An algebra for Kripke polynomial coalgebras

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

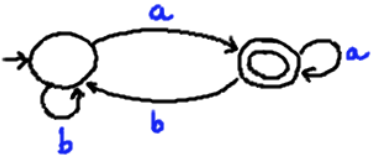
[1]Centrum voor Wiskunde en Informatica
[2]LIACS - Leiden University
[3]Vrije Universiteit Amsterdam

LICS, August 2009

Motivation

Deterministic automata (DA)

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



Regular expressions

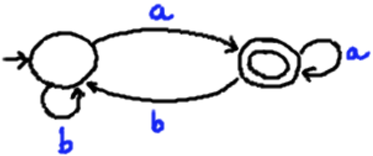
- *User-friendly* alternative to DA notation.
- Many applications: pattern matching (`grep`), specification of circuits, ...

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

Motivation

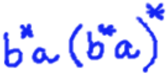
Deterministic automata (DA)

- 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, ...



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

Kleene Algebras

- Kleene asked for a complete set of axioms which would allow derivation of all equations among regular expressions.
- Kozen showed that the axioms of Kleene algebras solve this problem.

Axioms

$$
\begin{array}{rcl}
E_1 + E_2 & = & E_2 + E_1 \\
E_1 + (E_2 + E_3) & = & (E_1 + E_2) + E_3 \\
E_1 + E_1 & = & E_1 \\
E + \emptyset & = & E \\
& \vdots & \\
1 + aa^* & \leq & a^* \\
ax \leq x \rightarrow a^* x & \leq & x
\end{array}
$$

Motivation

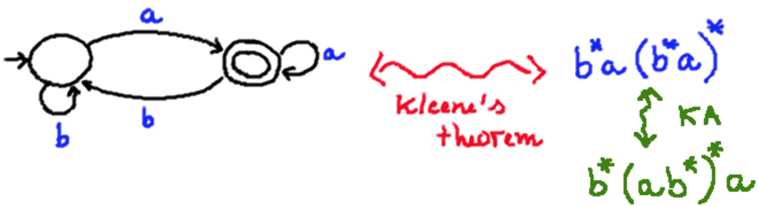
Kleene Algebras

- Kleene asked for a complete set of axioms which would allow derivation of all equations among regular expressions.
- Kozen showed that the axioms of Kleene algebras solve this problem.

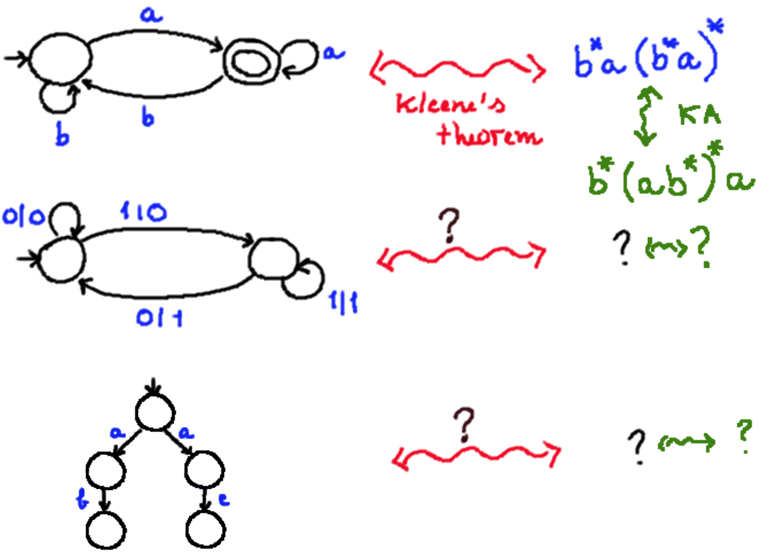
Axioms

$$
\begin{array}{rcl}
E_1 + E_2 & = & E_2 + E_1 \\
E_1 + (E_2 + E_3) & = & (E_1 + E_2) + E_3 \\
E_1 + E_1 & = & E_1 \\
E + \emptyset & = & E \\
& \vdots & \\
1 + aa^* & \leq & a^* \\
ax \leq x \rightarrow a^* x & \leq & x
\end{array}
$$

