Infinite Trace Semantics for Coalgebras

Christian Kissig[1]

University of Leicester
Department of Computer Science
University Road
Leicester LE17RH
United Kingdom

Abstract

Infinite trace semantics is a fundamental concept in the study of non-deterministic, and possibly non-terminating transition systems: most prominently, automata operating on infinite input.
This is a first exposition of our definition of infinite trace semantics for coalgebras which are structured in a transition type, given by a *Set*-functor and a branching type, given by a *Set*-monad. Branching subsumes possibilistic (non-deterministic), probabilistic, and graded branching.
Being a coinductive notion, infinite trace semantics is defined globablly over the transition structure, but has only a finite immediate horizon. Thus infinite trace semantics is not unique. We give an inductive construction of infinite trace semantics, yielding the largest one.
In application to automata, we show the acceptance behaviour of non-deterministic coalgebra automata an infinite trace semantics. Jacob's infinite trace semantics for non-deterministic coalgebra arises as a special case of non-deterministic coalgebra automata.

Keywords: Coalgebra, Trace Semantics, Automata Theory

---

1 Introduction

Semantics of Coalgebras over Set

Coalgebras in the category *Set* of sets and functions generalise state-based, and possibly non-terminating transition systems. Thereby functors on *Set* take the roll of transition types. Examples for transition types are $T(-) = (-) \times (-)$ (binary trees), $T(-) = \mathcal{P}(\{\surd\} + A \times (-))$ (non-deterministic A-labelled transition systems (A-LTSs) with successful termination, $\surd$), $T(-) = \mathcal{P}(A \times (-))$ (non-deterministic automata recognising infinite A-words).

Classically, the semantics of a coalgebra is given as the unique map into the final coalgebra for the transition type. Intuitively, the final coalgebra embodies all possible behaviours of coalgebras of the same transition type (T-behaviours). The final coalgebra for our A-LTS transition type, contains all finitely branching rooted

---

[1] Email:christian@mcs.le.ac.uk

A-LTSs, and decomposes such a LTS into its root and pairs of labels and immediate successors. The kernel of the final coalgebra map characterises T-bisimilarity.

Finite Trace Semantics of Coalgebras

In the examples of non-deterministic A-LTSs and non-deterministic word automata, it makes sense to distinguish the roles of non-determinism ($\mathcal{P}$) and transition ($\{\surd\} + A \times (-)$). In fact non-determinism may be seen more generally as branching, subsuming probabilistic ($\mathcal{D}_{\leq}$), possibilistic ($\mathcal{P}$), and graded branching $\left(\mathbb{N}^{(-)}\right)_{\omega}$). Monads are suitable branching types, because their units and multiplication respectively accomodate a deterministic subbehaviour and contraction of non-deterministic choices. We call coalgebras for a branching type B and a transition type T, (B, T)-coalgebras.

(BT)-behaviour of a (B, T)-coalgebra describes the future behaviour including the branching at every point in the future. However, a semantics that accounts for branching should take branching upfront, assigning to each point either a set ($\mathcal{P}$), a multiset ($\left(\mathbb{N}^{(-)}\right)_{\omega}$), or a distribution ($\mathcal{D}_{\leq}$) of T-behaviours. Previous work has shown that moreover the codomain of trace semantics should have the structure of an algebra for the branching type.

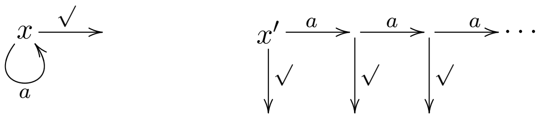
In their work on generic trace theory, Jacobs et al [Jac04,HJS06] defined finite trace semantics for (B, T)-coalgebras. Their work is built on the following ideas.

- Distributive laws allow to define the lifting $\overline{T}$ of a *Set*-functor T in the Kleisli-category $Kl(B)$ of B.
- (B, T)-coalgebras in *Set* are $\overline{T}$-coalgebras in $Kl(B)$.
- Finite trace semantics is the unique morphism into the final $\overline{T}$-coalgebra if the latter is the ω-limit of the step-wise retract of the initial $\overline{T}$-sequence.
- The ω-limit exists if it is isomorphic to the ω-colimit of the initial $\overline{T}$-sequence. In fact the ω-colimit of the initial T-sequence exists for many functors as the union in *Set* and is preserved by the free (left adjoint) functor $Set \to Kl(B)$.
- ω-limit and -colimit coincide if $Kl(B)$ can be enriched with a directed complete partial order with bottom element, and $\overline{T}$ commutes with order. This is a result of Smyth and Plotkin [SP82].

In [KK10] Kurz and the author point out that the construction of finite trace semantics excludes finitary branching, such as $\mathcal{P}_{\omega}$ and $\left(\mathbb{N}^{(-)}\right)_{\omega}$. Instead of striving for the limit construction of finite trace semantics, one should consider cones over the initial $\overline{T}$-sequence. Under assumptions on the size of B and T, one obtains the codomain of finite trace semantics as a non-free B-algebra.

Infinite Trace Semantics and Automata

At the beginning we mentioned that coalgebras generalise possibly non-terminating transition systems. However, finite trace semantics can distinguish (B, T)-coalgebras only up to a finite depth. Infinite trace semantics extends the horizon to a countable depth. The points x and x' in the example below are finitely, but not infinitely trace equivalent.



(1)

Although finite and infinite trace semantics are both coinductive notions, the first is unique and can be constructed inductively (due to the finiteness of depth), whereas the latter is purely coinductive.

If both, branching and transition type, distribute over countable products, the codomain of infinite trace semantics is an algebra for the branching type B with generators of infinite T-behaviours. The B-algebra is generally not free, for reasons similar to finite traces semantics. Infinite trace semantics is in general not unique, as Jacobs has pointed out for $(\mathcal{P}, T)$-coalgebras in [Jac04].

Infinite trace semantics has most prominently appeared as the language theory of automata. Venema [Ven04] introduced coalgebra automata generalising word- and tree-automata. The following ideas underlie the construction of coalgebra automata.

