Context-free Languages via Coalgebraic Trace Semantics

Ichiro Hasuo and Bart Jacobs

Institute for Computing and Information Sciences, Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
E-mail: {ichiro, B.Jacobs}@cs.ru.nl
URL: http://www.cs.ru.nl/{~ichiro, B.Jacobs}

Abstract. We show that, for functors with suitable mild restrictions, the initial algebra in the category of sets and functions gives rise to the final coalgebra in the (Kleisli) category of sets and relations. The finality principle thus obtained leads to the *finite trace semantics* of non-deterministic systems, which extends the *trace semantics* for coalgebras previously introduced by the second author. We demonstrate the use of our technical result by giving the first coalgebraic account on context-free grammars, where we obtain generated context-free languages via the finite trace semantics. Additionally, the constructions of both finite and possibly infinite parse trees are shown to be monads. Hence our extension of the application domain of coalgebras identifies several new mathematical constructions and structures.

1 Introduction

Context-free grammars and context-free languages are undoubtedly among the most fundamental notions in computer science. Introduced by Chomsky [Cho56], they have come to serve as a theoretical basis for formal (programming) languages [ASU86]. This paper presents the first steps in a coalgebraic analysis of those notions. In a sense it extends previous coalgebraic work [Jac05a,Rut03] on regular languages.

A context-free grammar is a clear example of a coalgebra: the state space consists of its non-terminal symbols and the coalgebraic structure is defined by its generation rules. Then the context-free language generated by the grammar should be the "behavior" of the coalgebra. Our motivation is to find a suitable setting which gives that behavior by *coinduction*, i.e. an argument using finality.

What is unusual here is that we are concerned only with the *finite* behavior (i.e. generated strings of only finite length). This suggests that the domain of the semantics might be the initial algebra, as the subcoalgebra of the final coalgebra consisting of all the finite behavior.

Interestingly, it turns out that for functors with mild restrictions, the initial algebra in **Sets** gives rise to the final coalgebra in the category **Rel** of sets and relations. The finality principle in **Rel** is called the *finite trace semantics* in this paper, in contrast to the (possibly infinite) *trace semantics* from [Jac04b] where the final coalgebra in **Sets** gives rise to a weakly final coalgebra in **Rel**.

A context-free grammar is identified as a coalgebra in **Rel** because of its non-deterministic nature. Now our technical result of finite trace semantics allows us to obtain the set of generated *finite skeletal parsed trees* (finite strings with additional tree structure) via finality. After applying the flattening function it yields the set of generated strings.

The category **Rel** can also be described as the Kleisli category $\mathbf{Sets}_{\mathcal{P}}$ of the powerset monad. This view is relevant in generalization of our work to other monads other than $\mathcal{P}$, such as the subdistribution functor $\mathcal{D}$.

The remainder of this paper is organized as follows. Later on this section gives a "sneak preview" of the technical result and its applications. Section 2 formulates context-free grammars as coalgebras, and introduces the notion of *skeletal parse trees (SPTs)* as strings with tree structure. It is shown in Section 3 that (finite) SPTs carry the initial algebra/final coalgebra for an appropriate functor, and that their formations have monad structures, related to one another via the "fundamental span" of monad maps. The details of our technical result of finite trace semantics for coalgebras are presented in Section 4. Section 5 puts the current work in the context of the previous work [Jac04b] of (possibly infinite) trace semantics for coalgebras. Section 6 is for conclusions and future work.

It is assumed that the reader is familiar with the basic categorical theory of algebras and coalgebras for both functors and monads. For these preliminaries see e.g. [Jac05b,Rut00,BW83].

1.1 Sneak preview

As motivation we briefly present our main technical result and two illustrating examples of non-deterministic automata and context-free grammars. The details of constructions, definitions and proofs will follow later. For a functor $F : \mathbf{Sets} \to \mathbf{Sets}$ with mild restrictions, we have an initial algebra in **Sets** and a canonical lifting $F : \mathbf{Rel} \to \mathbf{Rel}$.

Theorem 1.1 (Finite trace semantics for coalgebras) *Let $\alpha : FA \xrightarrow{\cong} A$ be the initial F-algebra in* **Sets**. *The coalgebra $\mathsf{graph}(\alpha^{-1}) : A \xrightarrow{\cong} FA$ in* **Rel** *is final for the lifted functor F. Hence, given a coalgebra $c : X \to FX$ in* **Rel** *there exists a unique arrow $\mathsf{ft}_c : X \to A$ which makes the following diagram in* **Rel** *commute.*

$$\begin{tikzcd} FX \arrow[r, dashed, "F(\mathsf{ft}_c)"] & FA \\ X \arrow[u, "c"] \arrow[r, dashed, "\mathsf{ft}_c"'] & A \arrow[u, "\mathsf{graph}(\alpha^{-1})"'] \end{tikzcd} \qquad (1)$$

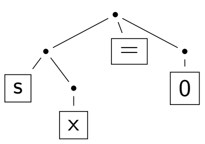
The relation ft_c thus obtained is called the ***finite trace*** *of c.*

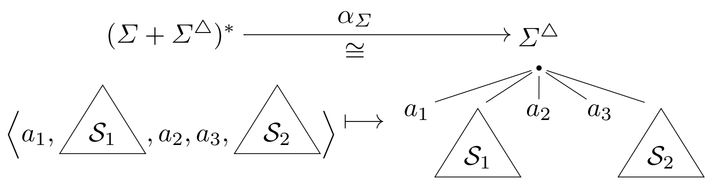
Translating back to the category **Sets**, Theorem 1.1 assigns to each non-deterministic coalgebra $c : X \to \mathcal{P}FX$ its finite trace $\mathsf{ft}_c : X \to \mathcal{P}A$ into the powerset of the carrier of the initial F-algebra.

Example 1.2 (Non-deterministic automata) A non-deterministic automaton over alphabet Σ can be described as a coalgebra $X \to (\mathcal{P}X)^{\Sigma} \times 2$, or equivalently as $c : X \to \mathcal{P}(1 + \Sigma \times X)$ in **Sets**. The set $c(x)$ then contains the unique element $\checkmark$ of $1 = \{\checkmark\}$ if and only if x is an accepting state. For the functor $F = 1 + \Sigma \times -$ involved, the initial F-algebra (in **Sets**) consists of the *strings* (or *lists*) over Σ, as in: $[\mathsf{nil}, \mathsf{cons}] \colon 1 + \Sigma \times \Sigma^* \xrightarrow{\cong} \Sigma^*$. Hence, given a non-deterministic automaton $c : X \to \mathcal{P}FX$, Theorem 1.1 yields its finite trace $\mathsf{ft}_c : X \to \mathcal{P}\Sigma^*$. It is shown later in Example 4.13 that the set $\mathsf{ft}_c(x) \subseteq \Sigma^*$ is indeed the *language accepted by* the automaton when it starts in state x.

Example 1.3 (Context-free grammars) A context-free grammar (CFG) consists of a set Σ of terminal symbols, a set X of non-terminal symbols, and a relation $R \subseteq X \times (\Sigma + X)^*$ consisting of generation rules. It is described as a coalgebra $c : X \to \mathcal{P}\big((\Sigma + X)^*\big)$ in **Sets**.

For the functor $F = (\Sigma + -)^*$, the initial algebra $\Sigma^{\triangle}$ consists of *finite skeletal parse trees* (finite SPTs), which are strings with tree structure. An example of a finite SPT is given below on the left: it describes the formula $\mathsf{s}(\mathsf{x}) = \mathsf{0}$. The initial algebra structure on $\Sigma^{\triangle}$ is illustrated below on the right, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are finite SPTs.





Given a CFG $c : X \to \mathcal{P}FX$, via the finite trace semantics we obtain a function $\mathsf{ft}_c : X \to \mathcal{P}\Sigma^{\triangle}$. Later in Example 4.14 it is shown that, for a non-terminal $x \in X$, the set $\mathsf{ft}_c(x)$ consists of all the finite SPTs that can be generated from x. By applying the flattening function $\Sigma^{\triangle} \to \Sigma^*$, which is defined via the initiality of $\Sigma^{\triangle}$, we obtain the set of strings generated from x.

1.2 Notations

The i-th coprojection into a coproduct $\coprod_{i \in I} X_i$ is denoted by κ_i. If the index set I is finite the coproduct will be written as $X_{i_1} + X_{i_2} + \cdots + X_{i_n}$.

The operator $(-)^*$ defined by $X^* = \coprod_{n<\omega} X^n$ is so-called the *Kleene star*. The set X^* consists of all the *strings* of finite length over X. It is standard that the Kleene star has a monad structure with unit $\overset{*}{\eta}$ creating a string of length one and multiplication $\overset{*}{\mu}$ "flattening" a string of strings into a string. We denote a string of length n by $\langle a_1, a_2, \ldots, a_n \rangle$; $\langle\rangle$ is then the empty string of length zero.