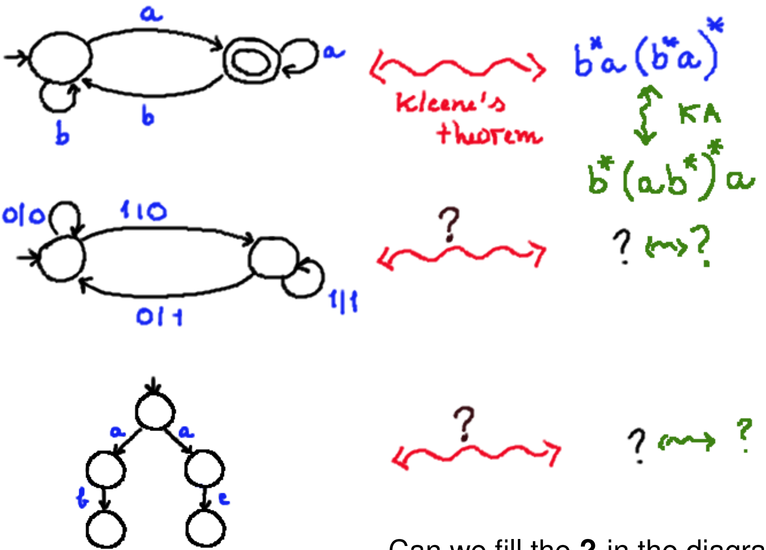
Motivation



Motivation

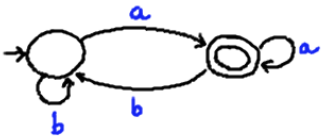


Motivation

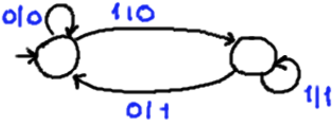


Can we fill the **?** in the diagram?

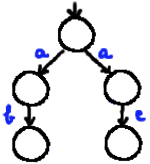
What do these things have in common?



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

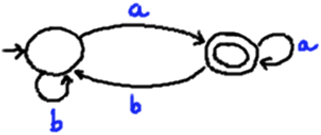


$(S, \delta : S \to (B \times S)^A)$

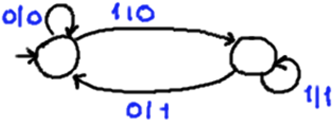


$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$

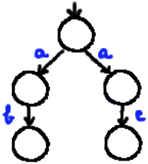
What do these things have in common?



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

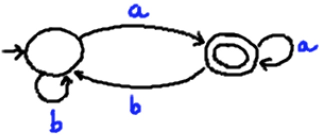


$$(S, \delta : S \to (B \times S)^A)$$

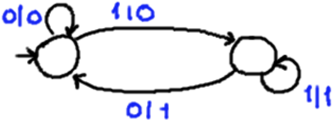


$$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$$

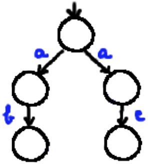
What do these things have in common?



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

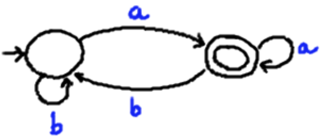


$(S, \delta : S \to (B \times S)^A)$

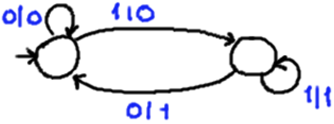


$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$

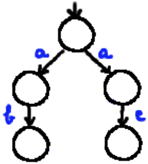
What do these things have in common?



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

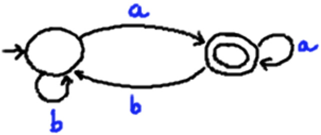


$$(S, \delta : S \to (B \times S)^A)$$

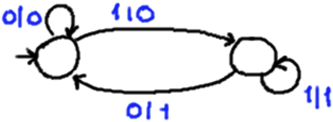


$$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$$

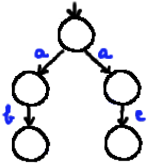
What do these things have in common?