- In the same way as T-coalgebras generalise words, trees, etc., do non-deterministic T-coalgebra automata generalise automata recognising words, trees, etc..
- The state space of a coalgebra automaton is a set Q. The transition structure of a non-deterministic T-coalgebra automaton is a $(\mathcal{P}, T)$-coalgebra.
- The acceptance behaviour is given in terms of graph games played by a verifier (∃, Eloise) and a falsifier (∀, Abelard).
- Acceptance of finite input is given by the transition structures of the automaton and the input structure, where acceptance of infinite input depends on an acceptance condition on the infinite matches of the acceptance game.

In Section 5 we shall see that the acceptance behaviour of an automaton can be suitably described as an infinite trace semantics. That the acceptance behaviour of a coalgebra automaton on infinite input is parameterised in an acceptance conditions witnesses the non-uniqueness of infinite trace semantics.

The Structure of This Paper In Section 3 we sketch, what we expect of infinite trace semantics on the example of graded A-LTSs. Analogous to finite trace semantics we develop a generic definition of infinite trace semantics in Section 4.3. In Section 4.4 we refine the codomain of infinite trace semantics to certain converging sequences. We show that generic infinite trace semantics is an infinite trace semantics in Theorem 4.6. Finally, we show that the acceptance behaviour of coalgebra automata is infinite trace semantics in Theorem 5.3. Jacob's infinite trace semantics for non-deterministic coalgebras [Jac04] arises as a special case of the trace semantics of coalgebra automata.

Acknowledgements

2 Preliminaries

Functors,Monads,Coalgebras,Algebras

Definition 2.1 [Coalgebras for Functors] Coalgebras for a functor T on a category $\mathcal{C}$ consist of a carrying object X of $\mathcal{C}$ and a transition map $\gamma : X \to TX$. A coalgebra morphism $\langle X, \gamma\rangle \to \langle Y, \delta\rangle$ is a morphism $f : X \to Y$ in $\mathcal{C}$ such that $Tf \circ \gamma = \delta \circ f$ commutes. The definition of coalgebras restricts to $\mathcal{C} = \mathit{Set}$. Henceforth we refer to coalgebras $\langle X, \gamma\rangle$ by their transition map, γ.

Definition 2.2 [Algebras for *Set*-Functors] Dually, algebras for a *Set*-functor B consist of a set X (carrier) and an algebra map $\alpha : BX \to X$. B-algebra morphisms $\langle X, \alpha\rangle \to \langle Y, \beta\rangle$ are functions $f : X \to Y$ such that $\beta \circ Bf = f \circ \alpha$ commutes. In later sections, we will refer to a B-algebra $\langle X, \alpha\rangle$ by its algebra map α.

Definition 2.3 [*Set*-Monads] *Set*-monads B are *Set*-functors that come with two natural transformations, $\eta : \mathit{Id} \Rightarrow B$ (unit) and $\mu : B^2 \Rightarrow B$ (multiplication) commuting such that $\mu \circ B\eta = \mathit{id} = \mu \circ \eta_B$ and $\mu \circ B\mu = \mu \circ \mu_B$.

Example 2.4 (i) The powerset monad $\mathcal{P}$ with set-union and the singleton constructor $\{(-)\}$ is a monad, and similarily the finitary powerset functor $\mathcal{P}_\omega$.

(ii) The Bag (finite multiset) functor $\left(\mathbb{N}^{(-)}\right)_\omega$ takes a set X to the set $\left(\mathbb{N}^X\right)_\omega$ of its finite multisets, and functions $f : X \to Y$ to multiset-functions $\left(\mathbb{N}^f\right)_\omega : \left(\mathbb{N}^X\right)_\omega \to \left(\mathbb{N}^Y\right)_\omega$ taking multisets $m \in \left(\mathbb{N}^X\right)_\omega$ to $\lambda y. \sum_{x\in f^{-1}(y)} m(x)$. The bag functor with multiset-union and the multiset-singleton constructor constitutes a monad.

(iii) A (sub-)distribution of a set X is a function $d : X \to [0, 1]$ such that $\sum_{x\in X} d(x) = 1$ ($\sum_{x\in X} d(x) \leq 1$). The (sub-)distribution functor $\mathcal{D}_=$ ($\mathcal{D}_\leq$) takes a set X to the set of its (sub-)distributions, and functions $f : X \to Y$ to $\lambda m.\lambda y. \sum_{x\in f^{-1}(y)} m(x)$. For each X we can define functions

$$\mu_X(d' \in \mathcal{D}^2X)(x) := \textstyle\sum_{d\in\mathcal{D}X} d'(d) * d(x) \qquad \eta_X(x) := \lambda y. \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise} \end{cases}$$

μ and η are transformations natural in X and form with B a monad.

(iv) $\mathcal{P}_\omega$ and $\mathcal{B}$ are examples of functors which take a set X into the set $\left(\mathcal{S}^X\right)_\omega$ of evaluations of X into a semiring $\mathcal{S}$ with finite support, and functions $f : X \to Y$ into functions $\left(\mathcal{S}^X\right)_\omega \to \left(\mathcal{S}^Y\right)_\omega$ such that $m \in \left(\mathcal{S}^X\right)_\omega \mapsto \lambda y. \sum_{x\in f^{-1}(x)} m(x)$. For $\mathcal{P}_\omega$ the semiring is the boolean algebra $\langle\{\top, \bot\}, \wedge, \vee, \top, \bot\rangle$, and for $\mathcal{B}$ the semiring are the natural numbers $\langle\mathbb{N}, +, *, 0, 1\rangle$.

(v) Another example of a semiring monad uses the min-semiring $\langle\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0\rangle$ of natural numbers augmented with a top element, ∞, with an idempotent additive operation, min, and a commutative multiplicative operation, $+$, such that ∞ is neutral w.r.t. min and 0 w.r.t. $+$, and 0 absorbs w.r.t. min.

Definition 2.5 [Algebras for *Set*-Monads] Algebras $\langle X, \alpha\rangle$ for monads B commute with the monad structure, that is $\alpha \circ \mu_X = \alpha \circ B\alpha$ and $\alpha \circ \eta_X = id_X$. The component $\mu_X : B(BX) \to BX$ of μ is the free B-algebra generated by X.

