Open access • Book Chapter • DOI:10.1007/11548133_14

Context-free languages via coalgebraic trace semantics — Source link

Ichiro Hasuo, Bart Jacobs

Institutions: Radboud University Nijmegen

Published on: 03 Sep 2005 - Conference on Algebra and Coalgebra in Computer Science

Topics: Denotational semantics, Coalgebra, Category of sets, Initial algebra and Monad (functional programming)

Context-free Languages via Coalgebraic Trace Semantics

Ichiro Hasuo and Bart Jacobs
Department of Computer Science, Radboud University
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}

January, 2005

Abstract

In this paper we identify context-free grammars as coalgebras. To obtain the associated context-free languages (consisting of only *finite-length* strings) we introduce a general and novel technique of *finite trace semantics* for coalgebras. It builds on top of the (possibly infinite) trace semantics introduced earlier by the second author, but extracts only finite behavior. Interestingly the finite trace is uniquely characterized corecursively and hence it yields a final coalgebra in a suitable Kleisli category, while the ordinary trace is not unique and yields a weakly final coalgebra. 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.

Keywords: coalgebra, coinduction, context-free grammar, formal language, monad

Classification: 18C50, 68Q70 (AMS 2000); D.3.1, F.4.2, F.4.3 (ACM CR 1998)

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 [Jac05, 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): usually the final coalgebra consists of all the (possibly) infinite behavior.

We give a solution by presenting the novel technique of *finite trace semantics* for coalgebras. It builds on (ordinary, possibly infinite) trace semantics from [Jac04b] but the domain of the semantics is now the powerset of the initial algebra, not that of the final coalgebra.

It is shown that finite trace semantics is uniquely determined by a corecursive characterization while the ordinary one is not. In other words, the finite trace yields a final coalgebra in a suitable Kleisli category (of sets and relations), while the ordinary one yields a weakly final coalgebra (without uniqueness).

The paper is organized as follows. Section 2 formulates context-free grammars as coalgebras, and introduces the notion of *skeletal parse trees (SPTs)* which correspond to parsed strings and which will be used instead of (flat) strings. It is shown in Section 3 that (finite) SPTs carry the initial algebra/final coalgebra for an appropriate functor, and that their formation has a monad structure. Section 4 reviews coalgebraic trace semantics from [Jac04b]. Then our main technical result is presented and applied to context-free grammars in Section 5. 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. [Rut00, BW83].

Notations. The i-th coprojection into a coproduct is denoted by κ_i.

The set $X^* = \coprod_{n<\omega} X^n$ is so-called a *Kleene star*, consisting of all the *strings* of finite length over X. It is standard that the mapping $(-)^* : X \mapsto X^*$ has a monad structure with unit $\overset{*}{\eta}$ creating strings of length one and multiplication $\overset{*}{\mu}$ "flattening" a string of strings into a string (*Kleene monad*).

We designate by a symbol on top of η or μ for which monad it works as a unit or multiplication. For example $\overset{*}{\eta}_\Sigma$ is the Σ-component of the unit of Kleene monad, and when mapped by the functor $(-)^*$ it is denoted by $(\overset{*}{\eta}_\Sigma)^*$.

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 of Y to its inverse image.

We will also use another notation $\coprod$ for direct images, especially for relations: for a relation $R \rightarrowtail X_1 \times X_2$ and functions $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$,

$$\coprod_{f_1 \times f_2}(R) = \{ \langle f_1(x_1) , f_2(x_2) \rangle \mid \langle x_1, x_2 \rangle \in R \} \enspace .$$

The empty string (of length zero) is denoted by ε.

2 Context-free grammars

In this section we formulate context-free grammars and context-free languages, from a coalgebraic perspective. For more about traditional treatment of those notions the reader is referred to [LP81]. In fact, what we are interested in here is not a *language*, i.e. a set of (flat) strings, but a set of parsed strings equipped with a tree structure, called *skeletal parse trees* [ASU86].

A *context-free grammar (CFG for short)* 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*. Using a common trick of "relation-into-function", we have the following coalgebraic presentation of a CFG.

$$\begin{tikzcd}
X \arrow[r, "g"] & \mathcal{P}\big((\Sigma + X)^*\big) \\
x \arrow[r, mapsto] & \{s \in (\Sigma + X)^* \mid \langle x, s\rangle \in R\}
\end{tikzcd}$$

This gives a bijective correspondence between coalgebras of the functor $\mathcal{P}\big((\Sigma + -)^*\big) : \mathbf{Sets} \to \mathbf{Sets}$ and context-free grammars with its terminal symbols from Σ. Here non-terminal symbols correspond to states of the coalgebra.

Example 2.1. Consider the following context-free grammar meant for the syntax of Peano arithmetic, where $\langle x, w\rangle \in R$ is denoted by $x \overset{R}{\to} w$:

$$\Sigma = \{\mathsf{0}, \mathsf{s}, =, \wedge, \vee, \supset, \neg, \forall, \exists\} \cup \mathit{Var}, \qquad X = \{\mathbf{T}, \mathbf{Q}, \mathbf{F}\},$$

$$\mathbf{T} \overset{R}{\to} \mathsf{0}, \qquad \mathbf{T} \overset{R}{\to} \mathsf{x} \quad (\mathsf{x} \in \mathit{Var}), \qquad \mathbf{T} \overset{R}{\to} \mathsf{s}\mathbf{T},$$

$$\mathbf{Q} \overset{R}{\to} \forall\mathsf{x} \quad (\mathsf{x} \in \mathit{Var}), \qquad \mathbf{Q} \overset{R}{\to} \exists\mathsf{x} \quad (\mathsf{x} \in \mathit{Var}),$$

$$\mathbf{F} \overset{R}{\to} \mathbf{T}{=}\mathbf{T}, \quad \mathbf{F} \overset{R}{\to} \mathbf{F}{\wedge}\mathbf{F}, \quad \mathbf{F} \overset{R}{\to} \mathbf{F}{\vee}\mathbf{F}, \quad \mathbf{F} \overset{R}{\to} \mathbf{F}{\supset}\mathbf{F}, \quad \mathbf{F} \overset{R}{\to} \neg\mathbf{F}, \quad \mathbf{F} \overset{R}{\to} \mathbf{QF}.$$

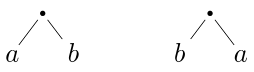
The corresponding coalgebraic presentation $g : X \to \mathcal{P}\big((\Sigma + X)^*\big)$ is as follows.

$$\begin{aligned}
g(\mathbf{T}) &= \{\mathsf{0}\} \cup \mathit{Var} \cup \{\mathsf{s}\mathbf{T}\}, \\
g(\mathbf{Q}) &= \{\forall\mathsf{x} \mid \mathsf{x} \in \mathit{Var}\} \cup \{\exists\mathsf{x} \mid \mathsf{x} \in \mathit{Var}\}, \\
g(\mathbf{F}) &= \{\mathbf{T}{=}\mathbf{T}, \mathbf{F}{\wedge}\mathbf{F}, \mathbf{F}{\vee}\mathbf{F}, \mathbf{F}{\supset}\mathbf{F}, \neg\mathbf{F}, \mathbf{QF}\}.
\end{aligned}$$

Definition 2.2 (Context-free grammars). Throughtout this paper we refer to a coalgebra for a functor $\mathcal{P}\big((\Sigma + -)^*\big)$ as a *context-free grammar (CFG) over* Σ.