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

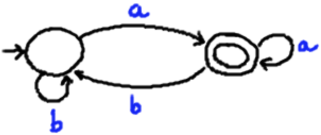


$(S, \delta : S \to (B \times S)^A)$

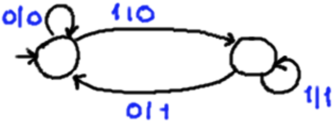


$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$

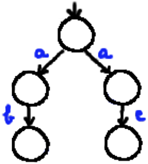
What do these things have in common?



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



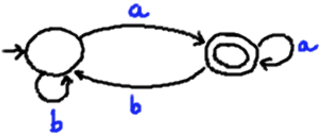
$$(S, \delta : S \to (B \times S)^A)$$



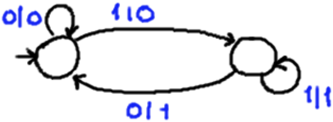
$$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$$

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

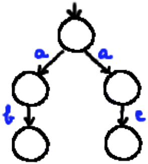
What do these things have in common?



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



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



$(S, \delta : S \to 1 + (\mathcal{P}S)^A)$

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

Coalgebras

Kripke polynomial coalgebras

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

$$G ::= Id \mid B \mid G \times G \mid G + G \mid G^A \mid \mathcal{P}G$$

$\mathcal{P}$ finite

Examples

- $G = 2 \times Id^A$ Deterministic automata
- $G = (B \times Id)^A$ Mealy machines
- $G = 1 + (\mathcal{P}Id)^A$ LTS (with explicit termination)
- ...

Coalgebras

Kripke polynomial coalgebras

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

$$G ::= Id \mid B \mid G \times G \mid G + G \mid G^A \mid \mathcal{P}G$$

$\mathcal{P}$ finite

Examples

- $G = 2 \times Id^A$ Deterministic automata
- $G = (B \times Id)^A$ Mealy machines
- $G = 1 + (\mathcal{P}Id)^A$ LTS (with explicit termination)
- ...

The power of G

The functor G determines:

1. notion of observational equivalence (coalg. bisimulation)
2. behaviour (final coalgebra)
3. set of expressions describing finite systems
4. axioms to prove bisimulation equivalence of expressions

The power of G

The functor G determines:

1. notion of observational equivalence (coalg. bisimulation)
2. behaviour (final coalgebra)
3. set of expressions describing finite systems
4. axioms to prove bisimulation equivalence of expressions

The power of G

The functor G determines:

1. notion of observational equivalence (coalg. bisimulation)
2. behaviour (final coalgebra)
3. set of expressions describing finite systems
4. axioms to prove bisimulation equivalence of expressions

1 + 2 are classic coalgebra; 3 is FoSSaCS 08 and 09; 4 is LICS 09

In a nutshell — 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 previous work (FoSSaCS'09):

- Framework where a (syntactic) notion of *G-expressions* for polynomial coalgebras can be uniformly derived.
- Proved equivalence between G-expressions and finite G-coalgebras (analogously to Kleene's theorem).

In a nutshell — 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 previous work (FoSSaCS'09):

- Framework where a (syntactic) notion of *G-expressions* for polynomial coalgebras can be uniformly derived.
- Proved equivalence between G-expressions and finite G-coalgebras (analogously to Kleene's theorem).

In this paper, we ...

- ...extend previous work (expressions + Kleene's theorem) to non-deterministic systems.
- ...provide an axiomatization for the expressions (thus generalizing Kleene algebras), parametric on G, and
- prove it sound a complete wrt $\sim$, in a purely coalgebraic fashion.

G-expressions

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

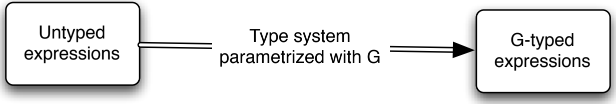
$$E_G \; ::= \; ?$$

G-expressions

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

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

How do we define E_G?