Example 2.6 (i) Algebras for the powerset monad $\mathcal{P}_\omega$ ($\mathcal{P}$) are (complete) semilattices.

(ii) Algebras for the Bag monad $\left(\mathbb{N}^{(-)}\right)_\omega$ are (left) $\mathbb{N}$-semimodules, where $\mathbb{N}$ is the semiring of natural numbers with $+$ and $*$ as addition and multiplication.

(iii) Algebras for the distribution monads are distribution algebras with convex linear combination.

(iv) Algebras for the semiring monad of a given semiring $\mathcal{S}$ are the (left) $\mathcal{S}$-semimodules

The Kleisli-Construction, Kleisli-Lifting, Distributive Laws

Definition 2.7 [The Kleisli-Category] The free B-algebras with B-algebra morphisms form a category, the Kleisli-category of B, denoted $Kl(B)$. Axiomatically, $Kl(B)$ is given by

- objects X which are sets
- morphisms $f : X \to Y$ which are functions $X \to BY$ in *Set*
- composition $(g : Y \to Z) \circ (f : X \to Y)$ which is $\mu_Z \circ Bg \circ f$ in *Set*

There is a forgetful functor $U : Kl(B) \to Set$ taking sets X in $Kl(B)$ to BX in *Set* and morphisms $f : X \to Y$ in $Kl(B)$ to $\mu_Y \circ Bf$ in *Set*. U has a left adjoint, $F : Set \to Kl(B)$ taking sets X in *Set* to X in $Kl(B)$ and functions $f : X \to Y$ to $\eta_Y \circ f$ in $Kl(B)$. F and U compose such that $B = UF$.

The axiomatic definition of Kleisli-categories allows us to compute in *Set* the commutativity of diagrams in $Kl(B)$. In later sections we will use the above axioms as abstract syntax in *Set* without prior warning.

In this paper we particularly consider (B,T)-coalgebras $\gamma : X \to BTX$ structured in a *Set*-monad B and a *Set*-functor T. The transpose of such a coalgebra γ under $F \dashv U : Set \to Kl(B)$ is a morphism $\gamma^\dagger : X \to TX$. $\gamma^\dagger$ is not yet a coalgebra, as coalgebra morphisms are not well-defined. We first need to lift T to a functor $\overline{T}$ on $Kl(B)$, and then we obtain the Kleisli-lifting $\overline{\gamma}$ of γ. The Kleisli-lifting of T is a functor $\overline{T}$ making $\overline{T}F = FT$ commute. The Kleisli-lifting exists iff there is a distributive law with B.

Definition 2.8 [Distributive Laws] A distributive law for a functor T and a monad B is a natural transformation $\pi : TB \Rightarrow BT$ making $\pi \circ T\mu = \mu_T \circ B\pi \circ \pi_B$ and $\pi \circ T\eta = \eta_T$ commute.

Example 2.9 All of the monads in Example 2.4 have a distributive law with the functor $T(-) = 1 + A \times X$.

(i) $\pi : T\mathcal{P} \Rightarrow \mathcal{P}T$: $\pi_X(*) := \{*\}$, $\pi_X(a, Y \subseteq X) := \{(a,x) \mid x \in Y\}$.

(ii) $\pi : T\left(\mathbb{N}^{(-)}\right)_\omega \Rightarrow \left(\mathbb{N}^{T(-)}\right)_\omega$: $\pi_X(*) := \eta_{\{*\}+A\times X}(*)$, and $\pi_X(a,m)(a,x) := \{(a,x) \mapsto m(x), (b,x) \mapsto 0, * \mapsto 0 \mid a \in A, b \in A, b \neq a, x \in X\}$

(iii) $\pi : T\mathcal{D} \Rightarrow \mathcal{D}T$: $\pi_X(*) := \eta_{\{*\}+A\times X}(*)$, and $\pi_X(a,d) := \{(a,x) \mapsto d(x), (b,x) \mapsto 0, * \mapsto 0 \mid a \in A, b \in A, b \neq a, x \in X\}$ where $D \in \{\mathcal{D}_{\leq}, \mathcal{D}_{=}\}$

Given a distributive law $\pi : TB \Rightarrow BT$ we can define the Kleisli-lifting as a continuous extension of the Kleisli-lifting $\overline{T}$ on free B-algebras. The latter is defined on objects X as $\overline{T}X = TX$ and on morphisms $f : X \to Y$ $\overline{T}f = \pi_Y \circ Tf$.

Sequence Diagrams

We denote sequence diagrams in categories $\mathcal{C}$ as Seq and mean a category $\mathcal{I}$, which is the skeleton of the sequence (natural numbers and $\leq$ as morphisms), together with a functor $Seq : \mathcal{I} \to \mathcal{C}$. Particular sequences are the initial and final sequence of a functor T, that we denote as $Seq_T : I \to Set$ taking $(-) \mapsto T^{(-)}\emptyset$, and $Seq^T : \mathcal{I}^{op} \to Set$ taking $(-) \mapsto T^{(-)}\{*\}$ respectively. Every diagram indexed in a category $\mathcal{I}$ can be made discrete by removing all morphisms from $\mathcal{I}$ but the identity morphisms. This way we obtain a discrete category $|\mathcal{I}|$ being a subcategory of $\mathcal{I}$. We denote the restriction of Seq_T and Seq^T to $|\mathcal{I}|$ as $|Seq_T|$ and $|Seq^T|$, respectively.

Other Notation

We denote

- the collection of finite words over an alphabet A as A^*, and the collection of infinite words as A^ω.
- the initial object of a category as 0, the final object as 1, maps from 0 into some object X as $¡_X$, and maps from 1 into some object X as $!_X$.

3 An Example of Infinite Trace Semantics

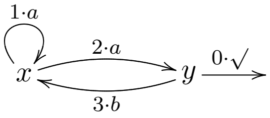
The calculations in this section may be counted to the folklore of trace theory, but they exemplify the ideas underlying our definition of infinite trace semantics of Section 4. All remarks are to be read as informal. We will detail the definitions in greater generality in later sections.

Graded transition systems with labels from an alphabet A, short: graded A-LTS, have the Bag monad of Example 2.4 as the branching type, $B = \left(\mathbb{N}^{(-)}\right)_\omega$, and the labelling functor with successful termination as the transition type, $T(-) = \{\surd\} + A \times (-)$.