We write a symbol on top of the unit η and multiplication μ to indicate the relevant monad. *E.g.* $\overset{*}{\eta}_\Sigma$ is the Σ-component of the unit of Kleene monad $(-)^*$. Hence $(\overset{*}{\eta}_\Sigma)^*$ is the application of the functor $(-)^*$ to the arrow $\overset{*}{\eta}_\Sigma$.

We will heavily use two powerset functors: one is covariant $\mathcal{P}$ and the other is contravariant $\overline{\mathcal{P}}$. They act the same on objects. For a function $f : X \to Y$, $\mathcal{P}f$ maps a subset of X to its direct image under f, and $\overline{\mathcal{P}}f$ maps a subset $u \subseteq Y$ to its inverse image $f^{-1}(u)$.

It is standard that there is a bijective correspondence between a relation $R \subseteq X \times Y$ and a function $f : X \to \mathcal{P}Y$, given by $f_R(x) = \{y \in Y \mid (x, y) \in R\}$ (*relation-into-function*). In this paper we identify a relation with the corresponding function, and vice versa. Hopefully this will not cause any confusion.

2 Context-free grammars as coalgebras

In this section we give a precise coalgebraic formulation of context-free grammars and context-free languages. For more about traditional treatment of those notions the reader is referred to [LP81].

A (traditional) context-free grammar, described as a triple of terminals, nonterminals and generation rules $R \subseteq X \times (\Sigma + X)^*$, is described as a coalgebra:

$$X \xrightarrow{\;c\;} \mathcal{P}\big((\Sigma + X)^*\big), \qquad \text{namely} \qquad x \longmapsto \{s \in (\Sigma + X)^* \mid \langle x, s\rangle \in R\}.$$

Definition 2.1 (Context-free grammar) In this paper a $\mathcal{P}\big((\Sigma + -)^*\big)$-coalgebra in **Sets** is called a *context-free grammar* (CFG in short) over Σ. Equivalently, via the relation-into-function, a CFG is a $(\Sigma + -)^*$-coalgebra in **Rel**.

Notice that not all $\mathcal{P}\big((\Sigma + -)^*\big)$-coalgebras are context-free grammars in the traditional sense, due to the lack of finiteness conditions on Σ, X and the generation rules. The above definition, which is more liberal and natural from the coalgebraic perspective, ignores algorithmic aspects of context-free grammars. They are not relevant in this paper.

Example 2.2 Consider the following CFG for the syntax of Peano arithmetic.

$$\begin{array}{l}
\Sigma = \{\mathsf{0}, \mathsf{s}, =, \wedge, \vee, \supset, \neg, \forall, \exists\} \cup \mathit{Var}, \qquad X = \{\mathbf{T}, \mathbf{Q}, \mathbf{F}\},\\
\mathbf{T} \to \mathsf{0}, \qquad \mathbf{T} \to \mathsf{x} \quad (\mathsf{x} \in \mathit{Var}), \qquad \mathbf{T} \to \mathsf{s}\mathbf{T},\\
\mathbf{Q} \to \forall\mathsf{x} \quad (\mathsf{x} \in \mathit{Var}), \qquad \mathbf{Q} \to \exists\mathsf{x} \quad (\mathsf{x} \in \mathit{Var}),\\
\mathbf{F} \to \mathbf{T}{=}\mathbf{T}, \quad \mathbf{F} \to \mathbf{F}{\wedge}\mathbf{F}, \quad \mathbf{F} \to \mathbf{F}{\vee}\mathbf{F}, \quad \mathbf{F} \to \mathbf{F}{\supset}\mathbf{F}, \quad \mathbf{F} \to \neg\mathbf{F}, \quad \mathbf{F} \to \mathbf{Q}\mathbf{F}.
\end{array}$$

The induced CFG $c : X \to \mathcal{P}\big((\Sigma + X)^*\big)$ is as follows.

$$c(\mathbf{T}) = \{\mathsf{0}\} \cup \mathit{Var} \cup \{\mathsf{s}\mathbf{T}\}, \qquad c(\mathbf{Q}) = \{\forall\mathsf{x} \mid \mathsf{x} \in \mathit{Var}\} \cup \{\exists\mathsf{x} \mid \mathsf{x} \in \mathit{Var}\},$$
$$c(\mathbf{F}) = \{\mathbf{T}{=}\mathbf{T}, \mathbf{F}{\wedge}\mathbf{F}, \mathbf{F}{\vee}\mathbf{F}, \mathbf{F}{\supset}\mathbf{F}, \neg\mathbf{F}, \mathbf{Q}\mathbf{F}\}.$$

Usually a context-free grammar over Σ is considered as a machine which generates strings over Σ, i.e. elements of Σ^*. However, from a coalgebraic perspective it is more natural to first obtain finite SPTs (i.e. strings with tree structure), and then by flattening obtain strings. In the following the precise definition of finite SPTs is presented, together with a few related notions.

The next definition is a bit complicated; the reader may find an alternative characterization (Proposition 3.1) in terms of initial algebra/final coalgebra.

Definition 2.3 ((Skeletal) parse trees) Let $c : X \to \mathcal{P}\big((\Sigma+X)^*\big)$ be a CFG over Σ. A *parse tree generated by c from $x \in X$* is a (possibly infinite-depth) tree which satisfy the following:

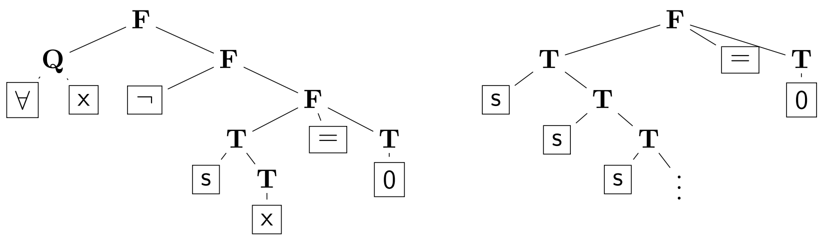
1. All leaf nodes are labelled from $\Sigma + X$;
2. All internal (i.e. non-leaf) nodes are labelled from X;
3. The root is labelled with x;
4. If a leaf node is labelled from X, say with y, then the empty string $\langle\rangle$ belongs to $c(y)$;
5. For each internal node let $y \in X$ be its label and let its immediate successors be labelled with $c_1, c_2, \ldots, c_m$ ($c_i \in \Sigma+X$) from left to right. Then the string $\langle c_1, c_2, \ldots, c_m\rangle$ is an element of $c(y)$.[1]

Condition 5 ensures that a parse tree is finitely-branching. A parse tree is *finite* if its depth is finite.

A *skeletal parse tree (SPT for short) generated by c from x* is a parse tree generated by c from x, with all of its labels from X deleted. It is *finite* if its depth is finite. A *skeletal parse tree (SPT) over Σ* is a skeletal parse tree generated by some CFG c over Σ. Equivalently, it is a finitely-branching, possibly infinite-depth tree with some of its leaves labelled from Σ and its internal nodes not labelled, and if it is trivial (i.e. root-only) then the sole node is not labelled.[2] An SPT is *finite* if its depth is finite.

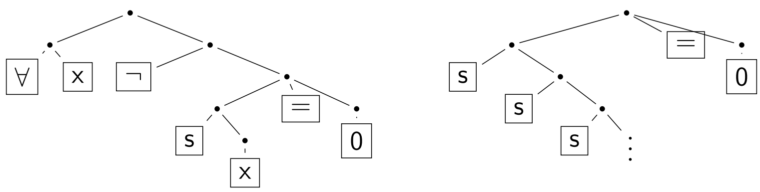
The set of all the (possibly infinite) SPTs over Σ is denoted by $\Sigma^{\wedge}$, and the set of the finite SPTs is denoted by $\Sigma^{\triangle}$.

Example 2.4 Below are two parse trees generated by the context-free grammar in Example 2.2, from the non-terminal symbol **F**. The one on the left is finite.



[1] Condition 4 may be considered as an instance of Condition 5 when $m = 0$.
[2] Since an SPT is generated from a non-terminal symbol.

Forgetting about the non-terminal symbols from X, we obtain the following SPTs generated by the grammar from $\mathbf{F}$.



The infinite one on the right has no corresponding well-formed formula, while the other one can be read as $\forall \mathsf{x}.\neg(\mathsf{s}(\mathsf{x}) = \mathsf{0})$.

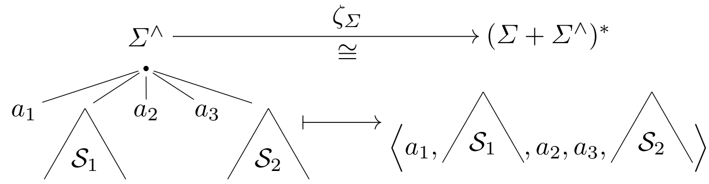
3 Monad structures on languages

In this section we first investigate (co)algebraic structures on the set of finite/infinite SPTs. Then it turns out that the formation of finite SPTs $\Sigma^{\triangle}$ and SPTs $\Sigma^{\wedge}$, from Σ, are all monads just like that of strings Σ^*. Moreover, the embedding $\Sigma^{\triangle} \rightarrowtail \Sigma^{\wedge}$ and the flattening function $\Sigma^{\triangle} \twoheadrightarrow \Sigma^*$ are shown to be both *maps of monads* [BW83].