Usually a context-free grammar over Σ is considered as a machine which generates strings of letters from Σ, i.e. elements of Σ^*. However, as noted before, here we prefer to work on skeletal parse trees rather than flat strings.

Remark 2.3. By a *tree* we mean an *ordered tree*; it is rooted and the order of the subtrees matters, so we distinguish the following two.



We will refer to the order of the subtrees as “left-to-right”.

Definition 2.4 ((Skeletal) parse trees). Let $g : X \to \mathcal{P}\big((\Sigma + X)^*\big)$ be a context-free grammar over Σ. A *parse tree generated by g from $x \in X$* is a (possibly infinite-depth) tree which satisfy the following:

1. All leaf nodes are labelled from $\Sigma \cup X$;
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 ε belongs to $g(y)$;
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 $c_1 c_2 \ldots c_m$ is an element of $g(y)$.[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 g from x* is a parse tree generated by g 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 context-free grammar g 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 above definition is a bit complicated; the reader may find illustrating an alternative characterization (Proposition 3.1) in terms of initial algebra/final coalgebra.

Remark 2.5. In a parse tree generated by a CFG g, a leaf node labelled with $y \in X$ corresponds to the application of generation rule $y \to \varepsilon$ of g. It is standard [ASU86, Exercise 4.20] that every CFG is effectively equivalent to an *ε-free* CFG, i.e. one without a rule with its right-hand side ε. For ε-free CFGs we can simplify the definition of (skeletal) parse trees that *all* of the leaf nodes are labelled from Σ.

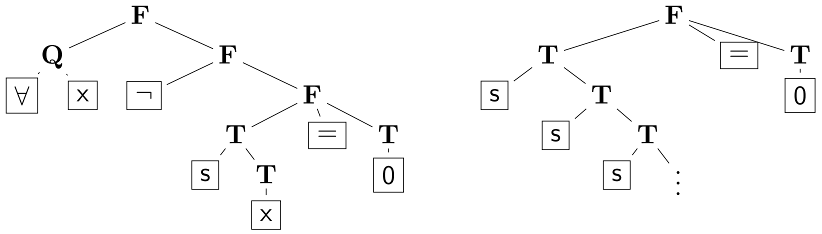
In many cases we are only concerned with finite SPTs, as is illustrated in the next example.

Example 2.6. The following two are parse trees generated by the context-free grammar in Example 2.1, from the non-terminal symbol $\mathbf{F}$. The one on the left

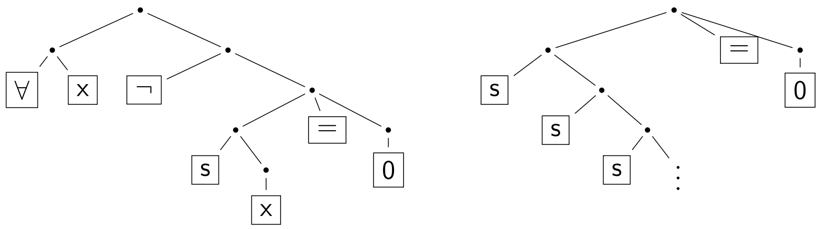
[1] Condition 4 may be considered as an instance of Condition 5 when $m = 0$.

[2] Since an SPT is generated from a non-terminal symbol.

is finite.



Forgetting about the non-terminal symbols from X, we obtain the following SPTs generated by the grammar from **F**.



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

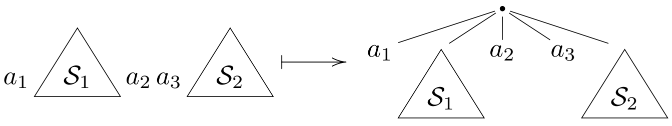
The notion of (finite) SPTs generated by a grammar will be given a coalgebraic characterization via (finite) trace semantics in Sections 4 and 5.

3 (Co)algebraic structures on skeletal parse trees

In this section we investigate (co)algebraic structures on the set of (finite) SPTs, monad structures in their formation, and the "flattening" function mapping SPTs to strings. The following observation is the first step.

Proposition 3.1. *The set of all finite SPTs over Σ, denoted by $\Sigma^{\triangle}$, carries the initial $(\Sigma + -)^*$-algebra. The algebraic structure α_Σ makes a sequence $c_1 c_2 \ldots c_n$ (where $c_i \in \Sigma + \Sigma^{\triangle}$) into a tree by adding a fresh root whose immediate successors are $c_1, c_2, \ldots, c_n$. For example,*

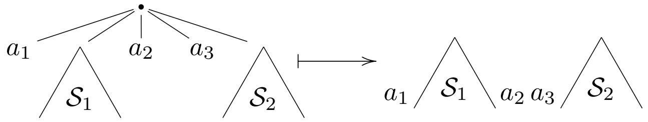
$$(\Sigma + \Sigma^{\triangle})^* \xrightarrow[\cong]{\alpha_\Sigma} \Sigma^{\triangle}$$



and $\alpha_\Sigma(\varepsilon)$ is the trivial tree which is not labelled.

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

$$\begin{tikzcd}
\Sigma^{\wedge} \arrow[rr, "\zeta_\Sigma", "\cong"'] & & (\Sigma + \Sigma^{\wedge})^*
\end{tikzcd}$$



and the trivial tree is mapped to the empty string ε.

Proof. The first part is by induction on the depth. The second part is shown similarly to that Σ^ω carries the final $(\Sigma \times -)$-coalgebra; see e.g. [JR97]. ∎

Remark 3.2. We can think of the functor $(\Sigma + -)^*$ as a "signature" in a traditional sense. Let $\star$ be a fresh symbol, and for each $s \in (\Sigma + \{\star\})^*$ let $\|s\|$ denote the number of $\star$'s appearing in s. Then we have an obvious isomorphism

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

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

Each of the mappings $(-)^{\triangle} : \Sigma \mapsto \Sigma^{\triangle}$ and $(-)^{\wedge} : \Sigma \mapsto \Sigma^{\wedge}$ extends to a functor as follows. For $f : \Sigma \to \Phi$,

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

This is an instance of standard results about initial algebras and final coalgebras for a functor $F(\Sigma, -)$, where F is a bifunctor.

