) \cdot [o_3, \delta_3] + i(o_2) \cdot \delta_3(a)\big)\\ &= (o_1, \delta_1) * \big((o_2, \delta_2) * (o_3, \delta_3)\big)\end{aligned}$$

What remains is the bilinearity of $*$ with respect to the S-Module structure. For bilinearity of $*$ in the first argument, we use the very same properties in A:

$$\begin{aligned}&\big((o_1, \delta_1) + (o_2, \delta_2)\big) * (o_3, \delta_3)\\ &= (o_1 + o_2, a \mapsto \delta_1(a) + \delta_2(a)) * (o_3, \delta_3)\\ &= \big((o_1 + o_2) \cdot o_3, a \mapsto (\delta_1(a) + \delta_2(a)) \cdot [o_3, \delta_3] + i(o_1 + o_2) \cdot \delta_3(a)\big)\\ &= \big(o_1 \cdot o_3 + o_2 \cdot o_3, a \mapsto \delta_1(a) \cdot [o_3, \delta_3] + \delta_2(a) \cdot [o_3, \delta_3] + (i(o_1) + i(o_2)) \cdot \delta_3(a)\big)\\ &= (o_1, \delta_1) * (o_3, \delta_3) + (o_2, \delta_2) * (o_3, \delta_3)\end{aligned}$$

$$\begin{aligned}\big(s \cdot (o_1, \delta_1)\big) * (o_2, \delta_2) &= \big(s \cdot o_1 \cdot o_2, a \mapsto s \cdot \delta_1(a) \cdot [o_2, \delta_2] + i(s \cdot o_1) \cdot \delta_2(a)\big)\\ &= \big(s \cdot o_1 \cdot o_2, a \mapsto s \cdot (\delta_1(a) \cdot [o_2, \delta_2] + i(o_1) \cdot \delta_2(a))\big)\\ &= s \cdot \big((o_1, \delta_1) * (o_2, \delta_2)\big)\end{aligned}$$

$$\begin{aligned}(0_S, a \mapsto 0_A) * (o, \delta) &= \big(0_S \cdot o, a \mapsto 0_A \cdot [o, \delta] + i(0_S) \cdot \delta(a)\big)\\ &= \big(0_S \cdot o, a \mapsto 0_A \cdot [o, \delta] + 0_A \cdot \delta(a)\big) = \big(0_S, a \mapsto 0_A\big)\end{aligned}$$

Finally, linearity in the second argument of $*$ using the identities for $[-]$:

$$\begin{aligned}&(o_1, \delta_1) * \big((o_2, \delta_2) + (o_3, \delta_3)\big)\\ &= (o_1 \cdot (o_2 + o_3), a \mapsto \delta_1(a) \cdot [o_2 + o_3, a \mapsto \delta_2(a) + \delta_3(a)] + i(o_1) \cdot (\delta_2(a) + \delta_3(a))\\ &= (o_1 \cdot o_2 + o_1 \cdot o_3, a \mapsto \delta_1(a) \cdot [o_2, \delta_2] + \delta_1(a) \cdot [o_3, \delta_3] + i(o_1) \cdot \delta_2(a) + i(o_1) \cdot \delta_3(a)\\ &= (o_1, \delta_1) * (o_2, \delta_2) + (o_1, \delta_1) * (o_3, \delta_3)\end{aligned}$$

$$
\begin{aligned}
&(o_1,\delta_1) * \big(s\cdot(o_2,\delta_2)\big) = (o_1,\delta_1) * (s\cdot o_2, a\mapsto s\cdot\delta_2)\big)\\
=\;& \big(o_1\cdot(s\cdot o_2), \delta_1(a)\cdot[s\cdot o_2, a\mapsto s\cdot\delta_2(a)] + i(o_1)\cdot(s\cdot\delta_2(a))\big)\\
=\;& \big(o_1\cdot(s\cdot o_2), \delta_1(a)\cdot(s\cdot[o_2,\delta_2]) + i(o_1)\cdot(s\cdot\delta_2(a))\big)\\
=\;& \big(s\cdot(o_1\cdot o_2), s\cdot(\delta_1(a)\cdot[o_2,\delta_2]) + s\cdot(i(o_1)\cdot\delta_2(a))\big) = s\cdot\big((o_1,\delta_1)*(o_2,\delta_2)\big)
\end{aligned}
$$