The following observation is the first step. It may also be read as a definition of $\Sigma^{\triangle}$ and $\Sigma^{\wedge}$. The proof is standard and left to the reader.

Proposition 3.1 *The set $\Sigma^{\triangle}$ of all finite SPTs over Σ carries the initial $(\Sigma + -)^*$-algebra. The algebraic structure α_Σ makes a sequence $\langle c_1, c_2, \ldots, c_n\rangle$ (where $c_i \in \Sigma + \Sigma^{\triangle}$) into a tree by adding a fresh root whose immediate successors are $c_1, c_2, \ldots, c_n$. An example is found in Example 1.3. The empty string $\langle\rangle$ is mapped by α_Σ to the trivial tree which is not labelled.*

The set $\Sigma^{\wedge}$ of (possibly infinite) SPTs over Σ carries the final $(\Sigma + -)^*$-coalgebra. The coalgebraic structure ζ_Σ removes the root and returns the sequence of its immediate successors. For example,[3]



The trivial tree is mapped to the empty string $\langle\rangle$. □

Remark 3.2 We can think of the functor $(\Sigma + -)^*$ as a "signature" in the sense of traditional universal algebra. Let $\star$ be a fresh symbol, and for each

[3] An open triangle designates a tree with a possibly infinite depth, while a closed one is a tree with a finite depth. This conforms to the notation $\Sigma^{\wedge}$ and $\Sigma^{\triangle}$.

$s \in (\Sigma + \{\star\})^*$ let $\|s\|$ denote the number of $\star$'s appearing in s. Then we have an obvious isomorphism

$$(\Sigma + X)^* \cong \textstyle\coprod_{s \in (\Sigma + \{\star\})^*} X^{\|s\|} \ .$$

Hence $(\Sigma + -)^*$ describes a signature such that $(\Sigma + \{\star\})^*$ is the set of operations and each operation $s \in (\Sigma + \{\star\})^*$ is $\|s\|$-ary.

The notation $(-)^{\triangle} : \Sigma \mapsto \Sigma^{\triangle}$ and $(-)^{\wedge} : \Sigma \mapsto \Sigma^{\wedge}$ is used to suggest an analogy with the Kleene (or list, string) monad $(-)^*$. Indeed, these constructions are closely related, as is shown in the sequel.

Each of the mappings $(-)^{\triangle}$ and $(-)^{\wedge}$ extends to a functor, using standard results about initial algebras and final coalgebras for a functor $F(\Sigma, -)$, where F is a bifunctor—in this case $F(X_1, X_2) = (X_1 + X_2)^*$. Moreover, it turns out that both functors $(-)^{\triangle}$ and $(-)^{\wedge}$ have a monad structure. The formation of units $\overset{\triangle}{\eta}, \hat{\eta}$ and multiplications $\overset{\triangle}{\mu}, \hat{\mu}$ is much like for the *free monad* and *free iterative monad* generated by a functor [AAMV03,Jac04a]. The difference is that here the parameter set Σ is inside the Kleene monad $(-)^*$, which adds some complexity. The concrete constructions are described in Appendix A.1. It is straightforward, but laborious, to show that the constructions satisfy the requirements of a monad.

Proposition 3.3 *The triples* $\big((-)^{\triangle}, \overset{\triangle}{\eta}, \overset{\triangle}{\mu}\big)$ *and* $\big((-)^{\wedge}, \hat{\eta}, \hat{\mu}\big)$ *are monads.* $\square$

Let $\iota_\Sigma : \Sigma^{\triangle} \rightarrowtail \Sigma^{\wedge}$ be the canonical embedding of the initial algebra into the final coalgebra. It is a mono by [Bar93, Theorem 3.2].

$$\begin{tikzcd}
(\Sigma + \Sigma^{\triangle})^* \arrow[r, tail, dashed, "(\Sigma + \iota_\Sigma)^*"] \arrow[d, "\alpha_\Sigma"', "\cong"] & (\Sigma + \Sigma^{\wedge})^* \\
\Sigma^{\triangle} \arrow[r, tail, dashed, "\iota_\Sigma"'] & \Sigma^{\wedge} \arrow[u, "\cong", "\zeta_\Sigma"']
\end{tikzcd}$$

It is straightforward to show that ι_Σ is natural in Σ, and is compatible with monad structures, i.e., is a map of monads.

The *flattening* function $\varphi_\Sigma : \Sigma^{\triangle} \to \Sigma^*$, which maps a finite SPT to a flat string demolishing the tree structure, is obtained via initiality of α_Σ.

$$\begin{tikzcd}
(\Sigma + \Sigma^{\triangle})^* \arrow[r, dashed, "(\Sigma + \varphi_\Sigma)^*"] \arrow[d, "\alpha_\Sigma"', "\cong"] & (\Sigma + \Sigma^*)^* \arrow[d, "\overset{*}{\mu}_\Sigma \circ [\, \overset{*}{\eta}_\Sigma \,,\, \Sigma^* \,]^*"] \\
\Sigma^{\triangle} \arrow[r, dashed, "\varphi_\Sigma"'] & \Sigma^*
\end{tikzcd}$$

It is easy to see that the flattening map is a map of monads. Moreover, it is obviously an epi: for a sequence $\langle a_1, a_2, \ldots, a_n \rangle \in \Sigma^*$ take the finite SPT of depth 2 that has leaves $a_1, a_2, \ldots, a_n$ from left to right.

Hence we have obtained the following result.

Proposition 3.4 *The embedding ι_Σ and the flattening map φ_Σ both form a map of monads. They yield the following "fundamental span of languages".*

$$\Sigma^* \xleftarrow{\varphi_\Sigma} \Sigma^\triangle \xrightarrow{\iota_\Sigma} \Sigma^\wedge \qquad \square$$

4 Finite trace semantics for coalgebras

This section presents the main technical result (already previewed as Theorem 1.1) that an initial algebra in **Sets** (of a suitable functor) yields a final coalgebra in **Rel**. Examples 1.2 and 1.3 are also fully elaborated in greater detail.

4.1 Shapely functors

The family of endofunctors F in **Sets** we are interested in is that of *shapely functors* [Jay95]. The following inductive definition is equivalent to the original one.

Definition 4.1 (Shapely functors) The family of *shapely functors* is defined inductively by the following BNF notation:

$$F, G, F_i ::= \mathrm{id} \mid \Sigma \mid F \times G \mid \coprod_{i \in I} F_i \ ,$$

where Σ denotes the constant functor into Σ.

Notice that we can take the exponentiation $(-)^\Sigma$ to the power of a finite set Σ in building a shapely functor, because X^Σ is isomorphic to the $|\Sigma|$-fold product of X's. A shapely functor is different from a polynomial functor in the following points: we cannot take an exponentiation with an infinite set (because it makes Lemma 4.2.2 fail), but we can take an infinite coproduct—so that we can form the Kleene star $(-)^* = \coprod_{n<\omega}(-)^n$. A shapely functor has the following properties needed for our purpose.

Lemma 4.2 *Let $F : \mathbf{Sets} \to \mathbf{Sets}$ be a shapely functor.*

1. *F preserves weak pullbacks.*
2. *For an arrow $?_X : 0 \rightarrowtail X$ with domain 0, $F?_X : F0 \to FX$ is mono. Hence F preserves all monos in* **Sets**.
3. *F preserves ω-colimits and ω^{op}-limits. Hence F has both the initial algebra and the final coalgebra. They are, together with the canonical embedding, denoted as follows.*

$$\begin{tikzcd} FA \arrow[r, tail, dashed, "F\iota"] \arrow[d, "\alpha"', "\cong"] & FZ \\ A \arrow[r, tail, dashed, "\iota"'] & Z \arrow[u, "\zeta"', "\cong"] \end{tikzcd}$$

Proof. The proofs are easy by induction on the construction of F. The preservation of ω-colimits (or ω^{op}-limits) allows us to obtain the initial F-algebra (or the final F-coalgebra) as the colimit (or limit) of the initial sequence of length ω (or final sequence, respectively): see e.g. [Bar93,AK95]. $\square$

4.2 Relation lifting, distributive law and Kleisli category

An endofunctor F yields a *relation lifting*: given a relation $\langle r_1, r_2\rangle : R \rightarrowtail X \times Y$, a lifted relation $\mathrm{Rel}_F(R) \rightarrowtail FX \times FY$ is defined by image factorization.

$$\begin{tikzcd} FR \arrow[r, two heads] \arrow[dr, "\langle Fr_1, Fr_2\rangle"'] & \mathrm{Rel}_F(R) \arrow[d, tail] \\ & FX \times FY \end{tikzcd}$$

The following compatibility results hold for a functor F which preserves weak pullbacks, hence in particular for a shapely F (Lemma 4.2).