It turns out that both $(-)^{\triangle}$ and $(-)^{\wedge}$ have a monad structure.[4] 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

[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 $\Sigma^{\triangle}$.

[4] The notation $(-)^{\triangle}$ and $(-)^{\wedge}$ is used to suggest an analogy with the Kleene (or *list*, *string*) monad $(-)^*$. Indeed, these constructions are closely related.

is that here the parameter set Σ is inside the Kleene monad $(-)^*$, which adds some complexity.

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

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

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

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

This map a_Σ is used in the coalgebraic structure map on the left in:

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

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

It is straightforward, but laborious, to show that the constructions above satisfy the requirements of a monad.

Proposition 3.3. *The constructions* $\langle(-)^{\triangle}, \overset{\triangle}{\eta}, \overset{\triangle}{\mu}\rangle$ *and* $\langle(-)^{\wedge}, \hat{\eta}, \hat{\mu}\rangle$ *are indeed monads.* ▮

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].

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

Again it is straightforward to show that ι_Σ is natural in Σ, and makes the following diagrams commute.

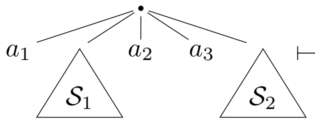
$$\begin{tikzcd}
\Sigma \arrow[r, "\overset{\triangle}{\eta}_\Sigma"] \arrow[dr, "\hat{\eta}_\Sigma"'] & \Sigma^{\triangle} \arrow[d, "\iota_\Sigma"] \\
& \Sigma^{\wedge}
\end{tikzcd}
\qquad
\begin{tikzcd}
\Sigma^{\triangle\triangle} \arrow[r, "\overset{\triangle}{\mu}_\Sigma"] \arrow[d, "(\iota \circ \iota)_\Sigma"'] & \Sigma^{\triangle} \arrow[d, "\iota_\Sigma"] \\
\Sigma^{\wedge\wedge} \arrow[r, "\hat{\mu}_\Sigma"'] & \Sigma^{\wedge}
\end{tikzcd}$$

Proposition 3.4. *The embedding* $\iota : (-)^{\triangle} \Rightarrow (-)^{\wedge}$ *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:

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

The definition says that φ_Σ maps the trivial (one-node) tree to the empty string ε, and for example



$$\xmapsto{\varphi_\Sigma} a_1 \big(\varphi_\Sigma(\mathcal{S}_1)\big)\, a_2\, a_3 \big(\varphi_\Sigma(\mathcal{S}_2)\big) \ ,$$

as desired. The next result is again straightforward.

Proposition 3.5. *The flattening map* $\varphi : (-)^{\triangle} \Rightarrow (-)^*$ *is a map of monads.* ▮

Remark 3.6. It is not clear whether we have an infinite version of φ_Σ, that is, a flattening map which takes a possibly infinite SPT to a possibly infinite string. It is standard [JR97] that the set $\Sigma^\infty = \Sigma^* + \Sigma^\omega$ of possibly infinite strings over Σ carries the final coalgebra for the functor $(1 + \Sigma \times -)$. Hence the key to obtaining a flattening map $\Sigma^\wedge \to \Sigma^\infty$ would be a suitable $(1+\Sigma\times-)$-coalgebra structure over $\Sigma^\wedge$. How to define it is not obvious.

4 Trace semantics for coalgebras

In this section we sketch and slightly improve the technique of *trace semantics* for coalgebras from [Jac04b]. This scheme gives a coalgebraic characterization of (possibly infinite) SPTs generated by a context-free grammar, and also forms a basis of *finite trace semantics* which is used for finite SPTs.

We will give trace semantics for $\mathcal{P}F$-coalgebras, where F is in the family of endofunctors on **Sets** defined by the following BNF notation:

$$F \quad ::= \quad I_{\mathbf{Sets}} \mid K_\Sigma \mid F \times F \mid \coprod_{n<\omega} F_n \mid F^\Sigma \enspace .$$

The functor K_Σ is a constant functor into a set Σ. This family is that of polynomial functors extended by countable coproducts,[5] and contains the functor $F = (\Sigma + -)^* = \coprod_{n<\omega}(\Sigma + -)^n$ which makes a $\mathcal{P}F$-coalgebra a context-free grammar.

It is easy to show that such functors F preserve ω-colimits and ω^{op}-limits. Therefore such F have both an initial algebra and a final coalgebra (see e.g. [Bar93, AK95]) which are, together with the canonical embedding, denoted as follows.

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

We will look at their construction in detail later, when it is necessary.

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.

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

Since a functor F considered here preserves weak pullbacks, we obtain the following standard compatibility results which are used heavily throughout this paper.

[5] The results in [Jac04b] are presented for polynomial functors with only finite coproducts, but they immediately generalize to this family of functors.

Lemma 4.1. *Relation liftings are compatible with such operations on relations as:*

1. *Equality:* $\mathrm{Rel}_F(=_X)$ *is equal to* $=_{FX}$*.*
2. *Composition: for* $R \rightarrowtail X \times Y$*,* $S \rightarrowtail Y \times Z$ *and their composition* $S \circ R = \{\langle x, z\rangle \in X \times Z \mid \exists y \in Y. \langle x, y\rangle \in R$ *and* $\langle y, z\rangle \in Z\}$ *we have*
$$\mathrm{Rel}_F(S \circ R) = \mathrm{Rel}_F(S) \circ \mathrm{Rel}_F(R) \ .$$
3. *Inclusion: if* $R \subseteq S$ *then* $\mathrm{Rel}_F(R) \subseteq \mathrm{Rel}_F(S)$*.*
4. *Opposite relation: for* $R \rightarrowtail X \times Y$ *and its opposite* $R^{\mathrm{op}} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R\}$ *we have*
$$\mathrm{Rel}_F(R^{\mathrm{op}}) = \big(\mathrm{Rel}_F(R)\big)^{\mathrm{op}}.$$
5. *Graph of a function and functor application: for a function* $f : X \to Y$ *and its graph* $\mathcal{G}_f = \{\langle x, f(x)\rangle \mid x \in X\}$ *we have*
$$\mathrm{Rel}_F(\mathcal{G}_f) = \mathcal{G}_{Ff} \ .$$
6. *Inverse image and direct image: for functions* $f_1 : X_1 \to Y_1$*,* $f_2 : X_2 \to Y_2$ *and relations* $R \rightarrowtail X_1 \times X_2$*,* $S \rightarrowtail Y_1 \times Y_2$ *we have*
$$\begin{aligned}&\mathrm{Rel}_F\big((f_1 \times f_2)^{-1}(S)\big) = (Ff_1 \times Ff_2)^{-1}\big(\mathrm{Rel}_F(S)\big) \ , \\ &\mathrm{Rel}_F\big(\textstyle\coprod_{f_1 \times f_2}(R)\big) = \coprod_{Ff_1 \times Ff_2}\big(\mathrm{Rel}_F(R)\big) \ .\end{aligned}$$ ∎

Points 4 and 5 are not presented in [Jac04b] but immediately follow from the definition.