$$
\begin{aligned}
(o,\delta) * (0_S, a\mapsto 0_A) &= \big(o\cdot 0_S, a\mapsto \delta(a)\cdot[0_S, a\mapsto 0_A] + i(o)\cdot 0_A\big)\\
&= \big(o\cdot 0_S, a\mapsto \delta(a)\cdot 0_A + 0_A\big) = (0_S, a\mapsto 0_A)
\end{aligned}
$$

So for any S-algebra A, $S\times A^{\Sigma}$ is an S-algebras too and hence $[-]: S\times A^{\Sigma}\to A$ an S-algebra homomorphism.

Proof of Lemma 4.9

Proof. In other words, let us prove that $R = \{([c(w)], w) \mid w\in A\}$ is a bisimulation. First, take $c = \langle o,\delta\rangle$ in Set (not in Set^T) and note that the following holds for any $b\in\Sigma$ and $v\in A$ (where $\varrho_b:\Sigma\to A$ with $\varrho_b(b)=1$ and $\varrho_b(a)=0$ for $a\neq b$):

$$
\begin{aligned}
c(j(b)) * c(v) = (0_S,\varrho_b) * c(v) &= (0_S,\varrho_b) * \big(o(v),\delta(v)\big)\\
&= \big(0_S\cdot o(v), a\mapsto \varrho_b(a)\cdot[c(v)] + i(0_S)\cdot\delta(v)(a)\big)\\
&= \big(0_S, a\mapsto \varrho_b(a)\cdot[c(v)]\big)
\end{aligned}
$$

The following shows that R is a bisimulation:

$$
\begin{aligned}
c([c(w)]) = c([o(w),\delta(w)]) &= c\left(i(o(w)) + \sum_{b\in\Sigma}\big(j(b)\cdot\delta(w)(b)\big)\right)\\
&= c\big(i(\underbrace{o(w)}_{\in S})\big) + \sum_{b\in\Sigma} c\big(j(b)\big) * c\big(\delta(w)(b)\big)\\
&= (o(w), a\mapsto 0_A) + \sum_{b\in\Sigma}\big(0_S, a\mapsto \varrho_b(a)\cdot[c(\delta(w)(b))]\big)\\
&= (o(w), a\mapsto 0_A) + \left(0_S, a\mapsto \sum_{b\in\Sigma}\varrho_b(a)\cdot[c(\delta(w)(b))]\right)\\
&= (o(w), a\mapsto 0_A) + (0_S, a\mapsto [c(\delta(w)(a))])\\
&= (o(w), a\mapsto [c(\delta(w)(a))])
\end{aligned}
$$

This says that for any $v\in A$, $o([c(v)]) = o(v)$ and for all $a\in\Sigma$

$$\delta([c(v)])(a) = [c(\delta(v)(a))]\; R\; \delta(v)(a).$$

i.e. R is a bisimulation. $\square$

Proof of Lemma 4.10

Proof (By induction on u w.r.t. the connectives of S-algebras). Put $c^\sharp = \langle o^\sharp, \delta^\sharp \rangle$.