Lemma 4.3 *Relation lifting is compatible with such operations on relations as:*

1. *Composition: for* $R \rightarrowtail X \times Y$*,* $S \rightarrowtail Y \times Z$ *and their composition* $S \circ R = \{(x, z) \in X \times Z \mid \exists y \in Y.(x, y) \in R$ *and* $(y, z) \in Z\}$ *we have* $\mathrm{Rel}_F(S \circ R) = \mathrm{Rel}_F(S) \circ \mathrm{Rel}_F(R)$*.*
2. *Graph of a function and functor application: for a function* $f : X \to Y$ *and its graph* $\mathsf{graph}(f) = \{(x, f(x)) \mid x \in X\}$ *we have* $\mathrm{Rel}_F(\mathsf{graph}(f)) = \mathsf{graph}(Ff)$*.*
3. *Inverse image and direct image: for functions* $f_1 : X_1 \to Y_1$*,* $f_2 : X_2 \to Y_2$ *and relations* $R \rightarrowtail X_1 \times X_2$*,* $S \rightarrowtail Y_1 \times Y_2$*, let us denote the inverse image and the direct image by* $(f_1 \times f_2)^{-1}(S) = \big\{(x_1, x_2) \mid \big(f_1(x_1), f_2(x_2)\big) \in S\big\}$*, and* $\coprod_{f_1 \times f_2}(R) = \{\, (\, f_1(x_1)\, , f_2(x_2)\,) \mid (x_1, x_2) \in R\}$*. Then we have*

$$\begin{aligned} &\mathrm{Rel}_F\big((f_1 \times f_2)^{-1}(S)\big) = (Ff_1 \times Ff_2)^{-1}\big(\mathrm{Rel}_F(S)\big) \ , \\ &\mathrm{Rel}_F\big(\textstyle\coprod_{f_1 \times f_2}(R)\big) = \textstyle\coprod_{Ff_1 \times Ff_2}\big(\mathrm{Rel}_F(R)\big) \ . \end{aligned}$$ □

The membership relation $\in_X \rightarrowtail X \times \mathcal{P}X$ on a set X is lifted to $\mathrm{Rel}_F(\in_X) \rightarrowtail FX \times F\mathcal{P}X$. By transposition we obtain the following function λ_X.

$$F\mathcal{P}X \xrightarrow{\lambda_X} \mathcal{P}FX \qquad\qquad u \longmapsto \{a \in FX \mid \langle a, u\rangle \in \mathrm{Rel}_F(\in_X)\}$$

Then the map λ_X is: 1) natural in X, and 2) compatible with the monad structure of $\mathcal{P}$: when we denote the unit (singleton map) by $\{-\}$ and the multiplication (union) by $\bigcup$, the following diagrams commute.

$$\begin{tikzcd} FX \arrow[r, "F\{-\}_X"] \arrow[dr, "\{-\}_{FX}"'] & F\mathcal{P}X \arrow[d, "\lambda_X"] \\ & \mathcal{P}FX \end{tikzcd} \qquad \begin{tikzcd} F\mathcal{P}^2X \arrow[r, "\lambda_{\mathcal{P}X}"] \arrow[d, "F\bigcup_X"'] & \mathcal{P}F\mathcal{P}X \arrow[r, "\mathcal{P}\lambda_X"] & \mathcal{P}^2FX \arrow[d, "\bigcup_{FX}"] \\ F\mathcal{P}X \arrow[rr, "\lambda_X"'] & & \mathcal{P}FX \end{tikzcd}$$

This says that the natural transformation $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ is a *distributive law*.[4]

[4] The use of a distributive law in coalgebraic settings is investigated elaborately in [Bar04].

Lemma 4.4 ([Jac04b]) *The maps λ_X thus defined form a distributive law of a functor F over a monad $\mathcal{P}$. It is called the "power law".* □

Example 4.5 For the functor $F = 1 + \Sigma \times -$, where $1 = \{\checkmark\}$, the lifted membership relation is as follows.

$$\mathrm{Rel}_{1+\Sigma\times-}(\in_X) = \{(\checkmark, \checkmark)\} \cup \big\{ \big((a,x),(a,u)\big) \mid a \in \Sigma, x \in u \big\} \ .$$

For the functor $F = (\Sigma + -)^*$, the lifted membership relation $\mathrm{Rel}_{(\Sigma+-)^*}(\in_X)$ between $(\Sigma + X)^*$ and $(\Sigma + \mathcal{P}X)^*$ is described concretely as follows: a pair $\langle\, c_1c_2 \dots c_m,\, d_1d_2 \dots d_m \,\rangle$ belongs to $\mathrm{Rel}_{(\Sigma+-)^*}(\in_X)$ if and only if for each $i = 1, 2, \dots, m$,

- if $c_i \in \Sigma$ then d_i is also from Σ and $c_i = d_i$;
- if $c_i \in X$ then d_i is in $\mathcal{P}X$ and $c_i \in d_i$.

The distributive law $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ gives rise to a lifting $F : \mathbf{Sets}_{\mathcal{P}} \to \mathbf{Sets}_{\mathcal{P}}$ of a functor F in the Kleisli category by

$$F : \quad \big(X \xrightarrow{\quad f \quad} Y \big) \quad \mapsto \quad \big(FX \xrightarrow{\lambda_Y \circ Ff} FY \big) \ .$$

In the sequel we identify the category **Rel** with the Kleisli category $\mathbf{Sets}_{\mathcal{P}}$ of the powerset monad. It is justified by the following straightforward observation.

Lemma 4.6 *The category* **Rel** *of sets and relations is isomorphic to the Kleisli category* $\mathbf{Sets}_{\mathcal{P}}$ *via the relation-into-function correspondence.*

Moreover, let F be an endofunctor in **Sets** *which preserves weak pullbacks. Then the canonical lifting of F in* **Rel** *in the sense of [CKW91], which maps an arrow $R : X \to Y$ to $\mathrm{Rel}_F(R) : FX \to FY$, coincides with the lifting of F in* $\mathbf{Sets}_{\mathcal{P}}$ *defined above via the distributive law.* □

Remark 4.7 As is already noted, working in the Kleisli category $\mathbf{Sets}_{\mathcal{P}}$ makes it easier to generalize to other monads than $\mathcal{P}$. A similar finality result holds for the *subdistribution monad* $\mathcal{D}$ such that $\mathcal{D}X = \{d : X \to [0,1] \mid \sum_{x \in X} d(x) \leq 1\}$. In that case we do not have the counterpart of the notion of relation lifting but start with a distributive law $F\mathcal{D} \Rightarrow \mathcal{D}F$. Details will be published later.

4.3 Contravariant powerset functor

We use the contravariant powerset functor $\overline{\mathcal{P}}$ in our construction. The following properties are used there.

Lemma 4.8 *1. For a mono $m : X \rightarrowtail Y$, $\mathcal{P}m$ is a split mono with its left inverse $\overline{\mathcal{P}}m$, i.e. $\overline{\mathcal{P}}m \circ \mathcal{P}m = \mathrm{id}_{\mathcal{P}X}$.*
2. For an iso $i : X \xrightarrow{\cong} Y$, $\mathcal{P}i$ is again an iso with inverse $\overline{\mathcal{P}}i$.
3. The union maps $\bigcup_X : \mathcal{P}^2 X \to \mathcal{P}X$ form a natural transformation $\mathcal{P}\overline{\mathcal{P}} \Rightarrow \overline{\mathcal{P}}$.
4. For each n, the maps $\lambda^n_X : F^n\mathcal{P}X \to \mathcal{P}F^nX$ in Lemma 4.9 form a natural transformation $F^n\overline{\mathcal{P}} \Rightarrow \overline{\mathcal{P}}F^n$.

Proof. See Appendix A.2. □

4.4 Construction of finite trace via composition of coalgebra

In the construction of the finite trace, we use the *n-fold composition* $c^n : X \to \mathcal{P}F^nX$ of a coalgebra $c : X \to \mathcal{P}FX$ in **Sets**. Intuitively, one transition of c^n corresponds to n successive transitions of the original coalgebra c. It is defined inductively on n as follows.

$$c^0 = \{-\}_X \ , \qquad \begin{tikzcd} X \arrow[r, "c"] \arrow[rrrd, "c^{n+1}"'] & \mathcal{P}FX \arrow[r, "\mathcal{P}Fc^n"] & \mathcal{P}F\mathcal{P}F^nX \arrow[r, "\mathcal{P}\lambda_{F^nX}"] & \mathcal{P}^2F^{n+1}X \arrow[d, "\bigcup_{F^{n+1}X}"] \\ & & & \mathcal{P}F^{n+1}X \end{tikzcd} \ .$$

The next observation is basic for the n-fold composition of a coalgebra.