G-expressions

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

G-expressions

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

G-expressions

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

G-expressions

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

G-expressions

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

G-expressions

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

Examples

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

$$\varepsilon \quad ::= \quad \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma}_{G} \mid$$

Examples

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

$$\varepsilon \quad ::= \quad \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma}_{G} \mid \underbrace{l\langle \qquad \rangle \qquad\qquad \mid r\langle \qquad \rangle}_{\times}$$

Examples

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

$$\varepsilon \quad ::= \quad \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma}_{G} \mid \underbrace{l\langle \underbrace{1}_{2} \rangle \mid l\langle \underbrace{0}_{2} \rangle \mid r\langle \underbrace{a(\varepsilon)}_{Id^A} \rangle}_{\times}$$

Examples

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

$$\varepsilon \;\; ::= \;\; \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma}_{G} \mid \underbrace{l\langle \underbrace{1}_{2} \rangle \mid l\langle \underbrace{0}_{2} \rangle \mid r\langle \underbrace{a(\varepsilon)}_{Id^A} \rangle}_{\times}$$

LTS expressions – $G = 1 + (\mathcal{P} Id)^A$

$$\varepsilon \;\; ::= \;\; \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma$$

Examples

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

$$\varepsilon \quad ::= \quad \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma}_{G} \mid \underbrace{l\langle \underbrace{1}_{2} \rangle \mid l\langle \underbrace{0}_{2} \rangle \mid r\langle \underbrace{a(\varepsilon)}_{Id^A} \rangle}_{\times}$$

LTS expressions – $G = 1 + (\mathcal{P}Id)^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \quad \surd \quad \mid \quad \delta \quad \mid \quad a.\varepsilon$$

Examples

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

$$\varepsilon \quad ::= \quad \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma}_{G} \mid \underbrace{l\langle \underbrace{1}_{2} \rangle \mid l\langle \underbrace{0}_{2} \rangle \mid r\langle \underbrace{a(\varepsilon)}_{Id^A} \rangle}_{\times}$$

LTS expressions – $G = 1 + (\mathcal{P} Id)^A$

$$\varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{\surd}_{l[*]} \mid \underbrace{\delta}_{r[\emptyset]} \mid \underbrace{a.\varepsilon}_{r[a(\{\varepsilon\})]}$$

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 Kripke polynomial coalgebras.

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 Kripke polynomial coalgebras.

$$\begin{array}{ccccc} S & \overset{h}{\dashrightarrow} & \Omega_G & \overset{[\![\cdot]\!]}{\dashleftarrow} & Exp_G \\ \downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \omega_G} & & \\ GS & \underset{Gh}{\dashrightarrow} & G\Omega_G & & \end{array}$$

The goal is:

$G-$ *expressions* **correspond to** Finite $G-$ *coalgebras* and vice-versa.
What does it mean **correspond**?

Final coalgebras exist for Kripke polynomial coalgebras.

$$
\begin{array}{ccccc}
S & \xrightarrow{\ h\ } & \Omega_G & \xleftarrow{\ [\![\cdot]\!]\ } & Exp_G \\
\downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \omega_G} & & \downarrow{\scriptstyle \lambda_G} \\
GS & \xrightarrow[\ Gh\ ]{} & G\Omega_G & \xleftarrow[\ G[\![\cdot]\!]\ ]{} & GExp_G
\end{array}
$$

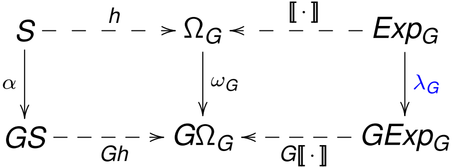
Kleene's theorem

The goal is:

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

What does it mean **correspond**?

Final coalgebras exist for Kripke polynomial coalgebras.



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

$\equiv$ **bisimilar**

A generalized Kleene theorem

G-coalgebras $\Leftrightarrow$ G-expressions

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* $\exists_{s \in S}\ s \sim \varepsilon$.