- *Base Case:* For any $x \in X$, $x \in S\langle X \rangle$ and $x \in S\langle X + \Sigma \rangle$ are behaviourally equivalent by construction of $\hat{c}$ and $c^\sharp$.
- *Step "S-Module-Structure":* The definition of $\hat{c}$ on S-Module connectives is pointwise [44, Sect. 3], and thus identical to the definition of $c^\sharp$.
- *Step "Monoid-Structure":* The neutral element is mapped by $\hat{c}$ to $(1, a \mapsto 0)$ [44, Sect. 4], and this is identical to the definition $c^\sharp$.
  For polynomials $v, w \in S\langle X \rangle$ and $v', w' \in S\langle X + \Sigma \rangle$, assume that $v \sim v'$, $w \sim w'$ (with $\sim$ denoting behavioural equivalence). We have

  $$\hat{o}(v * w) \overset{(3)}{=} \hat{o}(v) \cdot \hat{o}(w) \overset{\text{IH}}{=} o^\sharp(v') \cdot o^\sharp(w') \overset{(2)}{=} o^\sharp(v' * w').$$

  Note that final homomorphism $\hat{c}^\dagger : S\langle X \rangle \to \nu H$ preserves multiplication by [44, Prop 15] and the final $c^{\sharp\dagger}$ as well, because it lives in Set^T. So for any $x \sim x'$ and $y \sim y'$, $x, y, \in S\langle X \rangle$, $x', y' \in TX$, we have:

  $$\hat{c}^\dagger(x * y) = \hat{c}^\dagger(x) * \hat{c}^\dagger(y) \underset{y \sim y'}{\overset{x \sim x'}{=}} c^{\sharp\dagger}(x') * c^{\sharp\dagger}(y') = c^{\sharp\dagger}(x' * y'),$$

  i.e. $\sim$ is a congruence for $*$ (and also for $+$). The hypothesis $v \sim v'$ implies $\hat{\delta}(v, a) \sim \delta^\sharp(v', a)$. For $a \in \Sigma$,

  $$\begin{aligned}
  \hat{\delta}(v * w, a) &\overset{(3)}{=} \hat{\delta}(v, a) * w + \hat{o}(v) * \hat{\delta}(w, a) \\
  &\overset{\text{IH}}{\sim} \delta^\sharp(v', a) * w' + o^\sharp(v) * \delta^\sharp(w', a) \\
  &\overset{\text{Lemma 4.9}}{\sim} \delta^\sharp(v', a) * [o^\sharp(w'), \delta^\sharp(w')] + o^\sharp(v) * \delta^\sharp(w', a) \\
  &\overset{(2)}{=} \delta^\sharp(v' * w', a).
  \end{aligned}$$

  So $v * w \sim v' * w'$. □

Proof of Lemma 4.12

Proof. Let $D : \mathcal{D} \to \mathsf{Mnd_f}(\mathcal{C})$, $Di = (M_i, \eta^i, \mu^i)$ be a filtered diagram. Take its colimit $M = \operatorname{colim} D$ with injections $\mathsf{in}_i : M_i \to M$ in $\mathsf{Fun_f}(\mathcal{C})$ and define a monad unit by

$$\eta \equiv \left(\mathrm{Id} \xrightarrow{\eta^i} M_i \xrightarrow{\mathsf{in}_i} M\right), \qquad \text{for any } i \in \mathcal{D}.$$

Similarly, define the monad multiplication $\mu : MM \to M$ as the unique natural transformation with

$$\begin{tikzcd}
M_i M_i \arrow[r, "\mu^i"] \arrow[d, "\mathsf{in}_i * \mathsf{in}_i"'] & M_i \arrow[d, "\mathsf{in}_i"] \\
MM \arrow[r, dashed, "\mu"] & M
\end{tikzcd} \qquad \text{for any } i \in \mathcal{D}.$$

The filteredness of D proves the independence of the choice of i: for any other candidate $j \in \mathcal{D}$ choose an upper bound $m_{i,k} : M_i \to M_k \leftarrow M_j : m_{j,k}$ of M_i and M_j. Then we have a commutative diagram

$$\begin{tikzcd}
 & M_i \arrow[d, "m_{i,k}"] \arrow[dr, "\mathsf{in}_i"] & \\
\mathrm{Id} \arrow[ur, "\eta^i"] \arrow[r, "\eta^k"] \arrow[dr, "\eta^j"'] & M_k \arrow[r, "\mathsf{in}_k"] & M \\
 & M_j \arrow[u, "m_{j,k}"'] \arrow[ur, "\mathsf{in}_j"'] &
\end{tikzcd}$$

The left-hand triangles commute because $m_{i,k}, m_{j,k}$ are monad morphisms and the right-hand triangles because $m_{i,k}, m_{j,k}$ are connecting natural transformations of D and the in the colimit injections.