Lemma 4.9 ([Wor]) *The distributive law $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ extends to n-fold distributive law $\lambda^n : F^n\mathcal{P} \Rightarrow \mathcal{P}F^n$ in the following way.*

$$\lambda^0_X = \mathrm{id}_{\mathcal{P}X} \ , \qquad \begin{tikzcd} F^{n+1}\mathcal{P}X \arrow[r, "F^n\lambda_X"] \arrow[dr, "\lambda^{n+1}_X"'] & F^n\mathcal{P}FX \arrow[d, "\lambda^n_{FX}"] \\ & \mathcal{P}F^{n+1}X \end{tikzcd} \ .$$

Let $c : X \to \mathcal{P}FX$ be a coalgebra in **Sets**. *For each n, m the following diagrams commute.*

$$\begin{tikzcd} F^{n+m}\mathcal{P}X \arrow[r, "F^n\lambda^m_X"] \arrow[d, "F^m\lambda^n_X"'] \arrow[dr, "\lambda^{n+m}_X"] & F^n\mathcal{P}F^mX \arrow[d, "\lambda^n_{F^mX}"] \\ F^m\mathcal{P}F^nX \arrow[r, "\lambda^m_{F^nX}"'] & \mathcal{P}F^{n+m}X \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "c^n"] \arrow[rrdd, "c^{n+m}"'] & \mathcal{P}F^nX \arrow[r, "\mathcal{P}F^nc^m"] & \mathcal{P}F^n\mathcal{P}F^mX \arrow[d, "\mathcal{P}\lambda^n_{F^mX}"] \\ & & \mathcal{P}^2F^{n+m}X \arrow[d, "\bigcup_{F^{n+m}X}"] \\ & & \mathcal{P}F^{n+m}X \end{tikzcd}$$

Proof. By induction. □

Now we are ready to prove our main technical result.

Theorem 4.10 (Finite trace semantics for coalgebras, Theorem 1.1) *Let F be a shapely functor, and $\alpha : FA \xrightarrow{\cong} A$ be the initial F-algebra in* **Sets**. *The coalgebra $\{-\}_{FA} \circ \alpha^{-1} : A \xrightarrow{\cong} FA$ in $\mathbf{Sets}_{\mathcal{P}}$ is final for the lifted functor F.*

Proof. Given a coalgebra $c : X \to FX$ in $\mathbf{Sets}_{\mathcal{P}}$, we construct an arrow $\mathsf{ft}_c : X \to A$, and show that it is the unique arrow which makes the diagram in $\mathbf{Sets}_{\mathcal{P}}$

on the left (equivalently, the diagram in **Sets** on the right) commute.

$$
\begin{tikzcd}
FX \arrow[r, dashed, "F\mathsf{ft}_c"] & FA \\
X \arrow[u, "c"] \arrow[r, dashed, "\mathsf{ft}_c"'] & A \arrow[u, "\{-\}_{FA} \circ \alpha^{-1}"']
\end{tikzcd} ,
\qquad
\begin{tikzcd}
\mathcal{P}FX \arrow[r, dashed, "\mathcal{P}F\mathsf{ft}_c"] & \mathcal{P}F\mathcal{P}A \arrow[d, "\mathcal{P}\lambda_A"] \\
 & \mathcal{P}^2FA \arrow[d, "\bigcup_{FA}"] \\
 & \mathcal{P}FA \arrow[d, "\mathcal{P}\alpha", "\cong"'] \\
X \arrow[uuu, "c"] \arrow[r, dashed, "\mathsf{ft}_c"'] & \mathcal{P}A
\end{tikzcd} .
\tag{2}
$$

In the rest of the proof we work in the category **Sets**.

As is stated in Lemma 4.2, the initial F-algebra in **Sets** for shapely F is obtained via the initial sequence $0 \to F0 \to F^2 0 \to \cdots$ as follows.

$$
\begin{tikzcd}
 & & & & A \arrow[d, bend left, "\alpha^{-1}"] \\
\cdots \arrow[r] & F^n 0 \arrow[r, "F^n ?_{F0}"] \arrow[urrr, bend left, "\sigma_n"] \arrow[drrr, bend right, "F\sigma_{n-1}"'] & F^{n+1}0 \arrow[r] \arrow[urr, "\sigma_{n+1}"] \arrow[drr, "F\sigma_n"'] & \cdots & \\
 & & & & FA \arrow[u, bend left, "\alpha"] \arrow[u, phantom, "\cong"]
\end{tikzcd}
\tag{3}
$$

The cocone $\{\sigma_n : F^n 0 \to A\}_{n<\omega}$ is by construction the colimit of the initial sequence. Since a shapely F preserves ω-colimits the cocone $\{?_{FA} : 0 \to FA\} \cup \{F\sigma_n : F^{n+1}0 \to FA\}_{n<\omega}$ is again a colimit, yielding the initial algebra α as the mediating iso arrow. Lemma 4.2.2 shows that each σ_n is mono.

We define the *n-th trace* $\mathsf{trace}_c^n : X \to \mathcal{P}A$ of c by the following composite. The n-th trace $\mathsf{trace}_c^n(x) \subseteq A$ is understood as the set of behavior of x which terminates within n steps.

$$
\begin{tikzcd}
X \arrow[r, "c^n"] \arrow[drr, "\mathsf{trace}_c^n"'] & \mathcal{P}F^n X \arrow[r, "\overline{\mathcal{P}}F^n ?_X"] & \mathcal{P}F^n 0 \arrow[d, "\mathcal{P}\sigma_n"] \\
 & & \mathcal{P}A
\end{tikzcd}
$$

For n-th traces the following equality holds, which says that all behavior within n steps are already included in trace^n. For $n \le m$,

$$
\mathrm{Im}\,\sigma_n \cap \mathsf{trace}_c^m(x) = \mathsf{trace}_c^n(x) \ , \tag{4}
$$

where $\mathrm{Im}\,\sigma_n$ is the direct image $\sigma_n[F^n 0]$. The proof is given in Appendix A.3.

Finally, we define the finite trace $\mathsf{ft}_c : X \to \mathcal{P}A$ of c as the union of n-th traces: for each $x \in X$,

$$
\mathsf{ft}_c(x) \stackrel{\mathrm{def}}{=} \textstyle\bigcup_{n<\omega} \mathsf{trace}_c^n(x) \ .
$$

By the equality (4) we have another characterization of $\mathsf{ft}_c(x)$: for each n and $t_n \in F^n 0$, $\sigma_n(t_n) \in \mathsf{ft}_c(x)$ if and only if $\sigma_n(t_n) \in \mathsf{trace}_c^n(x)$. Hence, by Lemma 4.8.1, for each n we have the following equality of functions $X \to F^n 0$.

$$\overline{\mathcal{P}}\sigma_n \circ \mathsf{ft}_c = \overline{\mathcal{P}}\sigma_n \circ \mathsf{trace}_c^n = \overline{\mathcal{P}}F^n ?_X \circ c^n \ . \tag{5}$$

In the following Lemmas 4.11 and 4.12 we show that the arrow ft_c thus constructed is indeed the unique arrow that makes the diagram (2) commute. □

Lemma 4.11 *The arrow* $\mathsf{ft}_c : X \to \mathcal{P}A$ *in* **Sets***, as defined in the proof of Theorem 4.10, makes the diagram (2) commute.*

Proof. By the construction of the initial algebra as the colimit (i.e. coequalizer of coproduct), it suffices to prove that: for each $n < \omega$ and $t_n \in F^n 0$,

$$\sigma_n(t_n) \in \mathsf{ft}_c(x) \iff \sigma_n(t_n) \in (\mathcal{P}\alpha \circ \bigcup_{FA} \circ \mathcal{P}\lambda_A \circ \mathcal{P}F\mathsf{ft}_c \circ c)(x) \ .$$

When $n = 0$, we have $F^n 0 = 0$ hence the equivalence trivially holds. When $n > 0$, we proceed as follows.