Axiomatization

$$\left.\begin{array}{lcl}\varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon\end{array}\right\} G$$

$$\left.\begin{array}{lcl}\mu x.\gamma & \equiv & \gamma[\mu x.\gamma/x] \\ \gamma[\varepsilon/x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon\end{array}\right\} FP$$

$$\left.\begin{array}{lcl}\emptyset & \equiv & \bot_B \\ b_1 \oplus b_2 & \equiv & b_1 \vee b_2\end{array}\right\} B$$

Sound and complete w.r.t $\sim$

$$\left.\begin{array}{lcl}l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2)\end{array}\right\} G_1 \times G_2$$

Similar for $G_1 + G_2$ and G^A

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} G$$

$$\left.\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma/x] \\ \gamma[\varepsilon/x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}\right\} FP$$

$$\left.\begin{array}{lcl} \emptyset & \equiv & \bot_B \\ b_1 \oplus b_2 & \equiv & b_1 \vee b_2 \end{array}\right\} B$$

Sound and complete w.r.t $\sim$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} G_1 \times G_2$$

Similar for $G_1 + G_2$ and G^A

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} G$$

$$\left.\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma/x] \\ \gamma[\varepsilon/x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}\right\} FP$$

$$\left.\begin{array}{lcl} \emptyset & \equiv & \bot_B \\ b_1 \oplus b_2 & \equiv & b_1 \vee b_2 \end{array}\right\} B$$

Sound and complete w.r.t $\sim$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} G_1 \times G_2$$

Similar for $G_1 + G_2$ and G^A

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} G$$

$$\left.\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma / x] \\ \gamma[\varepsilon / x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}\right\} FP$$

$$\left.\begin{array}{lcl} \emptyset & \equiv & \bot_B \\ b_1 \oplus b_2 & \equiv & b_1 \vee b_2 \end{array}\right\} B$$

Sound and complete w.r.t $\sim$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} G_1 \times G_2$$

Similar for $G_1 + G_2$ and G^A

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} G$$

$$\left.\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma/x] \\ \gamma[\varepsilon/x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}\right\} FP$$

$$\left.\begin{array}{lcl} \emptyset & \equiv & \perp_B \\ b_1 \oplus b_2 & \equiv & b_1 \vee b_2 \end{array}\right\} B$$

Sound and complete w.r.t $\sim$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} G_1 \times G_2$$

Similar for $G_1 + G_2$ and G^A

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} G$$

$$\left.\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma/x] \\ \gamma[\varepsilon/x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}\right\} FP$$

$$\left.\begin{array}{lcl} \emptyset & \equiv & \perp_B \\ b_1 \oplus b_2 & \equiv & b_1 \vee b_2 \end{array}\right\} B$$

Sound and complete w.r.t $\sim$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} G_1 \times G_2$$

Similar for $G_1 + G_2$ and G^A

Axiomatization – example

LTS expressions – $G \equiv 1 + (\mathcal{P} Id)^A$

$$\varepsilon ::\equiv \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{\surd}_{l[*]} \mid \underbrace{\delta}_{r[\emptyset]} \mid \underbrace{a.\varepsilon}_{r[a(\{\varepsilon\})]}$$

$$\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \\ \varepsilon \oplus \delta & \equiv & \varepsilon \end{array}$$

No rule

$a.(\varepsilon_1 \oplus \varepsilon_2) \equiv a.\varepsilon_1 \oplus a.\varepsilon_2$

$$\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma / x] \\ \gamma[\varepsilon / x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}$$

Axiomatization – example

LTS expressions – $G \equiv 1 + (\mathcal{P} Id)^A$

$$\varepsilon ::\equiv \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\gamma \mid \underbrace{\surd}_{l[*]} \mid \underbrace{\delta}_{r[\emptyset]} \mid \underbrace{a.\varepsilon}_{r[a(\{\varepsilon\})]}$$

$$\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1 \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \\ \varepsilon \oplus \delta & \equiv & \varepsilon \end{array}$$

No rule

$a.(\varepsilon_1 \oplus \varepsilon_2) \equiv a.\varepsilon_1 \oplus a.\varepsilon_2$

$$\begin{array}{lcl} \mu x.\gamma & \equiv & \gamma[\mu x.\gamma / x] \\ \gamma[\varepsilon / x] \equiv \varepsilon & \Rightarrow & \mu x.\gamma \equiv \varepsilon \end{array}$$

