Beyond Kleene

A Kleene Theorem for polynomial coalgebras

Marcello Bonsangue[1,2] Jan Rutten[1,3] Alexandra Silva[1]

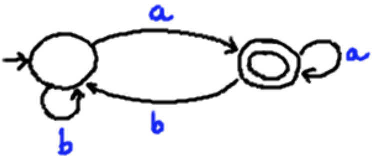
[1]Centrum voor Wiskunde en Informatica
[2]LIACS - Leiden University
[3]Vrije Universiteit Amsterdam

CWI Scientific Meeting, October 2008

Motivation

Deterministic automata (DA)

- Widely used model in Computer Science.
- Acceptors of languages



Regular expressions

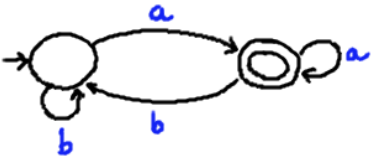
- *User-friendly* alternative to DA notation.
- Many applications: pattern matching (`grep`), specification of circuits, . . .

$$b^*a(b^*a)^*$$

Motivation

Deterministic automata (DA)

- Widely used model in Computer Science.
- Acceptors of languages



Regular expressions

- *User-friendly* alternative to DA notation.
- Many applications: pattern matching (`grep`), specification of circuits, ...

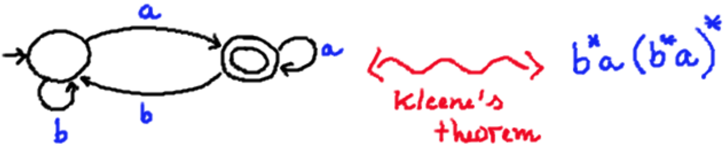
$$b^*a(b^*a)^*$$

Kleene's Theorem

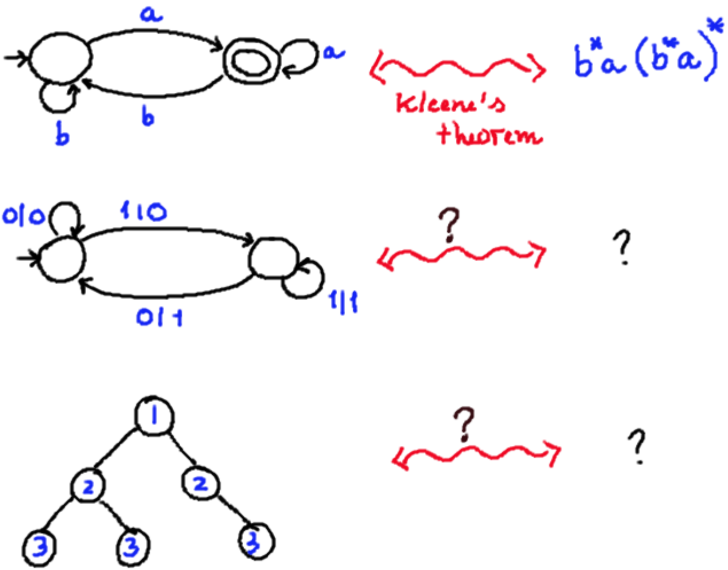
Let $A \subseteq \Sigma^*$. The following are equivalent.

1. $A = L(\mathcal{A})$, for some finite automaton $\mathcal{A}$.
2. $A = L(r)$, for some regular expression r.

Motivation



Motivation



Motivation

Questions

- Can we fill in the ? in the diagram?
- Can we do it uniformly?

Motivation

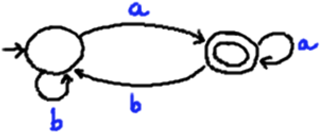
Questions

→ Can we fill in the ? in the diagram?

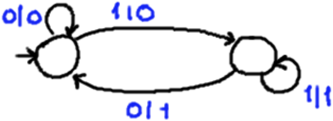
→ Can we do it uniformly?

Yes! :)

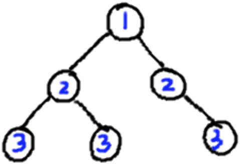
What do these things have in common?



$$(S, \delta : S \to 2 \times S^A)$$

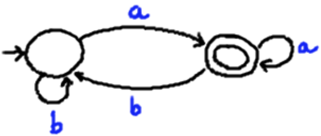


$$(S, \delta : S \to (B \times S)^A)$$

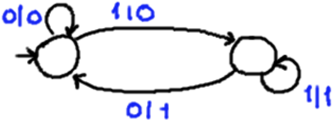


$$(S, \delta : S \to (1 + S) \times A \times (1 + S))$$

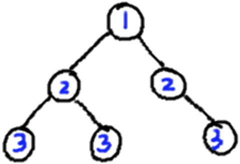
What do these things have in common?