$$\begin{aligned}
&\sigma_n(t_n) \in (\mathcal{P}\alpha \circ \textstyle\bigcup_{FA} \circ \mathcal{P}\lambda_A \circ \mathcal{P}F\mathsf{ft}_c \circ c)(x) \\
&\iff t_n \in (\overline{\mathcal{P}}\sigma_n \circ \mathcal{P}\alpha \circ \textstyle\bigcup_{FA} \circ \mathcal{P}\lambda_A \circ \mathcal{P}F\mathsf{ft}_c \circ c)(x) \\
&\iff t_n \in (\overline{\mathcal{P}}\sigma_n \circ \overline{\mathcal{P}}\alpha^{-1} \circ \textstyle\bigcup_{FA} \circ \mathcal{P}\lambda_A \circ \mathcal{P}F\mathsf{ft}_c \circ c)(x) \\
&\qquad\qquad (\mathcal{P}\alpha = (\mathcal{P}\alpha^{-1})^{-1} = \overline{\mathcal{P}}\alpha^{-1} \text{ by Lemma 4.8.2}) \\
&\iff t_n \in (\overline{\mathcal{P}}F\sigma_{n-1} \circ \textstyle\bigcup_{FA} \circ \mathcal{P}\lambda_A \circ \mathcal{P}F\mathsf{ft}_c \circ c)(x) \qquad (\alpha^{-1} \circ \sigma_n = F\sigma_{n-1}) \\
&\iff t_n \in (\textstyle\bigcup_{F^n 0} \circ \mathcal{P}\lambda_{F^{n-1}0} \circ \mathcal{P}F\overline{\mathcal{P}}\sigma_{n-1} \circ \mathcal{P}F\mathsf{ft}_c \circ c)(x) \quad (\text{Lemma 4.8.3,4}) \\
&\iff t_n \in (\textstyle\bigcup_{F^n 0} \circ \mathcal{P}\lambda_{F^{n-1}0} \circ \mathcal{P}F\overline{\mathcal{P}}F^{n-1}?_X \circ \mathcal{P}Fc^{n-1} \circ c)(x) \qquad (\text{By (5)}) \\
&\iff t_n \in (\overline{\mathcal{P}}F^n?_X \circ \textstyle\bigcup_{F^n X} \circ \mathcal{P}\lambda_{F^{n-1}X} \circ \mathcal{P}Fc^{n-1} \circ c) \qquad (\text{Lemma 4.8.4,3}) \\
&\iff t_n \in (\overline{\mathcal{P}}F^n?_X \circ c^n) \qquad (\text{Definition of } c^n) \\
&\iff \sigma_n(t_n) \in \mathsf{ft}_c(x) \ . \qquad (\text{By (5)})
\end{aligned}$$

This concludes the proof. □

Lemma 4.12 *If an arrow* $f : X \to \mathcal{P}A$ *in* **Sets** *makes the diagram (2) commute in place of* ft_c*, then* f *is equal to* ft_c *as defined in the proof of Theorem 4.10.*

Proof. It suffices to show that

$$\overline{\mathcal{P}}\sigma_n \circ f = \overline{\mathcal{P}}\sigma_n \circ \mathsf{ft}_c \ , \tag{6}$$

since, if it holds, for each $x \in X$, $n < \omega$ and $t_n \in F^n 0$ we have

$$\sigma_n(t_n) \in f(x) \iff \sigma_n(t_n) \in \mathsf{ft}_c(x)$$

which yields the lemma. We show (6) by induction on n.

When $n = 0$ the claim trivially holds. For $n + 1$,

$$\begin{aligned}
\overline{\mathcal{P}}\sigma_{n+1} \circ f &= \overline{\mathcal{P}}\sigma_{n+1} \circ \mathcal{P}\alpha \circ \bigcup\nolimits_{FA} \circ \mathcal{P}\lambda_A \circ \mathcal{P}Ff \circ c \\
&\hspace{6cm} \text{(f makes the diagram (2) commute)} \\
&= \bigcup\nolimits_{F^{n+1}0} \circ \mathcal{P}\lambda_{F^n 0} \circ \mathcal{P}F\overline{\mathcal{P}}\sigma_n \circ \mathcal{P}Ff \circ c \\
&\hspace{6cm} \text{(As in the proof of Lemma 4.11)} \\
&= \bigcup\nolimits_{F^{n+1}0} \circ \mathcal{P}\lambda_{F^n 0} \circ \mathcal{P}F\overline{\mathcal{P}}\sigma_n \circ \mathcal{P}F\mathsf{ft}_c \circ c \\
&\hspace{4cm} \text{($\overline{\mathcal{P}}\sigma_n \circ f = \overline{\mathcal{P}}\sigma_n \circ \mathsf{ft}_c$ by induction hypothesis)} \\
&= \overline{\mathcal{P}}\sigma_{n+1} \circ \mathsf{ft}_c \ . \quad \text{(Same calculation as above, but now backwards)}
\end{aligned}$$

This concludes the proof. □

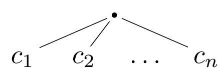
Example 4.13 (Non-deterministic automata) We continue from Example 1.2. For the functor $F = 1 + \Sigma \times -$, the commutation of the diagram (2) amounts to the following conditions.

$$\begin{aligned}
\langle\rangle \in \mathsf{ft}_c(x) &\iff \checkmark \in c(x) \ , \\
\mathsf{cons}(a, s) \in \mathsf{ft}_c(x) &\iff \exists x' \in X. \quad (a, x') \in c(x) \ \wedge \ s \in \mathsf{ft}_c(x') \ .
\end{aligned}$$

These conditions indeed (corecursively) characterize the language $\mathsf{ft}_c(x)$ accepted by the non-deterministic automaton c when we start from x.

Bartels [Bar04] gives an alternative characterization of the accepted language, using a different distributive law. The precise relationship with our work is yet to be determined.

Example 4.14 (Context-free grammar) We continue from Example 1.3. For the functor $F = (\Sigma + -)^*$, the commutation of the diagram (2) amounts to the following conditions. For each element



of $\mathsf{ft}_c(x)$ (here $c_1, c_2, \ldots, c_n \in \Sigma + \Sigma^{\triangle}$), there exists a string $\langle d_1, d_2, \ldots, d_n\rangle \in c(x)$ such that for each i:

- if $c_i \in \Sigma$ then d_i is also in Σ and $c_i = d_i$;
- if $c_i \in \Sigma^{\triangle}$ then d_i is in X and $c_i \in \mathsf{ft}_c(d_i)$.

Hence we obtain the set of finite SPTs generated by c from x as $\mathsf{ft}_c(x)$ via finality in $\mathbf{Sets}_{\mathcal{P}}$.

5 (Possibly infinite) trace semantics

In this section we relate our current work to earlier work [Jac04b], where the final coalgebra in **Sets** gives rise to a weakly final coalgebra in **Rel**.

Theorem 5.1 (Main result of [Jac04b]) *Let F be a shapely functor, and $\zeta : Z \xrightarrow{\cong} FZ$ be the final coalgebra in* **Sets**.

1. *The coalgebra $\mathsf{graph}(\zeta) : Z \to FZ$ is weakly final for the lifted functor F in* **Rel**. *That is, given a coalgebra $c : X \to FX$, there exists a (not necessarily unique) relation $t : X \to Z$ that makes the following diagrams commute.*

$$\begin{tikzcd} FX \arrow[r, rightsquigarrow, "Ft"] & FZ \\ X \arrow[u, "c"] \arrow[r, rightsquigarrow, "t"'] & Z \arrow[u, "\mathsf{graph}(\zeta)"'] \end{tikzcd} \tag{7}$$

2. *There is a canonical choice mt_c (maximum trace) of a trace of c, namely the maximum one with respect to the inclusion order.* □

It turns out that the finite trace of a coalgebra gives rise to the smallest trace via canonical embedding $\iota : A \rightarrowtail Z$.

Corollary 5.2 *Let F be a shapely functor, and $c : X \to \mathcal{P}FX$ be a coalgebra in* **Sets**.

1. *Each trace t of c gives rise to the finite trace of c by $X \xrightarrow{t} \mathcal{P}Z \xrightarrow{\overline{\mathcal{P}}\iota} \mathcal{P}A$ in* **Sets**.
2. *The finite trace ft_c gives rise to a trace of c by $X \xrightarrow{\mathsf{ft}_c} \mathcal{P}A \xrightarrow{\mathcal{P}\iota} \mathcal{P}Z$ in* **Sets**. *Moreover, this trace is the smallest among the traces of c.*

Proof. A trace induces the finite trace, and vice versa, since the following diagram in **Sets** commutes. For the former take the three squares on the left and put them on the right of the definition of a trace, and for the latter take those on the right.

$$\begin{tikzcd}[column sep=huge] \mathcal{P}F\mathcal{P}Z \arrow[r, "\mathcal{P}F\overline{\mathcal{P}}\iota"] \arrow[d, "\mathcal{P}\lambda_Z"'] \arrow[dr, phantom, "\text{(i)}"] & \mathcal{P}F\mathcal{P}A \arrow[r, "\mathcal{P}F\mathcal{P}\iota"] \arrow[d, "\mathcal{P}\lambda_A"] \arrow[dr, phantom, "\text{(iv)}"] & \mathcal{P}F\mathcal{P}Z \arrow[d, "\mathcal{P}\lambda_Z"] \\ \mathcal{P}^2FZ \arrow[r, "\mathcal{P}\overline{\mathcal{P}}F\iota"'] \arrow[d, "\bigcup_{FZ}"'] \arrow[dr, phantom, "\text{(ii)}"] & \mathcal{P}^2FA \arrow[r, "\mathcal{P}^2F\iota"'] \arrow[d, "\bigcup_{FA}"] \arrow[dr, phantom, "\text{(v)}"] & \mathcal{P}^2FZ \arrow[d, "\bigcup_{FZ}"] \\ \mathcal{P}FZ \arrow[r, "\overline{\mathcal{P}}F\iota"'] \arrow[d, bend left, "\overline{\mathcal{P}}\zeta"] \arrow[d, phantom, "\cong"] \arrow[dr, phantom, "\text{(iii)}"] & \mathcal{P}FA \arrow[r, "\mathcal{P}F\iota"'] \arrow[d, bend left, "\mathcal{P}\alpha"] \arrow[d, phantom, "\cong"] \arrow[dr, phantom, "\text{(vi)}"] & \mathcal{P}FZ \\ \mathcal{P}Z \arrow[r, "\overline{\mathcal{P}}\iota"'] \arrow[u, bend left, "\mathcal{P}\zeta"] & \mathcal{P}A \arrow[r, "\mathcal{P}\iota"'] \arrow[u, bend left, "\overline{\mathcal{P}}\alpha"] & \mathcal{P}Z \arrow[u, "\mathcal{P}\zeta"', "\cong"] \end{tikzcd}$$

Square (i) commutes by Lemma 4.8.4, (ii) by Lemma 4.8.3, (iii) is the definition of ι mapped by $\overline{\mathcal{P}}$, (iv) commutes by naturality of λ, (v) by naturality of $\bigcup$, and (vi) is the definition of ι.

