A coalgebraic take on regular and $\omega$-regular behaviour for systems with internal moves

Tomasz Brengos

Warsaw University of Technology, Poland

CMCS 2018, April 2018
Our goal

Goal of the talk

Put infinite behaviour with Büchi acceptance condition into a coalgebraic framework...
Our goal

Goal of the talk
Put infinite behaviour with Büchi acceptance condition into a coalgebraic framework...

Wait...
So... we are doing the same as Urabe, Shimizu, Hasuo (CONCUR’16).

Yes, but in a different manner!
Our primary interest

Kleene th. for regular languages

The set of regular languages for $NA$ is closed under $\cup$, $\cdot$, $\emptyset$, $\{\varepsilon\}$ and $(−)^*$. Moreover, it is the smallest set of languages which contains $\{a\}$ and is closed under these operations.

Kleene th. for $\omega$-regular languages

The $\omega$-regular languages for Büchi automata ($=NA$) are of the form

$$\bigcup_{i=1}^{n} R_i^\omega \cdot L_i$$

for regular languages $R_i, L_i$.
Contents

1 Büchi automata and their behaviour
   - Transition systems with silent moves

2 General monad construction

3 Regular and $\omega$-regular behaviour
   - Classical non-deterministic (Büchi) automata
   - Tree automata
   - Kleene theorems for $(T,F)$-automata
Non-deterministic automata with $\varepsilon$-transitions

**Definition**

An $\varepsilon$-NA is a tuple $(X, \Sigma_\varepsilon, \rightarrow, \mathcal{F})$, where $X$ is a set of states, $\Sigma$ is a finite set of alphabet letters and $\rightarrow \subseteq X \times \Sigma_\varepsilon \times X$ and $\mathcal{F} \subseteq X$ is the set of terminal states.

Any $\varepsilon$ – NA can be viewed as a coalgebra, i.e. a map:

$$\alpha' : X \rightarrow \mathcal{P}(\Sigma_\varepsilon \times X + 1);$$

where $1 = \{\sqrt{\ }\}$. However

**Note**

We can also view it as a pair $(\alpha, \mathcal{F})$, where

$$\alpha : X \rightarrow \mathcal{P}(\Sigma_\varepsilon \times X).$$
From our previous work...

Any $\alpha : X \to \mathcal{P}(\Sigma_\varepsilon \times X)$ is a labelled transition system with $\varepsilon$-moves. Other known systems with internal moves:

- Segala systems,
- fully probabilistic systems,
- ...

Systems with internal moves

Coalgebras over a monad $X \to TX$

Endomorphisms in the Kleisli category for $T$
What to do with LTS?

**Strategy 1**
Introduce a monad structure on $\mathcal{P}(\Sigma \times \text{Id})$

**Strategy 2**
Embed the functor $\mathcal{P}(\Sigma \times \text{Id})$ into $\mathcal{P}(\Sigma^* \times \text{Id})$ which is a monad.
What to do with LTS?

**Strategy 1**
Introduce a monad structure on $\mathcal{P}(\Sigma \varepsilon \times I d)$

**Strategy 2**
Embed the functor $\mathcal{P}(\Sigma \varepsilon \times I d)$ into $\mathcal{P}(\Sigma^* \times I d)$ which is a monad.

**Strategy 3**
Embed the functor $\mathcal{P}(\Sigma \varepsilon \times I d)$ into $\mathcal{P}(\Sigma^* \times I d + \Sigma^\omega)$ which is a monad: For $f : X \rightarrow \mathcal{P}(\Sigma^* \times Y + \Sigma^\omega)$ and $g : Y \rightarrow \mathcal{P}(\Sigma^* \times Z + \Sigma^\omega)$ the map $g \cdot f : X \rightarrow \mathcal{P}(\Sigma^* \times Z + \Sigma^\omega)$ is:

$x \xrightarrow{\sigma} g \cdot f z \iff \exists y \text{ s.t. } x \xrightarrow{\sigma_1} f y \text{ and } y \xrightarrow{\sigma_2} g z$, where $\sigma = \sigma_1 \sigma_2 \in \Sigma^*$,

$x \downarrow g \cdot f v \iff x \downarrow f v \text{ or } x \xrightarrow{\sigma} f y, \ y \downarrow g v' \text{ and } v = \sigma v' \in \Sigma^\omega$. 