Infinite traces through graded A-LTS are infinite words from A^ω or finite $\surd$-terminated words from $A^*\surd$. We must include finite words because the transition type allows explicitely for successful termination. One may think of finite $\surd$-terminated traces as continued by stuttering at $\surd$, e.g. $a\surd\surd\surd\cdots$.

Each infinite trace has a transition structure of type T taking $aw \mapsto (a, w)$ for any $a \in A$ and $x \in X$, and $\surd \to \surd$ (stuttering). The collection Z of all infinite traces with the combined behaviour is the final transition structure ξ of type T, so that there is for each T-transition structure a unique structure preserving map into ξ. In other terms, ξ is the final T-coalgebra.

Each state x in a graded A-LTS may have many traces. Consider the example



(2)

We compute the infinite traces of x for successively increasing depth as follows.

(i) $tr^{\infty}_{0}(x) = *$ and $tr^{\infty}_{0}(y) = *$
  - By looking at traces of depth 0 the infinite trace semantics of x and y remains obscure. We denote the unkown future as $*$.

(ii) $tr^{\infty}_{1}(x) = 1 \cdot a(tr^{\infty}_{0}(x)) + 2 \cdot a(tr^{\infty}_{0}(y)) = 1 \cdot a(*) + 2 \cdot a(*) = 3 \cdot a(*)$ and $tr^{\infty}_{1}(y) = 0 \cdot \surd + 3 \cdot b(tr^{\infty}_{0}(x)) = 0 \cdot \surd + 3 \cdot b(*)$.
  - The infinite trace semantics of depth 1 depends on the adjacent edges and the depth-0 trace semantics of neighbouring states.
  - The set of descending depth-1 traces has the structure of a B-algebra.

(iii) $tr^{\infty}_{2}(x) = 1 \cdot a(tr^{\infty}_{1}(x)) + 2 \cdot a(tr^{\infty}_{1}(y) = 1 \cdot a(3 \cdot a(*)) + 2 \cdot a(0 \cdot \surd + 3 \cdot b(*)) = (1+3) \cdot aa(*) + (2+0)a\surd + (2+3) \cdot ab(*)$. $tr^{\infty}_{2}(y)$ can be computed similarly.
  - We see that the weights can be gathered upfront (using the multiplication of the branching type B), and
  - branching can be distributed over transition (by means of a distributive law).

and so forth.

For the calculations in Section 4, projective morphisms will become important. These morphisms make forget the last transition step in traces of depth $n+1$, formally these projective morphisms are free B-homomorphic extensions of functions

$$T^{n}!_{T1} : T^{n}T1 \to T^{n}1 \tag{3}$$

taking $T^{2}!_{T1} : abc* \mapsto ab*$, for instance. The infinite trace semantics tr^{∞} above commutes not strictly with these projective maps, in the sense that $BT^{n}!_{T1} \circ tr^{\infty}_{n+1} \neq tr^{\infty}_{n}$. Branching spoils the commutativity as this example shows:

$$\begin{array}{l}
tr^{\infty}_{2}(x) = 1 \cdot a(tr^{\infty}_{1}(x)) + 2 \cdot (tr^{\infty}_{1}(y)) = 3 \cdot aa(*) + 2 \cdot a(tr^{\infty}_{1}(y)) = \\
BT!_{T1}(tr^{\infty}_{2}(x)) = 3 \cdot a(*) + BT!_{T1}(2 \cdot a(tr^{\infty}_{1}(y))) \\
tr^{\infty}_{1}(x) = 1 \cdot a(*) + 2 \cdot a(*)
\end{array}$$

It appears that $tr^{\infty}_{n} \leq BT^{n}!_{T1} \circ tr^{\infty}_{n+1}$, where $\leq$ stems from the order on the multiplicities, that are natural numbers in this example. However, the order carriers a more sophisticated structure as we shall see in Section 4.4.

4 Infinite Trace Semantics

4.1 A Regression into Final Coalgebra Semantics

The final T-sequence Seq^T is a diagram in Set generated from the final object $\{*\}$ and the final object map $!_{T\{*\}} : T\{*\} \to \{*\}$ by iterated application of T.

$$\begin{tikzcd} \{*\} & T\{*\} \arrow[l, "!_{T\{*\}}"'] & T^2\{*\} \arrow[l, "T!_{T\{*\}}"'] & \cdots \arrow[l] \end{tikzcd} \tag{4}$$

We say, Seq^T terminates in κ steps, if the limit of Seq^T is the limit of the restriction of Seq^T to the first κ objects. Seq^T can not be expected to terminate, for instance if $T = \mathcal{P}$. Termination of the latter would violate the continuum hypothesis. For finitary Set-functors T, Worrel [Wor05] showed that the final sequence terminates in $\omega * 2$ steps. Henceforth we shall make the stricter assumption that T is such that Seq^T terminates in ω steps. We denote the limit of Seq^T as Z.

If Seq^T terminates, Lambek's Lemma tells us that $Z \cong TZ$. The latter isomorphism, we call it ξ, is a T-coalgebra structure on Z which is final among all T-coalgebras. Moreover the final coalgebra morphism from a coalgebra $\gamma : X \to TX$ can be computed as the limit of the cone $(g_n : X \to T^n 1)_{n<\omega}$, defined inductively as

$$g_0 :=!_X, \; g_{n+1} := Tg_n \circ \gamma \tag{5}$$

Later on we will use that ξ has such a cone itself, we denote it as $(f_n)_{n<\omega}$.

Remark 4.1 Shuffling the definition in (5), we obtain an alternative definition of semantics for T-behaviour. Therefor we need the n-fold composition of γ

$$\gamma^0 = id_X, \; \gamma^{n+1} := T^n\gamma \circ \gamma^n \tag{6}$$

Then we obtain the semantics as the composition

$$g_n = T^n!_X \circ \gamma^n \tag{7}$$

That (5) and (7) coincide follows through an inductive argument by expanding the definitions of g_n and γ^n.