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

$$\begin{tikzcd}
F\mathcal{P}X \arrow[r, "\lambda_X"] & \mathcal{P}FX \\
u \arrow[r, mapsto] & \{a \in FX \mid \langle a, u\rangle \in \mathrm{Rel}_F(\in_X)\}
\end{tikzcd}$$

Then the map λ_X is

- natural in X, and
- 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*.[6]

Lemma 4.2 ([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 4.3. Here we are interested in the case where $F = (\Sigma + -)^*$. 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 $c_i \in d_i$.

Definition 4.4 (Traces of coalgebras). Let $g : X \to \mathcal{P}FX$ be a $\mathcal{P}F$-coalgebra and $\zeta : Z \overset{\cong}{\to} FZ$ be the final F-coalgebra. A map $t : X \to \mathcal{P}Z$ is said to be a *trace of* g if it makes the following diagram commute.

$$\begin{tikzcd}
\mathcal{P}FX \arrow[r, "\mathcal{P}Ft"] & \mathcal{P}F\mathcal{P}Z \arrow[d, "\mathcal{P}\lambda_Z"] \\
 & \mathcal{P}^2FZ \arrow[d, "\bigcup_{FZ}"] \\
 & \mathcal{P}FZ \\
X \arrow[uuu, "g"] \arrow[r, "t"'] & \mathcal{P}Z \arrow[u, "\mathcal{P}\zeta", "\cong"']
\end{tikzcd} \qquad (1)$$

That is, for each $x \in X$,

$$\{\zeta(z) \mid z \in t(x)\} = \bigcup \{\, (\lambda_Z \circ Ft)(y) \mid y \in g(x)\} \ .$$

Remark 4.5. The Kleisli category $\mathbf{Sets}_{\mathcal{P}}$ of the powerset monad $\mathcal{P}$ has sets as objects and relations as arrows, via "relation-into-function" trick. In this category the definition of traces becomes clearer [Jac04b, Section 5]. The distributive law $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ gives rise to a lifting $F_{\mathcal{P}} : \mathbf{Sets}_{\mathcal{P}} \to \mathbf{Sets}_{\mathcal{P}}$ of a functor F by

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

One can readily show that the diagram (1) in **Sets** commutes if and only if the following diagram in $\mathbf{Sets}_{\mathcal{P}}$ commutes:

$$\begin{tikzcd}
F_{\mathcal{P}}X \arrow[r, "F_{\mathcal{P}}t"] & F_{\mathcal{P}}Z \\
X \arrow[u, "g"] \arrow[r, "t"'] & Z \arrow[u, "\{-\}_{FZ} \circ \zeta"']
\end{tikzcd} ,$$

which says that t is a morphism of $F_{\mathcal{P}}$-coalgebras.

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

Theorem 4.6 (Main result of [Jac04b]). *Let $g : X \to \mathcal{P}FX$ a coalgebra.*

1. *A trace of g always exists, but need not be unique. In other words, the coalgebra $\{-\}_{FZ} \circ \zeta : Z \to F_{\mathcal{P}}Z$ is weakly final in $\mathbf{CoAlg}(F_{\mathcal{P}})$.*

2. *There is a canonical choice t_0 of a trace of g, namely the largest one.*[7] ∎

For later uses we sketch the construction of the largest trace t_0. Let the relations $R_n \rightarrowtail F^{n+1}X \times F^n X$ be defined inductively by

$$R_0 = (1 \times g)^{-1}(\in_{FX}) \ , \qquad R_{n+1} = \mathrm{Rel}_F(R_n) \ ,$$

and the set U be defined by

$$U = \{u \in \prod_{n<\omega} F^n X \mid \forall n < \omega.\, \langle u_{n+1}, u_n \rangle \in R_n\} \ .$$

As presented in [Bar93], the carrier A of the initial F-algebra is obtained as a colimit (denoted by σ below) of the initial sequence of F, and the carrier Z of the final F-coalgebra is as a limit (denoted by τ) of the terminal sequence.

$$\begin{tikzcd}
 & & & & & & & A \arrow[dddd, dashed, bend left, "\iota"] \\
0 \arrow[r, "h"] \arrow[d, "i"'] \arrow[urrrrrrr, bend left, "\sigma_0"] & F0 \arrow[r, "Fh"] \arrow[d, "Fi"] \arrow[urrrrrr, bend left, "\sigma_1"] & F^2 0 \arrow[r] \arrow[d, "F^2 i"] \arrow[urrrrr, bend left, "\sigma_2"] & \cdots \arrow[r] & F^n 0 \arrow[r, "F^n h"] \arrow[d, "F^n i"] \arrow[urrr, bend left, "\sigma_n"] & F^{n+1} 0 \arrow[r] \arrow[d, "F^{n+1} i"] \arrow[urr, bend left, "\sigma_{n+1}"] & \cdots & \\
X \arrow[d, "j"'] & FX \arrow[d, "Fj"] & F^2 X \arrow[d, "F^2 j"] & \cdots & F^n X \arrow[d, "F^n j"] & F^{n+1} X \arrow[d, "F^{n+1} j"] & \cdots & \\
1 & F1 \arrow[l, "k"] & F^2 1 \arrow[l, "Fk"] & \cdots \arrow[l] & F^n 1 \arrow[l] & F^{n+1} 1 \arrow[l, "F^n k"] & \cdots \arrow[l] & \\
 & & & & & & & Z \arrow[ulllllll, bend left, "\tau_0"] \arrow[ullllll, bend left, "\tau_1"] \arrow[ulllll, bend left, "\tau_2"] \arrow[ulll, bend left, "\tau_n"] \arrow[ull, bend left, "\tau_{n+1}"]
\end{tikzcd}$$

Here the arrows h, i, j, k are induced by the universality of 0 and 1. The embedding ι is obtained by the successive use of universality of the limit τ and that of the colimit σ; equivalently ι is characterized as the unique arrow which makes the diagram commute, i.e. for each $n < \omega$

$$F^n j \circ F^n i = \tau_n \circ \iota \circ \sigma_n \ .$$

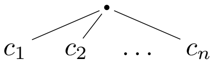
Then the largest trace t_0 of g is described as

$$t_0(x) = \{ \bigl(\prod_{n<\omega} F^n j\bigr)(u) \mid u \in U \wedge u_0 = x\} \ .$$

[7] In terms of pointwise inclusion, i.e. $t(x) \subseteq t_0(x)$ for every trace t and every $x \in X$.