A coalgebraic proof of soundness and completeness

Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness

$$\varepsilon_1 \equiv \varepsilon_2$$

$$\Leftrightarrow \quad \varepsilon_1 \sim \varepsilon_2$$

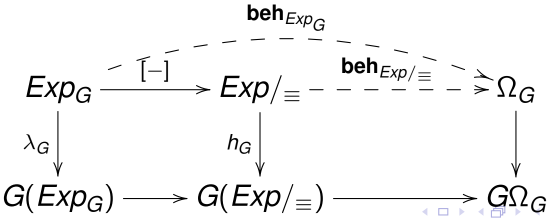
Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness

$\varepsilon_1 \equiv \varepsilon_2$

$\Leftrightarrow \quad \varepsilon_1 \sim \varepsilon_2$



A coalgebraic proof of soundness and completeness

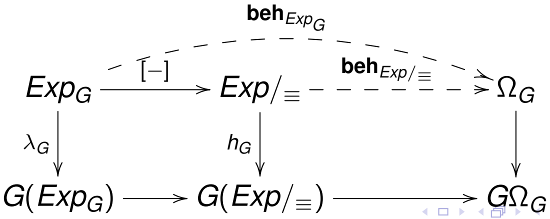
Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness

$$\varepsilon_1 \equiv \varepsilon_2 \quad \Leftrightarrow \quad [\varepsilon_1] = [\varepsilon_2]$$

$$\Leftrightarrow \quad \varepsilon_1 \sim \varepsilon_2$$

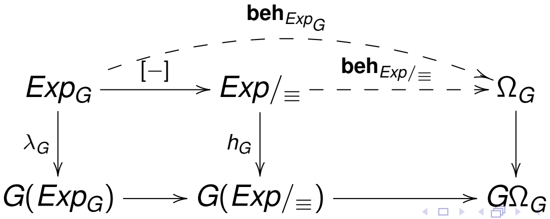


Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness

$$
\begin{array}{rcl}
\varepsilon_1 \equiv \varepsilon_2 & \Leftrightarrow & [\varepsilon_1] = [\varepsilon_2] \\
& \Rightarrow & \mathbf{beh}_{Exp/_\equiv}([\varepsilon_1]) = \mathbf{beh}_{Exp/_\equiv}([\varepsilon_2]) \\
& \Leftrightarrow & \varepsilon_1 \sim \varepsilon_2
\end{array}
$$



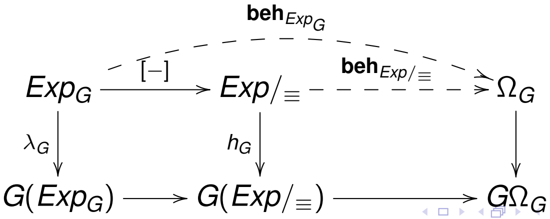
A coalgebraic proof of soundness and completeness

Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness

$$
\begin{array}{lll}
\varepsilon_1 \equiv \varepsilon_2 & \Leftrightarrow & [\varepsilon_1] = [\varepsilon_2] \\
 & \Rightarrow & \mathbf{beh}_{Exp/_\equiv}([\varepsilon_1]) = \mathbf{beh}_{Exp/_\equiv}([\varepsilon_2]) \\
 & \Leftrightarrow & \mathbf{beh}_{Exp_G}(\varepsilon_1) = \mathbf{beh}_{Exp_G}(\varepsilon_2) \\
 & \Leftrightarrow & \varepsilon_1 \sim \varepsilon_2
\end{array}
$$