The conceptual difference between (5) and (7) is that the first pre-composes γ, whereas the latter post-composes γ. The latter is thus closer related to the projections $T^n!_{T1}$. Although trivial, this idea will play a central role in our arguments below.

4.2 A Review of Finite Trace Semantics

The finite trace semantics of a (B, T)-coalgebra $\langle X, \gamma\rangle$ is obtained by iteration along the discrete initial sequence $|\mathit{Seq}_{\overline{T}}|$ for the functor $\overline{T}$ in $\mathit{Kl}(B)$. A fundamental assumption of the construction asserts an arrow $\perp_X : X \to \emptyset$ into the initial object[2] $\emptyset$ of $\mathit{Kl}(B)$.

Example 4.2 (i) For the powerset monad, $B = \mathcal{P}$, $\perp_X$ takes any $x \in X$ to $\emptyset$.

---

[2] $\emptyset$ is the initial object of Set and $F : \mathit{Set} \to \mathit{Kl}(B)$ preserves initial objects, being a left adjoint.

(ii) For the Bag monad, $B = \left(\mathbb{N}^{(-)}\right)_\omega$, $\perp_X$ takes any $x \in X$ to the empty multiset.

(iii) For the sub-distribution monad $\mathcal{D}_{\leq}$, $\perp_X$ takes any $x \in X$ to the empty distribution. There is map $\perp_X$ for the distribution monad $\mathcal{D}_{=}$ as $\mathcal{D}_{=}\emptyset$ is empty.

(iv) For any semiring monad, $\perp_X$ takes any $x \in X$ to the constantly 0 valuation.

The finite trace semantics of γ is then a cone $(tr_n^{fin})_{n<\omega}$ over $|Seq_{\overline{T}}|$ defined inductively such that $tr_0^{fin} := \perp_X$, and $tr_{n+1}^{fin} := \overline{T}\,tr_n^{fin} \circ \gamma$. For our intuition, we read tr_n^{fin} as a morphism that takes a point $x \in X$ to the (successfully) terminating traces of depth less than n.

$$\begin{tikzcd}
& X \arrow[r, "\overline{\gamma}"] \arrow[dl, bend right, "tr_0^{fin}"'] \arrow[dr, "tr_n^{fin}"'] \arrow[drr, "tr_{n+1}^{fin}"] & \overline{T}X \arrow[dr, "\overline{T}tr_n^{fin}"] & & \\
\emptyset & \cdots & \overline{T}^n\emptyset & \overline{T}^{n+1}\emptyset & \cdots
\end{tikzcd} \tag{8}$$

Jacobs et al [Jac04,HJS06] make further assumptions. Then $\perp_{\overline{T}\emptyset} : \overline{T}\emptyset \to \emptyset$ is a retract of the lifted empty morphism $F\emptyset : \emptyset \to \overline{T}\emptyset$, and so are all liftings under $\overline{T}^n$, $\overline{T}^n \perp_{\overline{T}\emptyset} \circ \overline{T}^n F\emptyset = id_{\overline{T}^n\emptyset}$, for any $n < \omega$. The sequence of retracts can be thought of as an embedded final sequence. There we have to bear in mind, that the final object, seeding the final sequence, may not be a free B-algebra.

4.3 Generic Infinite Trace Semantics

First, we give a generic definition of infinite trace semantics in analogy to the construction of finite trace semantics as a cone.

The cone $(tr_n^\infty : X \to \overline{T}^n 1)$ is defined inductively such that $tr_0^\infty := F!_X$ in the base case, and $tr_{n+1}^\infty := \widetilde{T}\,tr_n^\infty \circ \widetilde{\gamma}$ for all $n < \omega$.

$$\begin{tikzcd}
& X \arrow[r, "\overline{\gamma}"] \arrow[dl, bend right, "tr_0^{\infty}"'] \arrow[dr, "tr_n^{\infty}"'] \arrow[drr, "tr_{n+1}^{\infty}"] & \overline{T}X \arrow[dr, "\overline{T}tr_n^{\infty}"] & & \\
\{*\} & \cdots & \overline{T}^n\{*\} & \overline{T}^{n+1}\{*\} & \cdots
\end{tikzcd} \tag{9}$$

The definition above depends on the coalgebra structure $\overline{\gamma}$. Next, we give a characterisation of the codomain of infinite trace semantics, depending only on the branching and on the transition type.

4.4 Infinite Traces form Lax-Commuting Diagrams

The cone $(tr_n^\infty)_{n<\omega}$ does not commute with the morphisms $(\overline{T}^n F!_{T1})_{n<\omega}$. Intuitively, this is because $\overline{\gamma}$ could be strictly non-deterministic, so that $tr_1^\infty(x) := Fp \circ \overline{T}tr_0^\infty \circ \overline{\gamma}(x)$ could be above $tr_0^\infty(x)$ with respect to a certain order $\leq$, as we have seen in Section 3.

In fact the order are spans, which tells us whether the next step in a trace comes from the next step in some (B,T)-coalgebra. The definition of the spans involves the distributive law, $\pi : TB \Rightarrow BT$. We may describe an iterative application of the distributive law as a law $\pi^n : T^n B \Rightarrow BT^n$, which we define inductively

$$\pi^0 := \pi,\ \pi^{n+1} := \pi^n_T \circ T^n \pi \tag{10}$$

For depth $n < \omega$, we obtain the following span $\leq_n$.

$$\begin{tikzcd} BT^n1 & BT^nBT1 \arrow[l, "BT^n!_{BT1}"'] \arrow[r, "B\pi^n_{T1}"] & BBTT^n1 \arrow[r, "\mu_{TT^n1}"] & BTT^n1 \arrow[r, "BT^n!_{T1}"] & BT^n1 \end{tikzcd} \tag{11}$$

Example 4.3 For $B = \left(\mathbb{N}^{(-)}\right)_\omega$ and $T = \{\surd\} + A \times (-)$, we may have the following element of a span $\leq_n$ for $n = 2$.

$$\begin{tikzcd} 4ab* & 4ab2c* \arrow[l, mapsto, "BT^n!_{BT1}"'] \arrow[r, mapsto, "B\pi^n_{T1}"] & 42abc* \arrow[r, mapsto, "\mu_{TT^n1}"] & 8abc* \arrow[r, mapsto, "BT^n!_{T1}"] & 8ab* \end{tikzcd} \tag{12}$$

Proposition 4.4 (i) $\leq_n$ *is reflexive if* B *and* T *preserve epis.*

(ii) $\leq_n$ *is preserved by* $\overline{T}$.