Note that $(M_iM_i)_{i\in\mathcal{D}}$ is a filtered diagram with colimit MM in $\mathsf{Fun_f}(\mathcal{C})$. Let us check the monad laws:

- Unit laws: the diagrams

$$\begin{tikzcd}
M_i \arrow[rr, bend left, equal] \arrow[r, "\eta^i M_i"] \arrow[dd, "\mathsf{in}_i"'] \arrow[dr, phantom, "\text{Naturality of } \eta^i"] & M_iM_i \arrow[r, "\mu^i"] \arrow[d, "M_i\mathsf{in}_i"] \arrow[dr, phantom, "\text{Definition of } \mu"] & M_i \arrow[dd, "\mathsf{in}_i"] \\
 & M_iM \arrow[d, "\mathsf{in}_iM"] & \\
M \arrow[ur, "\eta^iM"] \arrow[r, "\eta M"'] \arrow[r, phantom, "\text{Def. } \eta"] & MM \arrow[r, "\mu"'] & M
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
M_i \arrow[rr, bend left, equal] \arrow[r, "M_i\eta^i"] \arrow[dd, "\mathsf{in}_i"'] \arrow[dr, "M_i\eta"'] & M_iM_i \arrow[r, "\mu^i"] \arrow[d, "M_i\mathsf{in}_i"] \arrow[dl, phantom, "\text{Def. } \eta"] \arrow[dr, phantom, "\text{Definition of } \mu"] & M_i \arrow[dd, "\mathsf{in}_i"] \\
 & M_iM \arrow[d, "\mathsf{in}_iM"] & \\
M \arrow[r, "M\eta"'] & MM \arrow[r, "\mu"'] & M
\end{tikzcd}$$

  commute. As the in_i are jointly epic, (M, η, μ) fulfills the unit laws.
- Associativity:

$$\begin{tikzcd}
M_iM_iM_i \arrow[rrr, "\mu^iM_i"] \arrow[ddd, "M_i\mu^i"'] \arrow[dr, "\mathsf{in}_i*\mathsf{in}_i*\mathsf{in}_i"] & & & M_iM_i \arrow[ddd, "\mu^i"] \arrow[dl, "\mathsf{in}_i*\mathsf{in}_i"'] \\
 & MMM \arrow[r, "\mu M"] \arrow[d, "M\mu"] & MM \arrow[d, "\mu"] & \\
 & MM \arrow[r, "\mu"] & M & \\
M_i \arrow[ur, "\mathsf{in}_i*\mathsf{in}_i"] \arrow[rrr, "\mu^i"'] & & & M_i \arrow[ul, "\mathsf{in}_i"']
\end{tikzcd}$$

  The outside commutes, and by definition of μ also all inner parts (except possibly for the middle square). As the $\mathsf{in}_i * \mathsf{in}_i * \mathsf{in}_i$ are jointly epic, the middle square commutes as well.

By definition of η and μ, each $\mathsf{in}_i : M_i \to M$ is a monad morphism. In fact, η and μ are the unique natural transformations making the diagrams (in the definition) commute, i.e. are the unique monad structure on M such that in_i is a monad morphism.