The construction of limits via products and equalizers says that the carrier Z consists of those elements of $\prod_{n<\omega} F^n 1$ which satisfy the *chain property* [Jac04b, Remark 6.4]; the definition of U ensures that $\bigl(\prod_{n<\omega} F^n j\bigr)(u)$ indeed satisfy this property and hence $t_0(x)$ is well-defined as an element of $\mathcal{P}Z$.

Example 4.7. We apply the results to context-free grammars. For the functor $F = (\Sigma + -)^*$ the definition of a trace is interpreted as follows, using Example 4.3. The map $t : X \to \mathcal{P}\Sigma^\wedge$ is a trace of a context-free grammar $g : X \to \mathcal{P}\bigl((\Sigma + X)^*\bigr)$ if and only if for each element



of $t(x)$ (here $c_1, c_2, \ldots, c_n \in \Sigma + \Sigma^\wedge$), there exists a string $d_1 d_2 \ldots d_n \in g(x)$ such that for each $i = 1, 2, \ldots, n$:

- if $c_i \in \Sigma$ then d_i is also in Σ and $c_i = d_i$;
- if $c_i \in \Sigma^\wedge$ then d_i is in X and $c_i \in t(d_i)$.

Although this definition would seem to determine a trace $t(x)$ uniquely as the set of all the generated SPTs in the sense of Definition 2.4, Theorem 4.6 says that it is not always the case. However a close look at the construction of t_0 reveals that the *largest* trace t_0 gives the set of all the generated SPTs.

5 Finite trace semantics for coalgebras

In this section we use the approach from the last section to obtain *finite trace semantics* for coalgebras. This gives the set of generated finite SPTs (which is usually of our interest, see Example 2.6) when applied to a context-free grammar. The finite trace is unique, differently from ordinary traces. The approach here makes an initial algebra (in the category of sets and functions) into a final coalgebra (in the category of sets and relations).

Definition 5.1 (Finite traces of coalgebras). Let $g : X \to \mathcal{P}FX$ be a $\mathcal{P}F$-coalgebra and $\alpha : FA \xrightarrow{\cong} A$ be the initial F-algebra. A map $f : X \to \mathcal{P}A$ is said to be a *finite trace of* g if it makes the following diagram commute.

In **Sets**,

$$\begin{tikzcd}
\mathcal{P}FX \arrow[r, dashed, "\mathcal{P}Ff"] & \mathcal{P}F\mathcal{P}A \arrow[d, "\mathcal{P}\lambda_A"] \\
& \mathcal{P}^2FA \arrow[d, "\bigcup_{FA}"] \\
& \mathcal{P}FA \\
X \arrow[uuu, "g"] \arrow[r, dashed, "f"'] & \mathcal{P}A \arrow[u, "\mathcal{P}\alpha^{-1}", "\cong"']
\end{tikzcd}$$
,

equivalently (Remark 4.5), in $\mathbf{Sets}_{\mathcal{P}}$,

$$\begin{tikzcd}
F_{\mathcal{P}}X \arrow[r, dashed, "F_{\mathcal{P}}f"] & F_{\mathcal{P}}A \\
X \arrow[u, "g"] \arrow[r, dashed, "f"'] & A \arrow[u, "\{-\}_{FA} \circ \alpha^{-1}"']
\end{tikzcd}$$
.

That is, for each $x \in X$, $\{\alpha^{-1}(a) \mid a \in f(x)\} = \bigcup \{\, (\lambda_A \circ Ff)(y) \mid y \in g(x)\}$. It is shown later (Corollary 5.9) that such f uniquely exists.

First we start with observing (Proposition 5.3) that any trace gives rise to a finite trace, and vice versa, using the following preliminaries. Recall that $\overline{\mathcal{P}}$ denotes the contravariant powerset functor.

Lemma 5.2. *1. For a mono $m : X \rightarrowtail Y$, $\mathcal{P}m$ is a split mono with its left inverse $\overline{\mathcal{P}}m$.*

2. For an epi $e : X \twoheadrightarrow Y$, $\mathcal{P}e$ is a split epi with right inverse $\overline{\mathcal{P}}e$.

$$\begin{tikzcd}
\mathcal{P}X \arrow[r, tail, "\mathcal{P}m"] \arrow[dr, equal] & \mathcal{P}Y \arrow[d, "\overline{\mathcal{P}}m"] \\
& \mathcal{P}X
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{P}X \arrow[r, tail, "\overline{\mathcal{P}}e"] \arrow[dr, equal] & \mathcal{P}Y \arrow[d, "\mathcal{P}e"] \\
& \mathcal{P}X
\end{tikzcd}$$

3. For an iso $i : X \xrightarrow{\cong} Y$, $\mathcal{P}i$ is also an iso with its inverse $\overline{\mathcal{P}}i$.

4. The union maps $\bigcup_X : \mathcal{P}^2 X \to \mathcal{P}X$ form a natural transformation $\mathcal{P}\overline{\mathcal{P}} \Rightarrow \overline{\mathcal{P}}$.

5. The maps $\lambda_X : F\mathcal{P}X \to \mathcal{P}FX$ in Lemma 4.2 form a natural transformation $F\overline{\mathcal{P}} \Rightarrow \overline{\mathcal{P}}F$.

To put Points 4 and 5 diagramatically: for each $f : X \to Y$ the following two squares commute.

$$\begin{tikzcd}
\mathcal{P}\overline{\mathcal{P}}Y \arrow[r, "\bigcup_Y"] \arrow[d, "\mathcal{P}\overline{\mathcal{P}}f"'] & \mathcal{P}Y \arrow[d, "\overline{\mathcal{P}}f"] \\
\mathcal{P}\overline{\mathcal{P}}X \arrow[r, "\bigcup_X"'] & \mathcal{P}X
\end{tikzcd}
\qquad
\begin{tikzcd}
F\overline{\mathcal{P}}Y \arrow[r, "\lambda_Y"] \arrow[d, "F\overline{\mathcal{P}}f"'] & \overline{\mathcal{P}}FY \arrow[d, "\overline{\mathcal{P}}Ff"] \\
F\overline{\mathcal{P}}X \arrow[r, "\lambda_X"'] & \overline{\mathcal{P}}FX
\end{tikzcd}$$

Proof. Points 1 and 2 are straightforward, from which Point 3 follows. Point 4 is equivalent to saying that union is preserved by taking an inverse image. For Point 5, let $s \in FX$ and $r \in F\overline{\mathcal{P}}Y$. Then

$$\begin{aligned}
s \in (\lambda_X \circ F\overline{\mathcal{P}}f)(r) &\iff \langle\, s,\, (F\overline{\mathcal{P}}f)(r)\,\rangle \in \mathrm{Rel}_F(\in_X) \\
&\iff \langle s, r\rangle \in (1 \times F\overline{\mathcal{P}}f)^{-1}\, \mathrm{Rel}_F(\in_X) \\
&\iff \langle s, r\rangle \in \mathrm{Rel}_F\big((1 \times \overline{\mathcal{P}}f)^{-1}(\in_X)\big) && \text{(Lemma 4.1)} \\
&\iff \langle s, r\rangle \in \mathrm{Rel}_F\big((f \times 1)^{-1}(\in_Y)\big) && (2) \\
&\iff \langle\, (Ff)(s), r\,\rangle \in \mathrm{Rel}_F\,(\in_Y) && \text{(Lemma 4.1)} \\
&\iff (Ff)(s) \in \lambda_Y(r) \\
&\iff s \in (\overline{\mathcal{P}}Ff \circ \lambda_Y)(r) \;\; ,
\end{aligned}$$

where (2) holds because

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

Recall that $\iota : A \to Z$ is the canonical embedding of the initial F-algebra in the final F-coalgebra.