(iii) $\leq_n$ *is preserved by* B.

Proof.

(i) To show $\leq_n$ *reflexive*, take the set $BT^n\eta_{T1}[BT^nBT1] \subseteq BT^nBT1$ of lifted generators of $BT1$. Then observe that $\mu_{TT^n1} \circ B\pi^n_{T1} \circ BT^n\eta_{T1} = id_{BTT^n1}$ and $BT^n!_{BT1} \circ BT^n\eta_{T1} = BT^n!_{T1}$. Thus both legs of the span restricted to these generators are the same. It remains to show that both legs are surjective. As we are in *Set*, surjective is equivalent to epic. As $!_{T1}$ is epic, and both B and T preserve epis, we obtain reflexivity on BT^n1.

(ii) Using naturality of π, $\pi \circ T\mu = \mu \circ B\pi \circ \pi$, and the definition of $\leq_n$ and $\overline{T}$, we obtain $\overline{T}(\leq_n) = \leq_{n+1}$.

(iii) That $\leq_n$ is preserved by B follows, because we post-compose the lifting $B \leq_n$ with μ and pre-compose with η_B. We obtain $\leq_n$ from $B \leq_n$ by using the naturality of μ, and the law $\mu \circ \eta_B = id$.

□

We call a sequence $w \in \prod_{n<\omega} BT^n1$ increasing, if for all $n < \omega$,

$$w(n) \leq_n BT^n!_{T1}(w(n+1)) \tag{13}$$

Definition 4.5 [Infinite Trace Semantics] An infinite trace semantics for a coalgebra $\langle X, \gamma \rangle$ is a set $(tr_n : X \to BT^n1)_{n<\omega}$ of functions such that

(i) $tr_{n+1} = \mu_{TT^n} \circ B\pi_{T^n1} \circ BT\,tr_n$

(ii) and $(tr_n)_{n<\omega}(x)$ is an increasing sequence for all $x \in X$.

The first condition means that infinite traces are stable under prepending a transition step, the second excludes implausible traces. The following theorem supports that our definition of infinite trace semantics is suitable.

Theorem 4.6 *Generic infinite trace semantics,* tr^∞, *is an infinite trace semantics.*

To prove the theorem, we have to take a conceptual detour. A coalgebra prepends branching and transition to future behaviour. The order $\leq_n$, however, is defined through forgetting about the farthest part of future behaviour.

Unfolding the inductive definition of tr^∞ tells us, that the infinite trace behaviour of a (B,T)-coalgebra γ up to depth n can be computed by first applying γ n times, and then bringing the branching in front (π), and contracting the branching (μ).

The iterated composition of π and μ yields a natural transformation $\theta^n : (BT)^n \Rightarrow BT^n$

$$\theta^0 := \eta, \ \theta^{n+1} := \mu_{T^nT} \circ B\pi^n_T \circ \theta^n_{BT}. \tag{14}$$

The above allows us to redefine tr^∞ as follows.

Lemma 4.7 *For all $n < \omega$, $tr^\infty_n = BT^n!_X \circ \theta^n_X \circ \gamma^n$.*

where γ^n is the n-fold composition of γ as in Remark 4.1.

Proof. is a straightforward argument expanding the definitions of tr^∞_n, θ^n, and γ^n repeatedly. ☐

Proof. (Theorem 4.6) The proof of the theorem is summarized in the following diagram.

$$\begin{tikzcd}
X \arrow[d, "\gamma^n"] \arrow[dr, bend left, "\gamma^{n+1}"] & & & & \\
(BT)^n X \arrow[r, "(BT)^n\gamma"] \arrow[d, "\theta^n_X"] & (BT)^n(BT)X \arrow[d, "\theta^n_{BTX}"] \arrow[drr, bend left, "\theta^{n+1}"] & & & \\
BT^n X \arrow[r, "(BT^n)\gamma"] \arrow[d, "(BT^n)!_X"] & BT^nBTX \arrow[r, "B\pi^n_{TX}"] \arrow[d, "(BT^n)(BT)!_X"] & BBT^nTX \arrow[r, "\mu_{T^nTX}"] \arrow[d, "BBT^nT!_X"] & BT^nTX \arrow[d, "BT^nT!_X"] & \\
BT^n 1 & BT^nBT1 \arrow[l, "BT^n!_{BT1}"'] \arrow[r, "B\pi^n_{T1}"] & BBT^nT1 \arrow[r, "\mu_{T^nT1}"] & BT^nT1 \arrow[r, "(BT^n)!_{T1}"] & BT^n 1
\end{tikzcd} \tag{15}$$

Both triangles commute because of the definition of γ^n and θ^n, respectively. All squares (with the exception of the bottom left one) commute due to naturality of θ^n, π^n, and μ.

The bottom row is Condition (11) and thus specifies $\leq_n$. The left and right outer leg are respectively tr^∞_n and tr^∞_{n+1} by Lemma 4.7. This concludes the proof of Theorem 4.6. ☐

Remark 4.8 In Remark 4.1, we have seen that the n-fold composition of γ, $\gamma^n : X \to (BT)^n X$ yields a definition of (BT)-behaviour by composing wiht $(BT)^n!_X$. Composing γ^n with θ^n, $\theta^n_X \circ \gamma_n : X \to BT^n X$, yields a definition of trace semantics by composing with $BT^n!_X$. We may thus see θ^n as a semantics of the axiomatisation of infinite trace equivalence.

5 Coalgebra Automata

Coalgebra automata were introduced by Venema [Ven04] as a generalisation of automata operating on streams and infinite trees. The transition structure of a non-deterministic automaton recognising T-coalgebras is a pointed $(\mathcal{P}, T)$-coalgebra augmented with an acceptance condition describing the acceptance behaviour on strictly infinite input.