T.Brengos  A coalgebraic take on regular and $\omega$-regular behaviour for systems
Let’s use the last strategy

\[
\alpha : X \rightarrow \mathcal{P}(\Sigma \varepsilon \times X)
\]

\[
\alpha : X \rightarrow \mathcal{P}(\Sigma^* \times X + \Sigma^\omega)
\]

\[
\alpha : X \rightarrow X \text{ is an endo in } \mathcal{K}l(\mathcal{P}(\Sigma^* \times Id + \Sigma^\omega))
\]

**Interesting observation**

Any subset \(\mathcal{F} \subseteq X\) may be encoded as an endomorphism in \(\mathcal{K}l(\mathcal{P}(\Sigma^* \times Id + \Sigma^\omega))\):

\[
f_{\mathcal{F}} : X \rightarrow \mathcal{P}(\Sigma \varepsilon \times X); x \mapsto \begin{cases} 
\{(\varepsilon, x)\} & \text{if } x \in \mathcal{F}, \\
\emptyset & \text{otherwise}.
\end{cases}
\]
Non-deterministic automata as a pair of endomorphisms

Observation 2

Any non-deterministic automaton can be viewed as a pair of endomorphisms in the Kleisli of $\mathcal{P}(\Sigma^* \times \text{Id} + \Sigma^\omega)$:

$$(\alpha, f_{\tilde{\Sigma}}).$$

How do we derive (in)finite behaviour of $(\alpha, f_{\tilde{\Sigma}})$?
Basic properties of the Kleisli category

The Kleisli of $\mathcal{P}(\Sigma^* \times \text{Id} + \Sigma^\omega)$ is:

- order enriched by $f \leq g \iff f(x) \subseteq g(x)$ for any $x \in X$,
- the ordering is complete,
- it is left distributive, i.e. $f \cdot (g \lor h) = f \cdot g \lor f \cdot h$.

This allows us to consider for any endo $\alpha : X \to X$ the maps $\alpha^* : X \to X$ and $\alpha^\omega : X \to 0$:

$$\alpha^* = \mu x.(\text{id} \lor x \cdot \alpha) \text{ and } \alpha^\omega = \nu x.x \cdot \alpha.$$
Finite behaviour

Finite behaviour of \((\alpha, f_\mathcal{F})\)

\[
! \cdot f_\mathcal{F} \cdot \alpha^* = X \xrightarrow{\alpha^*} X \xrightarrow{f_\mathcal{F}} X \xrightarrow{1} 1 \text{ in Klesli}
\]

Explanation

\[
x \xrightarrow{\alpha^*} \{(a_1a_2\ldots a_n, y) \mid x \xrightarrow{a_1} x_1 \ldots x_{n-1} \xrightarrow{a_n} x_n = y\} \cup \{(\varepsilon, x)\}
\]

\[
x \xrightarrow{f \cdot \alpha^*} \{(a_1a_2\ldots a_n, y) \mid x \xrightarrow{a_1} x_1 \ldots x_{n-1} \xrightarrow{a_n} x_n = y \text{ and } y \in \mathcal{F}\} \cup \{(\varepsilon, x) \mid x \in \mathcal{F}\}.
\]

\[
x \xrightarrow{! \cdot f \cdot \alpha^*} \{(w, 1) \mid w \text{ is accepted by the automaton } (\alpha, \mathcal{F})\}.
\]
The same finite behaviour would be obtained in the Kleisli of $\mathcal{P}(\Sigma^* \times X)$.

Why should we bother with $\mathcal{P}(\Sigma^* \times \text{Id} + \Sigma^\omega)$?
Let \((\alpha, \mathcal{F})\) be an automaton without \(\varepsilon\)-transitions. Then

**Infinite behaviour of \((\alpha, \mathcal{F})\) with BAC**

\[(f_{\mathcal{F}} \cdot \alpha^+)^\omega \text{ in Klesli}\]

In the above

\[\alpha^* = \mu x.(\text{id} \lor x \cdot \alpha), \quad \alpha^+ = \alpha^* \cdot \alpha \text{ and } \beta^\omega = \nu x.x \cdot \beta.\]

**Explanation**

For \(\beta : X \to \mathcal{P}(\Sigma \times X)\) we have \(\beta^\omega : X \to \mathcal{P}(\Sigma^\omega)\):

\[\beta^\omega(x) = \{(a_1, a_2, \ldots) \mid x \xrightarrow{a_1} x_1 \xrightarrow{a_2} x_2 \ldots\}\]
Question 1.

We embedded $\mathcal{P}(\Sigma \epsilon \times \mathcal{I}d)$ into $\mathcal{P}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$.

Problem
Can we do the same with $TF_\epsilon$ for a monad $T$ and $F_\epsilon = F + \mathcal{I}d$?

Goal
We want to embed $TF_\epsilon$ into $TF^\infty$, where

$$F^\infty = \text{the combination of free } F\text{-algebra and final coalgebra}$$

and $TF^\infty$ carries a monadic structure.
Answer to Q1

Take \( \text{Alg}_B(F) \) the category of Bloom algebras of type \( F \).

**Fact (Adámk, Haddadi, Milius, et al. 2014)**

The free Bloom \( F \)-algebra is the combination of the free \( F \)-algebra and the final coalgebra.

We get a monad \( F^\infty \) as the consequence of:  \[ C \leftrightarrow \bot \text{Alg}_B(F) \].

What about \( TF^\infty \)?

**Fact**

If \( F \) lifts to \( \mathcal{K}I(T) \) then the monad \( T \) lifts to a monad \( \bar{T}_B \) on \( \text{Alg}_B(F) \).

\[ C \leftrightarrow \bot \text{Alg}_B(F) \leftrightarrow \mathcal{K}I(\bar{T}_B) \].
### Examples...