Proposition 5.3. *For a coalgebra $g : X \to \mathcal{P}FX$, a trace t gives rise to a finite trace by*

$$X \xrightarrow{t} \mathcal{P}Z \xrightarrow{\overline{\mathcal{P}}\iota} \mathcal{P}A \quad ,$$

and a finite trace f gives rise to an (ordinary) trace by

$$X \xrightarrow{f} \mathcal{P}A \xrightarrow{\mathcal{P}\iota} \mathcal{P}Z \quad .$$

Proof. It suffices to show that the following diagram commute: for the first part of the proposition take the three squares on the left and put them on the right of the definition of a trace, and for the second part take those on the right.

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

Square (i) commutes by Lemma 5.2.5, (ii) by Lemma 5.2.4, (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. ∎

We will further investigate the *smallest* one among ordinary, possibly infinite traces (Theorem 5.8), which together with the previous proposition yields the uniqueness of a finite trace. To that goal we need some preliminaries. In the following we will heavily use the empty relation $\emptyset$, whose subscript designates between which sets the empty relation resides: for example $\emptyset_{C \times D} \rightarrowtail C \times D$.

Lemma 5.4. *For arbitrary sets X_1, X_2 and every $n < \omega$ we have*

$$\mathrm{Rel}_F^n(\emptyset_{X_1 \times X_2}) = \coprod\nolimits_{F^n !_{X_1} \times F^n !_{X_2}} (=_{F^n 0}) \ .$$

Proof. By induction on n. When $n = 0$ both sides of the equality are the empty relation $\emptyset_{X_1 \times X_2}$. For the step case use Lemma 4.1. ∎

Definition 5.5. For each $n < \omega$, a natural transformation

$$\lambda^n : F^n \mathcal{P} \Rightarrow \mathcal{P} F^n$$

is defined inductively by: $\lambda^0_X = 1_{\mathcal{P}X}$ and

$$\begin{tikzcd}
F^{n+1}\mathcal{P}X \arrow[r, "F^n \lambda_X"] \arrow[d, "F\lambda^n_X"'] \arrow[dr, "\lambda^{n+1}_X" description] & F^n \mathcal{P} F X \arrow[d, "\lambda^n_{FX}"] \\
F\mathcal{P}F^n X \arrow[r, "\lambda_{F^n X}"'] & \mathcal{P}F^{n+1}X
\end{tikzcd} .$$

It is easily shown by induction that the outer square commutes.

For a map $t : X \to \mathcal{P}Z$ (meant to be a trace) and each $n < \omega$, we define a map $t^n : F^n X \to \mathcal{P} F^n Z$ by $t^n = \lambda^n_Z \circ F^n t$.

Lemma 5.6 ([Jac04b], Lemma 4.2). *For each $n < \omega$ and each set X we have a characterization of $\mathrm{Rel}^n_F(\in_X)$ via the natural transformation λ^n:*

$$\mathrm{Rel}^n_F(\in_X) = (1 \times \lambda^n_X)^{-1}(\in_{F^n X}) \ . \qquad ∎$$

Recall that a coalgebra $g : X \to \mathcal{P}FX$ induces a relation

$$R_i = \mathrm{Rel}^i_F\big((1 \times g)^{-1}(\in_{FX})\big) \quad \rightarrowtail \quad F^{i+1}X \times F^i X$$

for each $i < \omega$, as defined in the construction of Theorem 4.6.

Lemma 5.7. *Let $t : X \to \mathcal{P}Z$ be an arbitrary trace of a coalgebra $g : X \to \mathcal{P}FX$, and $z \in Z$. The following is sufficient to have $z \in t(x)$:*

> *There exists a sequence of finite length*
>
> $$u_0 \in X,\ u_1 \in FX,\ \ldots,\ u_n \in F^n X$$
>
> *such that:*
>
> 1. $u_0 = x$;
> 2. $\langle u_{i+1}, u_i \rangle \in R_i$ *for each* $i = 0, 1, \ldots, n-1$;
> 3. $(F^{n-1}\zeta \circ F^{n-2}\zeta \circ \cdots \circ F\zeta \circ \zeta)(z) \in t^n(u_n)$.

Proof. By induction on n. When $n = 0$ the claim is obvious. Assume that the claim holds for n, and consider $n+1$. Since t is a trace we have

$$(\mathrm{LHS}) \overset{\mathrm{def}}{=} \{\zeta(z) \mid z \in t(x)\} = \bigcup \{\, (\lambda_Z \circ Ft)(y) \mid y \in g(x)\} \overset{\mathrm{def}}{=} (\mathrm{RHS}) \ .$$

We consider each side of this equality as a relation between FZ and X. Then as presented in the proof of [Jac04b, Theorem 6.5], if each side is repeatedly lifted the following hold:[8]

$$\begin{aligned}\mathrm{Rel}_F^n(\mathrm{LHS}) &= \textstyle\coprod_{F^n\zeta\times 1}\bigl(1\times F^n t\bigr)^{-1}\bigl(1\times\lambda_Z^n\bigr)^{-1}(\in_{F^n Z})\ ,\\ \mathrm{Rel}_F^n(\mathrm{RHS}) &= \{\langle v,w\rangle \mid \exists u\in F^{n+1}X.\ v\in t^{n+1}(u)\ \wedge\ \langle u,w\rangle\in R_n\}\ .\end{aligned}$$

Now our assumption 3 (for $n+1$ instead of n) says that

$$\langle\,(F^n\zeta\circ F^{n-1}\zeta\circ\cdots\circ\zeta)(z)\,,\ u_n\,\rangle\in\mathrm{Rel}_F^n(\mathrm{RHS})\ ,$$

hence replacing (RHS) by (LHS) we obtain

$$\langle\,(F^n\zeta\circ F^{n-1}\zeta\circ\cdots\circ\zeta)(z)\,,\ u_n\,\rangle\ \in\ \textstyle\coprod_{F^n\zeta\times 1}\bigl(1\times F^n t\bigr)^{-1}\bigl(1\times\lambda_Z^n\bigr)^{-1}(\in_{F^n Z})\ ,$$

then using that $F^n\zeta$ is an iso we have

$$\langle\,(F^{n-1}\zeta\circ\cdots\circ\zeta)(z)\,,\ (\lambda_Z^n\circ F^n t)(u_n)\,\rangle\in\bigl(\in_{F^n Z}\bigr)\ .$$