$$(S, \delta : S \to 2 \times S^A)$$

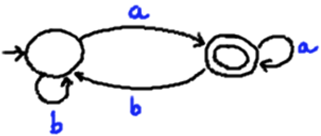


$$(S, \delta : S \to (B \times S)^A)$$

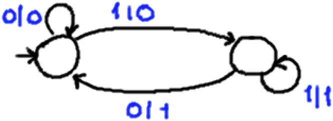


$$(S, \delta : S \to (1 + S) \times A \times (1 + S))$$

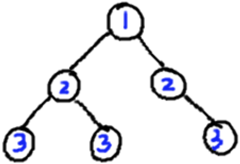
What do these things have in common?



$$(S, \delta : S \to 2 \times S^A)$$



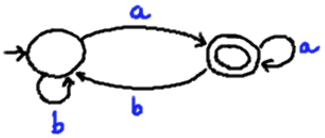
$$(S, \delta : S \to (B \times S)^A)$$



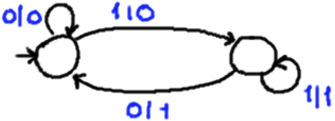
$$(S, \delta : S \to (1 + S) \times A \times (1 + S))$$

$$(S, \delta : S \to GS)$$

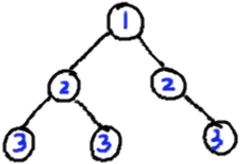
What do these things have in common?



$$(S, \delta : S \to 2 \times S^A)$$



$$(S, \delta : S \to (B \times S)^A)$$



$$(S, \delta : S \to (1 + S) \times A \times (1 + S))$$

$(S, \delta : S \to GS)$ G**-coalgebras**

Coalgebras

Polynomial coalgebras

- Generalizations of deterministic automata
- Polynomial coalgebras: set of states S and $t : S \to GS$

$$G ::= Id \mid B \mid G \times G \mid G + G \mid G^A$$

Examples

- $G = 2 \times Id^A$ Deterministic automata
- $G = (B \times Id)^A$ Mealy machines
- $G = (1 + Id) \times A \times (1 + Id)$ Binary tree automata
- ...

Coalgebras

Polynomial coalgebras

- Generalizations of deterministic automata
- Polynomial coalgebras: set of states S and $t : S \to GS$

$$G ::= Id \mid B \mid G \times G \mid G + G \mid G^A$$

Examples

- $G = 2 \times Id^A$ Deterministic automata
- $G = (B \times Id)^A$ Mealy machines
- $G = (1 + Id) \times A \times (1 + Id)$ Binary tree automata
- ...

Beyond deterministic automata

Deterministic automata
$Q \to 2 \times Q^{\Sigma}$

$\Updownarrow$

Regular Expressions

$\Updownarrow$

Formal Languages

Beyond deterministic automata

Deterministic automata $Q \longrightarrow 2 \times Q^{\Sigma}$	$\rightsquigarrow$	G-coalgebras $Q \longrightarrow GQ$

$$\Updownarrow$$

Regular Expressions

$$\Updownarrow$$

Formal Languages

Beyond deterministic automata

Deterministic automata $Q \rightarrow 2 \times Q^{\Sigma}$	$\rightsquigarrow$	G-coalgebras $Q \rightarrow GQ$
$\Updownarrow$		
Regular Expressions	$\rightsquigarrow$	?
$\Updownarrow$		
Formal Languages	$\rightsquigarrow$	?

Beyond deterministic automata

Deterministic automata $Q \to 2 \times Q^{\Sigma}$	$\rightsquigarrow$	G-coalgebras $Q \to GQ$
$\Updownarrow$		$\Updownarrow$
Regular Expressions	$\rightsquigarrow$	G-expressions
$\Updownarrow$		$\Updownarrow$
Formal Languages	$\rightsquigarrow$	Final coalgebra

In a nutshell

Our contributions are:

- A (syntactic) notion of *G-expressions* for polynomial coalgebras: each expression will denote an element of the final coalgebra.
- We show the equivalence between G-expressions and finite G-coalgebras (analogously to Kleene's theorem).

G-expressions

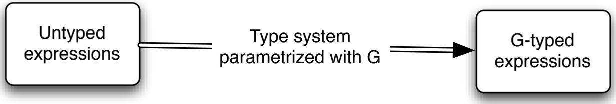
$$E \quad ::= \quad \emptyset \mid \epsilon \mid E \cdot E \mid E + E \mid E^*$$

$$E_G \quad ::= \quad ?$$

G-expressions

$$
\begin{aligned}
E \quad &::= \quad \emptyset \mid \epsilon \mid E \cdot E \mid E + E \mid E^* \\
E_G \quad &::= \quad ?
\end{aligned}
$$

How do we define E_G?