**Ex 1**

If $F = \Sigma \times I d$ and $T = \mathcal{P}$ then the monad

$$TF^\infty = \mathcal{P}(\Sigma^* \times I d + \Sigma^\omega)$$

**Ex 2**

If $F = \Sigma \times I d^2$ then $F^\infty X = T_\Sigma X$ is the monad of complete binary finite and infinite trees with nodes in $\Sigma$ and finitely many leaves, all in $X$. Then $\mathcal{P} T_\Sigma (-)$ is the monad of subsets of such trees.

In general, for a pair $(T = \text{monad}, F = \text{functor})$ on Set we consider the monad $TF^\infty$ and define finite and infinite behaviour of $(\alpha, \mathcal{F})$ according to:

$$! \cdot \mathcal{F} \cdot \alpha^* \text{ and } (\mathcal{F} \cdot \alpha^+)^\omega.$$
Coalgebraic modelling
General monad construction
Regular and $\omega$-regular behaviour

Classical non-deterministic (Büchi) automata
Tree automata
Kleene theorems for $(T, F)$-automata

Contents

1 Büchi automata and their behaviour
   - Transition systems with silent moves

2 General monad construction

3 Regular and $\omega$-regular behaviour
   - Classical non-deterministic (Büchi) automata
   - Tree automata
   - Kleene theorems for $(T, F)$-automata
Question 2.

$(-)^*$ and $(-)^\omega$ are operators on endomorphisms in Kleisli. How are they related to the classical operators on languages for finite automata?

**Kleene th. for regular languages**

The set of regular languages for $NA$ is closed under $\cup$, $\cdot$, $\emptyset$, $\{\varepsilon\}$ and $(-)^*$. Moreover, it is the smallest set of languages which contains $\{a\}$ and is closed under these operations.

**Kleene th. for $\omega$-regular languages**

The $\omega$-regular languages for Büchi automata ($=NA$) are of the form

$$\bigcup_{i=1}^{n} R_i^\omega \cdot L_i$$

for regular languages $R_i, L_i$. 
Contents

1. Büchi automata and their behaviour
   - Transition systems with silent moves

2. General monad construction

3. Regular and ω-regular behaviour
   - Classical non-deterministic (Büchi) automata
   - Tree automata
   - Kleene theorems for (T, F)-automata
What about tree automata?

Finite tree automata are: \((\alpha : [n] \to \mathcal{P}(\Sigma \times [n] \times [n]), \mathcal{F} \subseteq [n])\), where \([n] = \{1, 2, \ldots, n\}\) is the state-space. Let \((\omega) \text{Reg} = \) the set of (in)finite tree languages accepted by finite tree automata. We have to introduce \(\text{Rat}_n\).

**Definition**

\(\text{Rat}_n\) is defined to be the smallest set of tree languages with variables in \([n]\) such that it contains \(\{\varepsilon\}, \{i\}\) for \(i \in [n]\), is closed under \(\cup\), and \(T_1, \ldots T_n \in \text{Rat}_m, T \in \text{Rat}_n\) we have:

\[
[T_1, \ldots, T_n] \cdot T \in \text{Rat}_m \text{ and } T^{*,i} \in \text{Rat}_n.
\]

**Classical Kleene th. for tree languages**

\[
\text{Reg} = \text{Rat}_1
\]

\[
\omega\text{Reg} = \{[T_1, \ldots T_n]^{\omega} \cdot T \mid T, T_i \in \text{Rat}_n\}.
\]
# Contents

1. Büchi automata and their behaviour
   - Transition systems with silent moves

2. General monad construction

3. Regular and $\omega$-regular behaviour
   - Classical non-deterministic (Büchi) automata
   - Tree automata
   - Kleene theorems for $(T, F)$-automata
The general picture

1. Start with a finite \((T, F)\)-automaton \((\alpha : X \rightarrow TFX, \mathcal{F} \subseteq X)\).
2. Consider the monad \(TF^\infty\) and look at \((\alpha, \mathcal{F})\) as a pair of endomorphisms \((\alpha, f)\) in the Kleisli for \(TF^\infty\).
3. Define finite and infinite behaviour with BAC:
   \[
   ! \cdot f \cdot \alpha^* \quad (f \cdot \alpha^+)^\omega.
   \]
4. Define \(\text{Rat}_n\) to be the smallest set of elements from \(TF^\omega[n]\) containing 0, \(i\) and being closed under \(\lor\) and
   \[
   [r_1, \ldots r_n] \cdot r \in \text{Rat}_m \text{ and } r^{*i} \in \text{Reg}_n \text{ if } r \in \text{Rat}_n, r_i \in \text{Rat}_m.
   \]

Kleene theorem

\[
\text{Reg} = \text{Rat}_1
\]
\[
\omega\text{Reg} = \{[r_1, \ldots r_n]^\omega \cdot r \mid r, r_i \in \text{Rat}_n\}.
\]
Thank you!