A coalgebraic proof of soundness and completeness

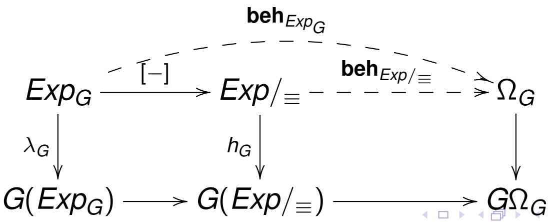
Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness

$$
\begin{array}{rcl}
\varepsilon_1 \equiv \varepsilon_2 & \Leftrightarrow & [\varepsilon_1] = [\varepsilon_2] \\
 & \Rightarrow & \mathbf{beh}_{Exp/_{\equiv}}([\varepsilon_1]) = \mathbf{beh}_{Exp/_{\equiv}}([\varepsilon_2]) \\
 & \Leftrightarrow & \mathbf{beh}_{Exp_G}(\varepsilon_1) = \mathbf{beh}_{Exp_G}(\varepsilon_2) \\
 & \Leftrightarrow & \varepsilon_1 \sim \varepsilon_2
\end{array}
$$

$\Leftrightarrow$: $[-]$ homomorphism



A coalgebraic proof of soundness and completeness

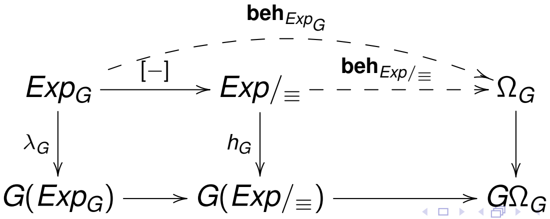
Theorem

$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness and Completeness

$$
\begin{aligned}
\varepsilon_1 \equiv \varepsilon_2 \quad & \Leftrightarrow \quad [\varepsilon_1] = [\varepsilon_2] \\
& \Leftrightarrow \quad \mathbf{beh}_{Exp/_\equiv}([\varepsilon_1]) = \mathbf{beh}_{Exp/_\equiv}([\varepsilon_2]) \\
& \Leftrightarrow \quad \mathbf{beh}_{Exp_G}(\varepsilon_1) = \mathbf{beh}_{Exp_G}(\varepsilon_2) \\
& \Leftrightarrow \quad \varepsilon_1 \sim \varepsilon_2
\end{aligned}
$$

$\Leftrightarrow$: $[-]$ homomorphism



A coalgebraic proof of soundness and completeness

Theorem

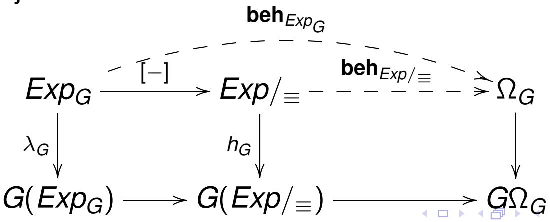
$\varepsilon_1 \sim \varepsilon_2 \iff \varepsilon_1 \equiv \varepsilon_2$

Soundness and Completeness

$$\begin{aligned}
\varepsilon_1 \equiv \varepsilon_2 &\Leftrightarrow [\varepsilon_1] = [\varepsilon_2] \\
&\Leftrightarrow \mathbf{beh}_{Exp/_\equiv}([\varepsilon_1]) = \mathbf{beh}_{Exp/_\equiv}([\varepsilon_2]) \\
&\Leftrightarrow \mathbf{beh}_{Exp_G}(\varepsilon_1) = \mathbf{beh}_{Exp_G}(\varepsilon_2) \\
&\Leftrightarrow \varepsilon_1 \sim \varepsilon_2
\end{aligned}$$

$\Leftrightarrow$: $[-]$ homomorphism

$\Leftrightarrow$: $\mathbf{beh}_{Exp/_\equiv}$ injective



Conclusions and Future work

Conclusions

- Framework to uniformly derive language and axioms for Kripke polynomial coalgebras
- Generalization of Kleene theorem and Kleene algebra, parametric on the functor.
- Proof of soundness and completeness purely coalgebraic

Future work

- Automation: `Circ` — Coinductive prover
- Deriving syntax and axioms for quantitative systems (probabilistic/weighted automata): CONCUR'09

Conclusions and Future work

Conclusions

- Framework to uniformly derive language and axioms for Kripke polynomial coalgebras
- Generalization of Kleene theorem and Kleene algebra, parametric on the functor.
- Proof of soundness and completeness purely coalgebraic

Future work

- Automation: `Circ` — Coinductive prover
- Deriving syntax and axioms for quantitative systems (probabilistic/weighted automata): CONCUR'09