To see that (M, η, μ) is a colimiting cocone, consider another cocone $n_i : M_i \to N$ in $\mathsf{Mnd_f}(\mathcal{C})$. This induced a unique natural transformation $m : M \to N$ with $n_i = m \cdot \mathsf{in}_i$. To see that m is also a monad morphism, use the jointly epicness of the in_i:

$$m \cdot \eta = m \cdot \mathsf{in}_i \cdot \eta^i = n_i \cdot \eta^i = \eta^N,$$

Consider the following diagram:

$$\begin{tikzcd}
M_iM_i \arrow[rrr, "\mu^i"] \arrow[dr, "\mathsf{in}_i * \mathsf{in}_i"] \arrow[ddr, bend right, "n_i * n_i"'] & & & M_i \arrow[dl, "\mathsf{in}_i"'] \arrow[ddl, bend left, "n_i"] \\
& MM \arrow[r, "\mu"] \arrow[d, "m * m"'] \arrow[dr, phantom, "?"] & M \arrow[d, "m"] & \\
& NN \arrow[r, "\mu^N"] & N &
\end{tikzcd}$$

The outside commutes, because n_i is a monad morphism. The outer triangles commute on the level of $\mathsf{Fun_f}(\mathcal{C})$ and the upper part commutes because in_i is a monad morphism. Again, as the $\mathsf{in}_i * \mathsf{in}_i$ are jointly epic, the inner square commutes as well, hence m is a monad morphism. □

Proof of Lemma 4.14

Proof. Strong epis in slice categories are carried by strong epis, so consider a strong epi $q : A \to B$ in $\mathsf{Mnd_f(Set)}$. Consider the (strong epi,mono)-factorizations of the components in $\mathsf{Fun_f}$:

$$\begin{tikzcd}
M \arrow[r, two heads, "e"] \arrow[rr, bend left, "q"] & I \arrow[r, tail, "m"] & N
\end{tikzcd}$$

The factorization lifts further to $\mathsf{Mnd_f(Set)}$, i.e. we have factorized the monad morphism q into an epi e and a mono m in $\mathsf{Mnd_f(Set)}$. As any strong epi is also extremal, we get that m is an isomorphism. Hence q has epic components. All Set-functors preserve (strong) epis, so $Hq_X + \mathrm{Id}$ is epic for any set X and so the natural transformation $Hq + \mathrm{Id}$ as well.

Proof of Lemma 4.15

Proof. (B, β) is the strong quotient of a $(F^{H+V}, \hat{\kappa} \cdot \mathsf{inl})$, which again is a quotient of $(F^{H+P}, \hat{\kappa} \cdot \mathsf{inl})$, where P a polynomial functor and therefore an epi-projective in $\mathsf{Fun_f}(\mathcal{C})$.

$$\begin{tikzcd}
(F^{H+P}, \hat{\kappa} \cdot \mathsf{inl}) \arrow[r, two heads, "q_P"] \arrow[rr, bend right, two heads, "=:q"'] & (F^{H+V}, \hat{\kappa} \cdot \mathsf{inl}) \arrow[r, two heads, "q_V"] & (B, \beta) \arrow[r, "b"] & \mathcal{H}(B, \beta)
\end{tikzcd}$$

This corresponds to a natural transformation $\overline{b \cdot q} : P \to HB + \mathrm{Id}$. As P is projective and by Lemma 4.14 $\mathcal{H}_{\mathfrak{f}} q$ is epic as a natural transformation, we get a natural transformation $p : P \to HF^{H+P} + \mathrm{Id}$ such that the diagram on the left below commutes:

$$\begin{tikzcd} P \arrow[r, dashed, "p"] \arrow[dr, "\overline{b\cdot q}"'] & HF^{H+P} + \mathrm{Id} \arrow[d, two heads, "Hq+\mathrm{Id}"] \\ & HB + \mathrm{Id} \end{tikzcd} \iff \begin{tikzcd} (F^{H+P}, \hat{\kappa} \cdot \mathsf{inl}) \arrow[r, "\bar{p}"] \arrow[d, two heads, "q"'] & \mathcal{H}_{\mathfrak{f}}(F^{H+P}, \hat{\kappa} \cdot \mathsf{inl}) \arrow[d, "Hq+\mathrm{Id}"] \\ (B, \beta) \arrow[r, "b"] & \mathcal{H}_{\mathfrak{f}}(B, \beta) \end{tikzcd}$$

It follows that the coalgebra b is the strong quotient of the coalgebra $\bar{p}$, which is a coalgebra in EQ.