It remains to be shown that the trace $\mathcal{P}\iota \circ \mathsf{ft}_c$ is the smallest trace. Take an arbitrary trace $t : X \to \mathcal{P}Z$ of c. It induces the finite trace by $\overline{\mathcal{P}}\iota \circ t$, and by Theorem 4.10 (uniqueness of the finite trace) we have $\mathsf{ft}_c = \overline{\mathcal{P}}\iota \circ t$. Since in general $(\mathcal{P}f \circ \overline{\mathcal{P}}f)(u) \subseteq u$ holds, we have $\mathcal{P}\iota \circ \mathsf{ft}_c = \mathcal{P}\iota \circ \overline{\mathcal{P}}\iota \circ t \subseteq t$. $\square$

6 Conclusions and future work

We have presented that under suitable mild restrictions the initial algebra in **Sets** gives rise to the final coalgebra in **Rel**. The relation induced by the finality in **Rel** extracts the set of finite behavior of non-deterministic systems. The technical result is applied to non-deterministic automata and the first coalgebraic account of context-free grammars/languages. The (co)algebraic and monadic structures on strings and skeletal parse trees have been also elaborated.

The well-known relationship between context-free languages and *pushdown automata* (see e.g.[LP81]) would be an interesting topic to consider from a coalgebraic perspective. So is the problem of *parsing*, which is a partial inverse of the flattening function φ_Σ in Section 3.

As mentioned in Remark 4.7 we are now applying the current approach to another monad than $\mathcal{P}$, namely the subdistribution monad.

Acknowledgements

Thanks are due to Ana (Sokolova) Woracek and anonymous referees for valuable comments and discussion.

References

[AAMV03] P. Aczel, J. Adámek, S. Milius, and J. Velebil. Infinite trees and completely iterative theories: a coalgebraic view. *Theor. Comp. Sci.*, 300:1–45, 2003.

[AK95] J. Adámek and V. Koubek. On the greatest fixed point of a set functor. *Theor. Comp. Sci.*, 150(1):57–75, 1995.

[ASU86] A.V. Aho, R. Sethi, and J.D. Ullman. *Compilers: Principles, Techniques, and Tools.* Addison-Wesley series in Computer Science. Addison-Wesley, 1986.

[Bar93] M. Barr. Terminal coalgebras in well-founded set theory. *Theor. Comp. Sci.*, 114:299–315, 1993.

[Bar04] F. Bartels. *On Generalized Coinduction and Probabilistic Specification Formats: Distributive Laws in Coalgebraic Modelling.* PhD thesis, Free Univ. Amsterdam, 2004.

[BW83] M. Barr and C. Wells. *Toposes, Triples and Theories.* Springer-Verlag, 1983. Available free for downloading at `http://www.cwru.edu/artsci/math/wells/pub/ttt.html`.

[Cho56] N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124, 1956.
[CKW91] A. Carboni, G. Kelly, and R. Wood. A 2-categorical approach to change of base and geometric morphisms I. *Cah. de Top. et Géom. Diff.*, 32(1):47–95, 1991.
[Jac04a] B. Jacobs. Relating two approaches to coinductive solution of recurisve equations. In *Coalgebraic Methods in Computer Science (CMCS 2004)*, volume 106 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, Amsterdam, 2004.
[Jac04b] B. Jacobs. Trace semantics for coalgebras. In *Coalgebraic Methods in Computer Science (CMCS 2004)*, volume 106 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, Amsterdam, 2004.
[Jac05a] B. Jacobs. A bialgebraic review of regular expressions, deterministic automata and languages. Techn. Rep. NIII-R05003, Inst. for Computing and Information Sciences, Radboud Univ. Nijmegen, 2005.
[Jac05b] B. Jacobs. Introduction to coalgebra. Towards mathematics of states and observations. Draft of a book, `www.cs.ru.nl/B.Jacobs/PAPERS/index.html`, 2005.
[Jay95] C. Jay. A semantics for shape. *Science of Comput. Progr.*, 25:251–283, 1995.
[LP81] H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation.* Prentice-Hall, 1981.
[Rut00] J. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000.
[Rut03] J. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *Theor. Comp. Sci.*, 308:1–53, 2003.
[Wor] A. (Sokolova) Woracek. Personal communication.

A Appendix

A.1 Functor/monad structure of $(-)^{\triangle}$ and $(-)^{\wedge}$

On a function $f : \Sigma \to \Phi$, the action of $(-)^{\triangle}$ and $(-)^{\wedge}$ is obtained as follows.

$$
\begin{tikzcd}
(\Sigma + \Sigma^{\triangle})^* \arrow[r, dashed, "(\Sigma + f^{\triangle})^*"] \arrow[dd, "\alpha_\Sigma"', "\cong"] & (\Sigma + \Phi^{\triangle})^* \arrow[d, "(f + \Phi^{\triangle})"] \\
& (\Phi + \Phi^{\triangle})^* \arrow[d, "\alpha_\Phi", "\cong"'] \\
\Sigma^{\triangle} \arrow[r, dashed, "f^{\triangle}"'] & \Phi^{\triangle}
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
(\Phi + \Sigma^{\wedge})^* \arrow[r, dashed, "(\Phi + f^{\wedge})^*"] & (\Phi + \Phi^{\wedge})^* \\
(\Sigma + \Sigma^{\wedge})^* \arrow[u, "(f + \Sigma^{\wedge})^*"] & \\
\Sigma^{\wedge} \arrow[u, "\zeta_\Sigma", "\cong"'] \arrow[r, dashed, "f^{\wedge}"'] & \Phi^{\wedge} \arrow[uu, "\cong", "\zeta_\Phi"']
\end{tikzcd}
$$

The monad structure is constructed as follows.

$$
\begin{tikzcd}
\Sigma \arrow[r, "\kappa_1"] \arrow[ddr, bend right, "\overset{\triangle}{\eta}_\Sigma"'] & \Sigma + \Sigma^{\triangle} \arrow[d, "\overset{*}{\eta}_{\Sigma + \Sigma^{\triangle}}"] \\
& (\Sigma + \Sigma^{\triangle})^* \arrow[d, "\alpha_\Sigma", "\cong"'] \\
& \Sigma^{\triangle}
\end{tikzcd}
\qquad
\begin{tikzcd}
(\Sigma + \Sigma)^* \arrow[r, dashed, "(\Sigma + \hat{\eta}_\Sigma)^*"] & (\Sigma + \Sigma^{\wedge})^* \\
\Sigma + \Sigma \arrow[u, "\overset{*}{\eta}_{\Sigma + \Sigma}"] & \\
\Sigma \arrow[u, "\kappa_1"] \arrow[r, dashed, "\hat{\eta}_\Sigma"'] & \Sigma^{\wedge} \arrow[uu, "\cong", "\zeta_\Sigma"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
(\Sigma^{\triangle} + \Sigma^{\triangle\triangle})^* \arrow[rr, dashed, "(\Sigma^{\triangle} + \overset{\triangle}{\mu}_{\Sigma})^*"] \arrow[ddd, "\cong"', "\alpha_{\Sigma^{\triangle}}"] & & (\Sigma^{\triangle} + \Sigma^{\triangle})^* \arrow[d, "{[\alpha_{\Sigma}^{-1}, \overset{*}{\eta}_{\Sigma+\Sigma^{\triangle}} \circ \kappa_2]^*}"] \\
& & (\Sigma + \Sigma^{\triangle})^{**} \arrow[d, "\overset{*}{\mu}_{\Sigma+\Sigma^{\triangle}}"] \\
& & (\Sigma + \Sigma^{\triangle})^* \arrow[d, "\cong"', "\alpha_{\Sigma}"] \\
\Sigma^{\triangle\triangle} \arrow[rr, dashed, "\overset{\triangle}{\mu}_{\Sigma}"'] & & \Sigma^{\triangle}
\end{tikzcd}
$$