G-expressions

$$
\begin{array}{lllll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon & \mid \mu x.\gamma & \\
 & & & \mid b & B \\
 & & & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle & G_1 \times G_2 \\
 & & & \mid l[\varepsilon] \mid r[\varepsilon] & G_1 + G_2 \\
 & & & \mid a(\varepsilon) & G^A
\end{array}
$$

G-expressions

$$
\begin{array}{llll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon & \mid \mu x.\gamma & \\
 & & & \mid b & B \\
 & & & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle & G_1 \times G_2 \\
 & & & \mid l[\varepsilon] \mid r[\varepsilon] & G_1 + G_2 \\
 & & & \mid a(\varepsilon) & G^A
\end{array}
$$

G-expressions

$$
\begin{array}{llll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon & \mid \mu x.\gamma & \\
 & & & \mid b & B \\
 & & & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle & G_1 \times G_2 \\
 & & & \mid l[\varepsilon] \mid r[\varepsilon] & G_1 + G_2 \\
 & & & \mid a(\varepsilon) & G^A
\end{array}
$$

G-expressions

$$
\begin{array}{rcll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon & \mid \mu x.\gamma \\
 & & & \mid b \qquad\qquad B \\
 & & & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle \qquad G_1 \times G_2 \\
 & & & \mid l[\varepsilon] \mid r[\varepsilon] \qquad G_1 + G_2 \\
 & & & \mid a(\varepsilon) \qquad\qquad G^A
\end{array}
$$

G-expressions

$$
\begin{array}{rclll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon & \mid \mu x.\gamma & \\
& & & \mid b & B \\
& & & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle & G_1 \times G_2 \\
& & & \mid l[\varepsilon] \mid r[\varepsilon] & G_1 + G_2 \\
& & & \mid a(\varepsilon) & G^A
\end{array}
$$

Examples

Deterministic automata expressions – $G = 2 \times Id^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid l\langle 1\rangle \mid l\langle 0\rangle \mid r\langle a(\varepsilon)\rangle$$

Examples

Deterministic automata expressions – $G = 2 \times Id^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid l\langle 1\rangle \mid l\langle 0\rangle \mid r\langle a(\varepsilon)\rangle$$

Mealy expressions – $G = (B \times Id)^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid a(l\langle b\rangle) \mid a(r\langle \varepsilon\rangle)$$

Examples

Deterministic automata expressions – $G = 2 \times Id^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid l\langle 1\rangle \mid l\langle 0\rangle \mid r\langle a(\varepsilon)\rangle$$

Mealy expressions – $G = (B \times Id)^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{a(l\langle b\rangle)}_{a\downarrow b} \mid \underbrace{a(r\langle \varepsilon\rangle)}_{a(\varepsilon)}$$

Examples

Deterministic automata expressions – $G = 2 \times Id^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid l\langle 1\rangle \mid l\langle 0\rangle \mid r\langle a(\varepsilon)\rangle$$

Mealy expressions – $G = (B \times Id)^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{a(l\langle b\rangle)}_{a\downarrow b} \mid \underbrace{a(r\langle \varepsilon\rangle)}_{a(\varepsilon)}$$

Binary tree expressions – $G = (1 + Id) \times A \times (1 + Id)$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid l\langle r[\varepsilon]\rangle \mid l\langle l[*]\rangle \mid a \mid r\langle r[\varepsilon]\rangle \mid r\langle l[*]\rangle$$

Examples

Deterministic automata expressions – $G = 2 \times Id^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid l\langle 1\rangle \mid l\langle 0\rangle \mid r\langle a(\varepsilon)\rangle$$

Mealy expressions – $G = (B \times Id)^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{a(l\langle b\rangle)}_{a\downarrow b} \mid \underbrace{a(r\langle \varepsilon\rangle)}_{a(\varepsilon)}$$

Binary tree expressions – $G = (1 + Id) \times A \times (1 + Id)$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{l\langle r[\varepsilon]\rangle}_{l\langle\varepsilon\rangle} \mid \underbrace{l\langle l[*]\rangle}_{l\uparrow} \mid a \mid \underbrace{r\langle r[\varepsilon]\rangle}_{r\langle\varepsilon\rangle} \mid \underbrace{r\langle l[*]\rangle}_{r\uparrow}$$

Kleene's theorem

The goal is:

$G-expressions$ **correspond to** Finite $G-coalgebras$ and vice-versa.

What does it mean **correspond**?