5.1 Preliminaries of (Non-Deterministic) Coalgebra Automata

Definition 5.1 [Coalgebra Automata] A coalgebra automaton $\mathbb{A} = \langle Q, \theta : Q \to \mathcal{P}TQ, q_I, \Omega\rangle$ with parity acceptance condition consists of a pointed coalgebra $\langle Q, \theta : Q \to \mathcal{P}TQ, q_I\rangle$ and a ranking function $\Omega : Q \to \mathbb{N}$.

The acceptance behaviour of a coalgebra automaton $\mathbb{A}$ with parity acceptance condition is given in terms of acceptance games. These are two-player ($\exists$,$\forall$) parity graph games $\langle V_\exists, V_\forall, E, v_I, \Omega_\mathcal{G}\rangle$. $V_\exists$ and $V_\forall$ bi-partition the set $V = V_\exists \cup V_\forall$ of vertices (positions). The edge relation $E \subseteq V \times V$ specifies admissible moves. $\Omega_\mathcal{G} : V \to \mathbb{N}$ is a ranking function. The acceptance game for a pointed T-coalgebra $\langle S, \tau : S \to TS, S_I \in S\rangle$ is summarised in the following table.

Position	Player	Sets of admissible moves	$\Omega_\mathcal{G}$
$(q, x) \in Q \times S$	$\exists$	$\{(a, x)\mid \mid a \in \theta(q)\}$	$\Omega(q)$
$(a, x) \in TQ \times S$	$\exists$	$\{Z \subseteq Q \times S \mid (a, \tau(x)) \in \mathit{RelT}(Z)\}$	0
$Z \subseteq Q \times S$	$\forall$	$\{(q', x') \in Q \times S \mid (q', x') \in Z\}$	0

$\exists$ wins a finite match, if $\forall$ can not move, and wins an infinite match, if the largest priority (w.r.t. $\Omega_\mathcal{G}$) occurring infinite often is even; otherwise $\forall$ wins.

A coalgebra automaton $\mathbb{A} = \langle Q, \theta, q_I, \Omega\rangle$ accepts a pointed coalgebra $\mathbb{S} = \langle S, \sigma, s_I\rangle$ if $\exists$ has a winning strategy in the acceptance game for $\mathbb{A}$ and $\mathbb{S}$ from (q_I, s_I). The set of all accepted coalgebras is called the language of $\mathbb{A}$.

5.2 Acceptance Behaviour of Automata as Infinite Trace Semantics

Recall from above, that Acc^ω assigns to a state $q \in Q$, the set of points in a T-coalgebra $\langle S, \sigma\rangle$. Composing σ with η_{TS} reveals σ as a $\overline{T}$-coalgebra in $\mathit{Kl}(\mathcal{P})$. To prove $\eta_{TS} \circ \tau$ a $\overline{T}$-coalgebra we need to show the left commuting in $\mathit{Kl}(B)$.

$$\begin{tikzcd}
Q \arrow[r, "\tau"] \arrow[d, "\mathit{Acc}^\Omega"'] & \overline{T}Q \arrow[d, "\overline{T}\mathit{Acc}^\Omega"] \\
S \arrow[r, "\eta_{TS}\circ\sigma"'] & \overline{T}S
\end{tikzcd}
\qquad
\begin{tikzcd}
Q \arrow[rr, "\tau"] \arrow[d, "\mathit{Acc}^\Omega"'] & & \mathcal{P}TQ \arrow[d, "\mathcal{P}\pi\circ\mathcal{P}T\mathit{Acc}^\Omega"] \\
\mathcal{P}S \arrow[r, "\mathcal{P}\sigma"'] & \mathcal{P}TS & \mathcal{P}\mathcal{P}TS \arrow[l, "\mu_{TS}"]
\end{tikzcd}
\tag{16}$$

The right is equivalent because of the axiomatic definition of Kleisli-categories and $\mu \circ \mathcal{P}\eta = id$.

$$\begin{array}{ll}
\overline{T}\mathit{Acc}^\Omega \circ \tau = \eta_{TS} \circ \sigma \circ \mathit{Acc}^\Omega & \textit{in } \mathit{Kl}(B) \\
\mu_{TS} \circ \mathcal{P}\pi \circ \mathcal{P}T\mathit{Acc}^\Omega \circ \tau = \mu \circ \mathcal{P}\eta_{TS} \circ \mathcal{P}\sigma \circ \mathit{Acc}^\Omega & \in \mathit{Set} \\
\mu_{TS} \circ \mathcal{P}\pi \circ \mathcal{P}T\mathit{Acc}^\Omega \circ \tau = \mathcal{P}\sigma \circ \mathit{Acc}^\Omega &
\end{array}$$

Proposition 5.2 *For each priority function Ω, Acc^{Ω} is a $\overline{T}$-coalgebra morphism from $\tau : Q \to \overline{T}Q$ into the coalgebra $\eta_{TS} \circ \sigma : S \to \overline{T}S$.*

Proof. If a point $s \in S$ is accepted by $\mathbb{A}$ from $q \in Q$, $s \in Acc^{\Omega}(q)$, then there are $A \in \tau(q)$ and $Z \subseteq Q \times S$ with $(A, \sigma(s)) \in RelT(Z)$ such that $Z \subseteq Acc^{\Omega}$. For standard *Set*-functors T, the latter is equivalent to $\sigma(s) \in \overline{T}Acc^{\Omega}(A)$. The latter implies $\sigma(s) \in \bigcup \mathcal{P}\overline{T}Acc^{\Omega}[\tau(q)] = \mu_{TS} \circ \mathcal{P}\overline{T}Acc^{\Omega}[\tau(q)]$, so that the right diagram of (16) commutes. □

The coalgebra $\sigma : S \to TS$ in Proposition 5.2 may be replaced with the final coalgebra $\xi : Z \to TZ$. Z is in bijection with the set $Nat(\Delta_1, Seq^T)$ of all infinite words through the final T-sequences. In fact $Acc^{\Omega} : Q \to PZ$ assigns to τ an infinite trace semantics, as the following theorem shows.