By the definition of t^n we have

$$(F^{n-1}\zeta\circ\cdots\circ\zeta)(z)\ \in\ t^n(u_n)\ ,$$

which allows us to use the induction hypothesis to conclude that $z\in t(x)$. ∎

Theorem 5.8. *Let $g: X\to\mathcal{P}FX$ be a coalgebra and $t: X\to\mathcal{P}Z$ be an arbitrary (possibly infinite) trace of g. Then the trace*

$$\begin{tikzcd} X \arrow[r, "t"] & \mathcal{P}Z \arrow[r, two heads, "\overline{\mathcal{P}}\iota"] & \mathcal{P}A \arrow[r, tail, "\mathcal{P}\iota"] & \mathcal{P}Z \end{tikzcd}$$

is the smallest among all traces of g.

Proof. Proposition 5.3 yields that the composite is indeed a trace. To show that it is the smallest one, we can take the largest trace t_0 as a trace t above, since $\overline{\mathcal{P}}\iota$ and $\mathcal{P}\iota$ are both monotonic .

The following general fact is shown straightforwardly: given a map $l: C\to D$, the map $\mathcal{P}l\circ\overline{\mathcal{P}}l$ is such that

$$\begin{tikzcd} \mathcal{P}D \arrow[r, "\overline{\mathcal{P}}l"] & \mathcal{P}C \arrow[r, "\mathcal{P}l"] & \mathcal{P}D \\ u \arrow[rr, mapsto] & & u\cap\mathrm{Im}\,l \end{tikzcd} \quad .$$

This fact, together with the construction of t_0 in Theorem 4.6, gives the following characterization: $z\in Z$ belongs to $(\mathcal{P}\iota\circ\overline{\mathcal{P}}\iota\circ t_0)(x)$ if and only if there exist $a\in A$ and $u\in U$ such that

[8] In [Jac04b] the second equality is presented only as an inclusion $\subseteq$. In fact Lemma 3.1 used therein is strengthened to obtain an equality, not an inclusion; use the fact that $T=S\circ R$ in the lemma.

- $u_0 = x$, and
- $(\coprod_{n<\omega} F^n j)(u) = \iota(a) = z$.

Now to prove the theorem take an arbitrary trace t of g, an element x of X and $z \in (\mathcal{P}\iota \circ \overline{\mathcal{P}}\iota \circ t_0)(x)$. Let $a \in A$ and $u \in U$ be such as in the above characterization of z. We use Lemma 5.7 to establish that $z \in t(x)$.

Recall the construction of the initial F-algebra as a colimit, as presented after Theorem 4.6. It is standard (see e.g. [AK95]) that the structure map α and its inverse α^{-1} are characterized as the unique arrows which make the following diagram commute for each $m < \omega$.

$$\begin{tikzcd}
 & & A \arrow[dd, bend left, "\alpha^{-1}"] \\
F^m0 \arrow[r, "F^m h"] \arrow[urr, bend left, "\sigma_m"] \arrow[drr, bend right, "F\sigma_{m-1}"'] & F^{m+1}0 \arrow[ur, "\sigma_{m+1}"] \arrow[dr, "F\sigma_m"] & \\
 & & FA \arrow[uu, bend left, "\alpha", "\cong"']
\end{tikzcd}$$

Since the carrier A is a quotient of the coproduct $\coprod_{n<\omega} F^n 0$ we can choose a number n such that some element a_n of $F^n 0$ yields $a \in A$ as its equivalence class, i.e. $a = \sigma_n(a_n)$. Pick the first $n+1$ components of u as the sequence $u_0, u_1, \ldots, u_n$ in Lemma 5.7. By the definition of $u \in U$ the sequence satisfies Conditions 1 and 2 of the lemma. Condition 3 is equivalent to

$$(F^{n-1}\zeta \circ F^{n-2}\zeta \circ \cdots \circ \zeta \circ \iota)(a) \in t^n(u_n) \ ,$$

which is transformed by the repeated use of the definition $\zeta \circ \iota = F\iota \circ \alpha^{-1}$ of ι to

$$(F^n\iota \circ F^{n-1}\alpha^{-1} \circ F^{n-2}\alpha^{-1} \circ \cdots \circ F\alpha^{-1} \circ \alpha^{-1})(a) \in t^n(u_n) \ .$$

This is successively transformed to equivalent statements as follows:

$$\begin{aligned}
&(F^n\iota \circ F^{n-1}\alpha^{-1} \circ \cdots \circ \alpha^{-1})(a) \in (\lambda^n_Z \circ F^n t)(u_n) \ , && \text{(Definition of } t^n\text{)} \\
&\langle (F^n\iota \circ F^{n-1}\alpha^{-1} \circ \cdots \circ \alpha^{-1})(a), (F^n t)(u_n) \rangle \in (1 \times \lambda^n_Z)^{-1}(\in_{F^n Z}) \ , && \\
&\langle (F^n\iota \circ F^{n-1}\alpha^{-1} \circ \cdots \circ \alpha^{-1})(a), (F^n t)(u_n) \rangle \in \mathrm{Rel}^n_F(\in_Z) \ . && \text{(Lemma 5.6)}
\end{aligned}$$

Since we have $a = \sigma_n(a_n)$ and $\alpha^{-1} \circ \sigma_{m+1} = F\sigma_m$ for each m, this is equivalent to

$$\langle (F^n\iota \circ F^n\sigma_0)(a_n), (F^n t)(u_n) \rangle \in \mathrm{Rel}^n_F(\in_Z) \ . \tag{$*$}$$

Our aim is to show $(*)$. First, applying the n-th projection to the equality

$$z = (\prod_{n<\omega} F^n j)(u) = (\iota \circ \sigma_n)(a_n)$$

which defines u and a_n, we have

$$\begin{aligned}
(F^n j)(u_n) &= (\tau_n \circ \iota \circ \sigma_n)(a_n) \\
&= \big(F^n(j \circ i)\big)(a_n) \ .
\end{aligned} \tag{3}$$

The second equality holds by the characterization of ι presented after Theorem 4.6. Then we proceed as follows.