Final coalgebras exist for polynomial coalgebras.

$$
\begin{tikzcd}
S \arrow[r, dashed, "h"] \arrow[d, "\alpha"'] & \Omega_G \arrow[d, "\omega_G"] & Exp_G \arrow[l, dashed, "\llbracket \cdot \rrbracket"'] \\
GS \arrow[r, dashed, "Gh"'] & G\Omega_G &
\end{tikzcd}
$$

correspond $\equiv$ **mapped to the same element of the final coalgebra**
$\equiv$ **bisimilar**

Kleene's theorem

The goal is:

G – *expressions* **correspond to** Finite G – *coalgebras* and vice-versa.

What does it mean **correspond**?

Final coalgebras exist for polynomial coalgebras.

$$\begin{tikzcd} S \arrow[r, dashed, "h"] \arrow[d, "\alpha"'] & \Omega_G \arrow[d, "\omega_G"] & Exp_G \arrow[l, dashed, "[\![\cdot]\!]"'] \\ GS \arrow[r, dashed, "Gh"'] & G\Omega_G & \end{tikzcd}$$

correspond $\equiv$ **mapped to the same element of the final coalgebra**
$\equiv$ **bisimilar**

Kleene's theorem

The goal is:

$G - expressions$ **correspond to** Finite $G - coalgebras$ and vice-versa.

What does it mean **correspond**?

Final coalgebras exist for polynomial coalgebras.

$$\begin{tikzcd} S \arrow[r, dashed, "h"] \arrow[d, "\alpha"'] & \Omega_G \arrow[d, "\omega_G"] & Exp_G \arrow[l, dashed, "[\![\cdot]\!]"'] \\ GS \arrow[r, dashed, "Gh"'] & G\Omega_G & \end{tikzcd}$$

correspond $\equiv$ **mapped to the same element of the final coalgebra**
$\equiv$ **bisimilar**

Kleene's theorem

The goal is:

$G-$*expressions* **correspond to** Finite $G-$*coalgebras* and vice-versa.

What does it mean **correspond**?

Final coalgebras exist for polynomial coalgebras.

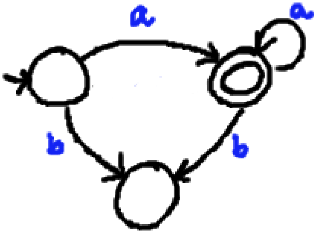
$$\begin{tikzcd} S \arrow[r, dashed, "h"] \arrow[d, "\alpha"'] & \Omega_G \arrow[d, "\omega_G"] & Exp_G \arrow[l, dashed, "[\![\cdot]\!]"'] \\ GS \arrow[r, dashed, "Gh"'] & G\Omega_G & \end{tikzcd}$$

correspond $\equiv$ **mapped to the same element of the final coalgebra**
$\equiv$ **bisimilar**

Theorem

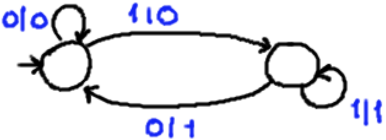
1. *Let (S,g) be a G-coalgebra. If S is* finite *then there exists for any* $s \in S$ *a G-expression* ε_s *such that* $\varepsilon_s \sim s$.
2. *For all G-expressions* ε, *there exists a finite G-coalgebra* (S, g) *such that* $\exists_{s \in S}\ s \sim \varepsilon$.

Examples of application



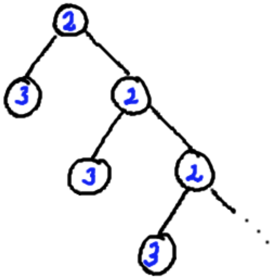
$$\varepsilon \quad = \quad \mu x.a(1 \oplus x)$$

Examples of application



$$\begin{aligned} \varepsilon &= \mu x.\, 0(x) \oplus 1(\varepsilon') \oplus 0{\downarrow}0 \oplus 1{\downarrow}0 \\ \varepsilon' &= \mu y.\, 0(x) \oplus 1(y) \oplus 0{\downarrow}1 \oplus 1{\downarrow}1 \end{aligned}$$

Examples of application



$$\varepsilon = \mu x.2 \oplus r\langle x\rangle \oplus l\langle \underline{3}\rangle$$
$$\underline{a} = a \oplus r\uparrow \oplus l\uparrow$$

Conclusions and Future work

Conclusions

- Language of regular expressions for polynomial coalgebras
- Generalization of Kleene theorem

Future work

- Enlarge the class of functors treated: add $\mathcal{P}$
- Axiomatization of the language
- Model checking

Conclusions and Future work

Conclusions

- Language of regular expressions for polynomial coalgebras
- Generalization of Kleene theorem

Future work

- Enlarge the class of functors treated: add $\mathcal{P}$
- Axiomatization of the language
- Model checking