Theorem 5.3 *The acceptance condition of a coalgebra automaton induces an infinite trace semantics.*

That Condition 1 of Definition 4.5 is met, follows from Proposition 5.2. For the proof of Condition 2 we use the ideas of the proof of Theorem 4.6. The infinite trace semantics tr^{Ω} factors through the cone of the final T-coalgebra over Seq^T, which simplifies our argument. First, we redefine f_n through the n-fold iteration $\xi^n : Z \to T^n Z$ of ξ.

$$\xi^1 := \xi, \ \xi^{n+1} := T\xi \circ \xi^n \tag{17}$$

Lemma 5.4 *For all $n < \omega$, $f_n = T^n!_Z \circ \xi^n$.*

Proof. The proof of this Lemma is a straightforward argument by successively expanding the definitions of ξ^n and f_n. □

Lemma 5.5 *The cone $(f_n)_{n<\omega}$ commutes laxly with $FSeq^T$.*

Proof. The proof is summarized in the following diagram:

$$\begin{tikzcd}
Q \arrow[r, "tr^{\Omega}_{n+1}"] \arrow[d, "tr^{\Omega}_{n}"] & \mathcal{P}T^nT1 \arrow[d, "\mathcal{P}T^n\eta_{T1}"] \arrow[drr, bend left, "="] & & & \\
\mathcal{P}T^n1 & \mathcal{P}T^n\mathcal{P}T1 \arrow[l, "\mathcal{P}T^n!_{\mathcal{P}T1}"] \arrow[r, "\mathcal{P}\pi^n"] & \mathcal{P}\mathcal{P}T^nT1 \arrow[r, "\mu"] & \mathcal{P}TT^n1 \arrow[r, "\mathcal{P}T^n!_{T1}"] & \mathcal{P}T^n1
\end{tikzcd} \tag{18}$$

The triangle commutes due to naturality of π and $\mu \circ \mathcal{P}\eta = id$. The square commutes by definition of tr^{Ω}_n. Note that the bottom row defines the span $\leq_n$. □

Example 5.6 Jacob's definition of infinite trace semantics [Jac04] arises as a special case of a non-deterministic coalgebra automaton, where $\Omega(q) = 0$ for all automaton states q.

6 Conclusions and Future Directions

We have characterised infinite trace semantics for (B, T)-coalgebras in the form of lax commuting cones over the B-lifting of the final T-sequence, and have defined a generic construction of infinite trace semantics. Generic infinite trace semantics, Jacob's infinite trace semantics of $(\mathcal{P}, T)$-coalgebras, as well as the acceptance behaviour of coalgebra automata proved to be infinite trace semantics.

We have defined infinite trace semantics as a lax commuting cone over $BSeq^T$. We argued that the set Z^∞ of such cones has the structure of a B-algebra. Thereby one requires B to distribute over countable products. This is clear for monads with a presentation by operations and equations. However, the B-algebra structure on Z^∞ is not free. In some cases, for instance for graded A-LTSs, one may recover a coalgebra structure on Z^∞. The transition type for this coalgebra structure is a continuous extension $\widetilde{T}$ of $\overline{T}$ into the Eilenberg-Moore category. $\widetilde{T}$ may be computed as the left Kan-extension. Showing its existence, however, is beyond the scope of this paper. To recover the coalgebra structure on Z^∞, one needs to make additional assumptions which assert a division structure which inverts the action of θ. For graded A-LTSs, it allows to recover from an increasing sequence $(2*, 4a*, 12ab*, ...)$, the (BT)-behaviour $(2*, 2(2a*), 2(2a(3b*))$. We plan to make these ideas formal in succeeding work.

Infinite trace semantics show resemblence of Böhm tree semantics [Bar81,Abr90] of λ-calculus. Consider for instance the transition type $T(-) = \{\surd\} + (-) + A \times (-)$ which allows respectively for successful termination, silent steps, and audible (A-labelled) steps. An increasing sequence for a possible infinite trace semantics would be $(*, a*, a*, a*, ...)$. The sequence stabilises at $a*$, which means that after the a-step only silent steps are possible, that is no further behaviour can be observed. These ideas define $\perp$ semantically; possibly a syntactic characterisation can be obtained.

Venema, Kupke and the author have established [Ven04,KV08,KV09] a correspondence between coalgebra automata and coalgebraic logics with fixpoint operators by means of closure properties. These may be generalised from nondeterministic automata to arbitrarily branching coalgebras with finite carrier. This could lead to a definition of coalgebraic logics for infinite traces. More immediately is an extension of the recent work of the author with Kurz [KK10] on generic trace logics, which from which a coalgebraic modal logics with fixpoint operators could be expected. The class of languages accepted by coalgebra automata has a topological structure which characterises regular languages. Regularity depends on the finiteness of the set of states and on the regularity of the acceptance condition. As of yet it is unclear how this generalises to other (B, T)-coalgebras and what a regular language for other branching types is.

References

[Abr90] Samson Abramsky. The lazy lambda calculus. pages 65–116, 1990.

[Bar81] Henk Barendregt. *The Lambda Calculus - Its Syntax and Semantics*, volume 103 of *Studies in Logic*. North-Holland, 1981.

[HJS06] Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace theory. In *International Workshop on Coalgebraic Methods in Computer Science (CMCS 2006)*, volume 164 of *Elect. Notes in Theor. Comp. Sci.*, pages 47–65. Elsevier, 2006.

[Jac04] Bart Jacobs. Trace semantics for coalgebras. *Electronic Notes in Theoretical Computer Science*, 106:167–184, December 2004.

[KK10] Christian Kissig and Alexander Kurz. Generic trace logics. 2010.

[KV08] Clemens Kupke and Yde Venema. Coalgebraic automata theory: Basic results. Nov 2008.

[KV09] Christian Kissig and Yde Venema. Complementation of coalgebra automata. 2009.

[Mos99] Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(1-3):277–317, 1999.

[SP82] M. B. Smyth and Gordon D. Plotkin. The category theoretic solution of recursive domain equations. *SIAM Journal of Computing*, 11:761–783, 1982.

[Ven04] Yde Venema. Automata and fixed point logics for coalgebras. *Electronic Notes in Computer Science*, 106:355–375, 2004.

[Wor05] James Worrell. On the final sequence of a finitary set functor. *Theoretical Computer Science*, 338(1-3):184–199, June 2005.