$$
\begin{aligned}
& \langle a_n, a_n \rangle \in (=_{F^n 0}) \\
\Longrightarrow\quad & \langle\, (F^n\sigma_0)(a_n),\ \big(F^n(j \circ i)\big)(a_n)\,\rangle \in \textstyle\coprod_{F^n\sigma_0 \times F^n(j\circ i)} (=_{F^n 0}) \\
\Longleftrightarrow\quad & \langle\, (F^n\sigma_0)(a_n),\ \big(F^n(j \circ i)\big)(a_n)\,\rangle \in \mathrm{Rel}_F^n(\emptyset_{A\times 1}) && \text{(Lemma 5.4)} \\
\Longleftrightarrow\quad & \langle\, (F^n\sigma_0)(a_n),\ (F^n j)(u_n)\,\rangle \in \mathrm{Rel}_F^n(\emptyset_{A\times 1}) && \text{(Using (3))} \\
\Longleftrightarrow\quad & \langle\, (F^n\sigma_0)(a_n),\ u_n\,\rangle \in \big(1 \times F^n j\big)^{-1}\,\mathrm{Rel}_F^n(\emptyset_{A\times 1}) = \mathrm{Rel}_F^n(\emptyset_{A\times X}) \\
& && \text{(Lemma 4.1)} \\
\Longrightarrow\quad & \langle\, (F^n\iota \circ F^n\sigma_0)(a_n),\ (F^n t)(u_n)\,\rangle \in \textstyle\coprod_{F^n\iota \times F^n t} \mathrm{Rel}_F^n(\emptyset_{A\times X}) = \mathrm{Rel}_F^n(\emptyset_{Z\times \mathcal{P}Z})
\end{aligned}
$$

Our goal $(*)$ follows from $\emptyset_{Z\times\mathcal{P}Z} \subseteq\, \in_Z$ and Lemma 4.1. This concludes the proof. ∎

Now that we know both the greatest and the least (ordinary) trace, we obtain the uniqueness of a finite trace as a corollary. This is our main technical result.

Corollary 5.9. *For a coalgebra $g : X \to \mathcal{P}FX$, its finite trace $f : X \to \mathcal{P}A$ uniquely exists. Equivalently, the initial F-algebra $\alpha : FA \xrightarrow{\cong} A$ in* **Sets** *gives rise to a final $F_{\mathcal{P}}$-coalgebra $\{-\}_{FA} \circ \alpha^{-1}$ in* $\mathbf{Sets}_{\mathcal{P}}$.

Proof. We show that an arbitrary finite trace f of g is equal to $\overline{\mathcal{P}}\iota \circ t_0$. Note that both $\mathcal{P}\iota$ and $\overline{\mathcal{P}}\iota$ are monotonic, and from Lemma 5.2.2 we have $\overline{\mathcal{P}}\iota \circ \mathcal{P}\iota = 1_{\mathcal{P}A}$. Lemma 5.3 yields that $\mathcal{P}\iota \circ f$ is a trace. Hence by the theorem we have

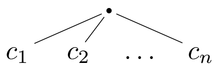
$$
\begin{aligned}
& \mathcal{P}\iota \circ \overline{\mathcal{P}}\iota \circ t_0 \subseteq \mathcal{P}\iota \circ f \\
\Longrightarrow\quad & \overline{\mathcal{P}}\iota \circ \mathcal{P}\iota \circ \overline{\mathcal{P}}\iota \circ t_0 \subseteq \overline{\mathcal{P}}\iota \circ \mathcal{P}\iota \circ f \\
\Longleftrightarrow\quad & \overline{\mathcal{P}}\iota \circ t_0 \subseteq f \ .
\end{aligned}
$$

For the opposite inclusion we use that t_0 is the largest trace:

$$
\begin{aligned}
& \mathcal{P}\iota \circ f \subseteq t_0 \\
\Longrightarrow\quad & f = \overline{\mathcal{P}}\iota \circ \mathcal{P}\iota \circ f \subseteq \overline{\mathcal{P}}\iota \circ t_0 \ .
\end{aligned}
$$

By the antisymmetry of $\subseteq$ we obtain the claim. ∎

Example 5.10. Now we go back to context-free grammars. For the functor $F = (\Sigma + -)^*$ the definition of a finite trace is interpreted as follows. The map $f : X \to \mathcal{P}\Sigma^{\triangle}$ is a finite trace of a context-free grammar $g : X \to \mathcal{P}\big((\Sigma + X)^*\big)$ if and only if for each element



of $f(x)$ (here $c_1, c_2, \ldots, c_n \in \Sigma + \Sigma^{\triangle}$), there exists a string $d_1 d_2 \ldots d_n \in g(x)$ such that for each $i = 1, 2, \ldots, n$:

- 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 $c_i \in f(d_i)$.

In contrast to the case in Example 4.7, Corollary 5.9 says that the above corecursive statement determines $f(x)$ uniquely, as the set of all the *finite* SPTs generated by g from x. Hence we have obtained the set of generated finite SPTs by $F_{\mathcal{P}}$*-coinduction*, in the Kleisli category $\mathbf{Sets}_{\mathcal{P}}$ of sets and relations.

Theorem 5.8 says that the finite trace, when embedded, yields the smallest trace. For the grammar in Example 2.1 this smallest trace $\mathcal{P}\iota \circ f$ is strictly smaller than the largest one t_0: the infinite SPT on the right in Example 2.6 is in $t_0(\mathbf{F})$ but not in $(\mathcal{P}\iota \circ f)(\mathbf{F})$.

Example 5.11. In relation to [Jac04b, Section 2], we consider here the functor $F = \Sigma \times -$, which makes a $\mathcal{P}F$-coalgebra a *labelled transition system (LTS)*. Traditionally the word *trace* is used for what is called the *largest trace* in this paper. Hence Theorem 4.6 can be read as: *the notion of trace (in a traditional sense) cannot be captured only by our corecursive equation in Definition 4.4.*

Since the carrier of the initial $(\Sigma \times -)$-algebra is empty, the finite trace of any LTS maps each state to the emptyset, as is implied by the "never-terminating" nature of LTSs. The smallest trace, induced by this finite trace via embedding, gives a more trivial counterexample of the uniqueness of traces (in our sense) than the one presented in [Jac04b].

6 Conclusions and future work

We have presented an abstract framework for extracting only the finite behavior of a (non-deterministic) coalgebra, which is uniquely determined via corecursive characterization, and then applied it to context-free grammars/languages. The (co)algebraic structures on 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.

References

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

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

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

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

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

[BW83] Michael Barr and Charles Wells. *Toposes, Triples and Theories*, volume 278 of *Grundlehren der math. Wissenschaften*. Springer-Verlag, 1983. Available free for downloading at `http://www.cwru.edu/artsci/math/wells/pub/ttt.html`.

[Cho56] Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124, 1956.

[Jac04a] Bart 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] Bart 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.

[Jac05] Bart 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.

[JR97] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of EATCS*, 62:222–259, 1997.

[LP81] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1981.

[Rut00] Jan 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.