The definition of $\hat{\mu}$ is rather complicated. Let a_{Σ} be the following composite on the left.

$$
\begin{tikzcd}
\Sigma^{\wedge\wedge} \arrow[r, "\zeta_{\Sigma^{\wedge}}", "\cong"'] \arrow[d, "a_{\Sigma}"'] & (\Sigma^{\wedge} + \Sigma^{\wedge\wedge})^* \arrow[r, "(\zeta_{\Sigma} + \Sigma^{\wedge\wedge})^*", "\cong"'] & \big((\Sigma + \Sigma^{\wedge})^* + \Sigma^{\wedge\wedge}\big)^* \arrow[d, "{[\kappa_1^*, \overset{*}{\eta}_{(\Sigma+\Sigma^{\wedge})+\Sigma^{\wedge\wedge}} \circ \kappa_2]^*}"'] \\
\big(\Sigma + (\Sigma^{\wedge} + \Sigma^{\wedge\wedge})\big)^* & \big((\Sigma + \Sigma^{\wedge}) + \Sigma^{\wedge\wedge}\big)^* \arrow[l, "{[\Sigma + \kappa_1, \kappa_2 \circ \kappa_2]^*}"] & \big((\Sigma + \Sigma^{\wedge}) + \Sigma^{\wedge\wedge}\big)^{**} \arrow[l, "\overset{*}{\mu}_{(\Sigma+\Sigma^{\wedge})+\Sigma^{\wedge\wedge}}"]
\end{tikzcd}
$$

This map a_{Σ} is used in the coalgebraic structure map on the left in:

$$
\begin{tikzcd}
\big(\Sigma + (\Sigma^{\wedge} + \Sigma^{\wedge\wedge})\big)^* \arrow[r, dashed, "(\Sigma + b_{\Sigma})^*"] & (\Sigma + \Sigma^{\wedge})^* \\
\Sigma^{\wedge} + \Sigma^{\wedge\wedge} \arrow[u, "{[(\Sigma + \kappa_1)^* \circ \zeta_{\Sigma},\, a_{\Sigma}]}"] \arrow[r, dashed, "b_{\Sigma}"'] & \Sigma^{\wedge} \arrow[u, "\cong", "\zeta_{\Sigma}"'] \\
\Sigma^{\wedge\wedge} \arrow[u, "\kappa_2"] \arrow[ur, bend right, "\hat{\mu}_{\Sigma}"'] &
\end{tikzcd}
\quad .
$$

By finality one easily obtains $b_{\Sigma} \circ \kappa_1 = \Sigma^{\wedge}$.

A.2 Proof of Lemma 4.8

Points 1 and 2 are straightforward. Point 3 is equivalent to saying that union is preserved by taking an inverse image. For Point 4, we show the proof for $n = 1$. The case for general n is easy by induction. Let $s \in FX$ and $r \in F\overline{\mathcal{P}}Y$. Then

$$
\begin{aligned}
s \in (\lambda_X \circ F\overline{\mathcal{P}}f)(r) &\iff (\, s,\, (F\overline{\mathcal{P}}f)(r)\,) \in \mathrm{Rel}_F(\in_X) && \text{(Definition of } \lambda\text{)} \\
&\iff (s, r) \in (\mathrm{id} \times F\overline{\mathcal{P}}f)^{-1}\, \mathrm{Rel}_F(\in_X) && \\
&\iff (s, r) \in \mathrm{Rel}_F\big((\mathrm{id} \times \overline{\mathcal{P}}f)^{-1}(\in_X)\big) && \text{(Lemma 4.3.3)} \\
&\iff (s, r) \in \mathrm{Rel}_F\big((f \times \mathrm{id})^{-1}(\in_Y)\big) && \text{(}\dagger\text{, see below)} \\
&\iff (\, (Ff)(s), r\,) \in \mathrm{Rel}_F\,(\in_Y) && \text{(Lemma 4.3.3)} \\
&\iff (Ff)(s) \in \lambda_Y(r) && \text{(Definition of } \lambda\text{)} \\
&\iff s \in (\overline{\mathcal{P}}Ff \circ \lambda_Y)(r) \ ,
\end{aligned}
$$

where (†) holds because

$$\begin{aligned}(x,u) \in (\mathrm{id} \times \overline{\mathcal{P}}f)^{-1}(\in_X) &\iff x \in (\overline{\mathcal{P}}f)(u) \\ &\iff f(x) \in u \\ &\iff (x,u) \in (f \times \mathrm{id})^{-1}(\in_Y) \ . \end{aligned}$$ □

A.3 Proof of Theorem 4.10

First we show that, for each n,

$$\mathrm{Im}\,\sigma_n \cap \mathsf{trace}_c^{n+1}(x) = \mathsf{trace}_c^n(x) \ . \tag{8}$$

It is proved as follows.

$$\begin{aligned}
&\mathrm{Im}\,\sigma_n \cap \mathsf{trace}_c^{n+1}(x) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}\sigma_n \circ \mathsf{trace}_c^{n+1})(x) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_{F0} \circ \overline{\mathcal{P}}\sigma_{n+1} \circ \mathsf{trace}_c^{n+1})(x) && (\sigma_n = \sigma_{n+1} \circ F^n ?_{F0} \text{ by (3)}) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_{F0} \circ \overline{\mathcal{P}}F^{n+1} ?_X \circ c^{n+1})(x) \\
& \qquad (\text{Definition of } \mathsf{trace}_c^{n+1}, \text{ and } \overline{\mathcal{P}}\sigma_n \circ \mathcal{P}\sigma_n = \mathrm{id} \text{ by Lemma 4.8.1}) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_{FX} \circ c^{n+1})(x) && (F?_X \circ ?_{F0} = ?_{FX}) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_{FX} \circ \textstyle\bigcup_{F^{n+1}X} \circ \mathcal{P}\lambda^n_{FX} \circ \mathcal{P}F^n c \circ c^n)(x) \\
&= (\mathcal{P}\sigma_n \circ \textstyle\bigcup_{F^n 0} \circ \mathcal{P}\lambda^n_0 \circ \mathcal{P}F^n\overline{\mathcal{P}}?_{FX} \circ \mathcal{P}F^n c \circ c^n)(x) && (\text{Lemma 4.8.3,4}) \\
&= (\mathcal{P}\sigma_n \circ \textstyle\bigcup_{F^n 0} \circ \mathcal{P}\lambda^n_0 \circ \mathcal{P}F^n\overline{\mathcal{P}}?_{X} \circ \mathcal{P}F^n \{-\}_X \circ c^n)(x) \\
& \qquad (\overline{\mathcal{P}}?_{FX} \circ c = \overline{\mathcal{P}}?_X \circ \{-\}_X : X \to \mathcal{P}0, \text{ with terminal codomain } 1 = \mathcal{P}0) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_X \circ \textstyle\bigcup_{F^n X} \circ \mathcal{P}\lambda^n_X \circ \mathcal{P}F^n \{-\}_X \circ c^n)(x) && (\text{Lemma 4.8.4,3}) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_X \circ \textstyle\bigcup_{F^n X} \circ \mathcal{P}\{-\}_{F^n X} \circ c^n)(x) \\
& && (\lambda^n \text{ is compatible with the unit of } \mathcal{P}) \\
&= (\mathcal{P}\sigma_n \circ \overline{\mathcal{P}}F^n ?_X \circ c^n)(x) && (\text{Unit law of the monad } \mathcal{P}) \\
&= \mathsf{trace}_c^n(x) \ .
\end{aligned}$$

Obviously the sequence $\{\mathrm{Im}\,\sigma_n\}_{n<\omega}$ of subsets of A is increasing, since $\sigma_n = \sigma_{n+1} \circ F^n ?_{F0}$. Now, for arbitrary $n \leq m$,

$$\begin{aligned}
\mathrm{Im}\,\sigma_n \cap \mathsf{trace}_c^m(x) &= \mathrm{Im}\,\sigma_n \cap \mathrm{Im}\,\sigma_{n+1} \cap \cdots \cap \mathrm{Im}\,\sigma_{m-1} \cap \mathsf{trace}_c^m(x) \\
& \qquad (\text{Since } \mathrm{Im}\,\sigma_n \subseteq \mathrm{Im}\,\sigma_{n+1} \subseteq \cdots \subseteq \mathrm{Im}\,\sigma_{m-1}) \\
&= \mathrm{Im}\,\sigma_n \cap \mathrm{Im}\,\sigma_{n+1} \cap \cdots \cap \mathsf{trace}_c^{m-1}(x) && (\text{By (8)}) \\
&= \cdots \\
&= \mathsf{trace}_c^n(x) \ .
\end{aligned}$$ □