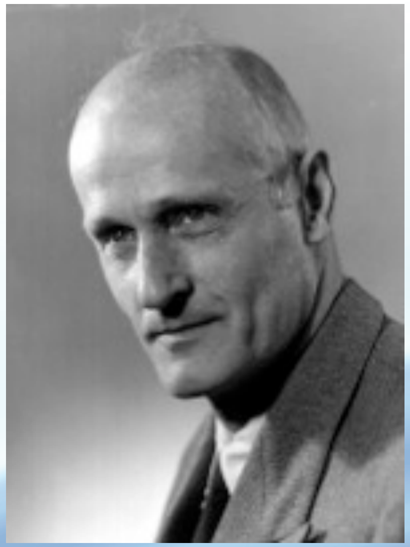
Kleene Coalgebra

Alexandra Silva

(joint work with Jan Rutten, Marcello Bonsangue and Filippo Bonchi)

History



Kleene proposed a *language* to talk about the most basic state based system

History

Regular expressions

Kleene proposed a *language* to talk about the most basic state based system

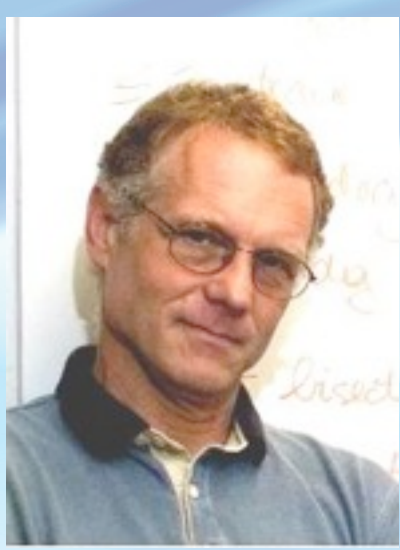
Deterministic automata

History

Regular expressions

Kleene proposed a *language* to talk about the most basic state based system

Deterministic automata



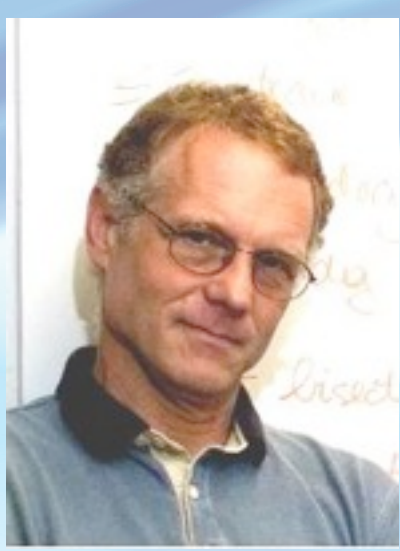
Kozen proposed a fully algebraic set of equations to reason about Kleene's language

History

Regular expressions

Kleene proposed a *language* to talk about the most basic state based system

Deterministic automata



Kozen proposed a fully algebraic set of equations to reason about Kleene's language

Kleene algebra

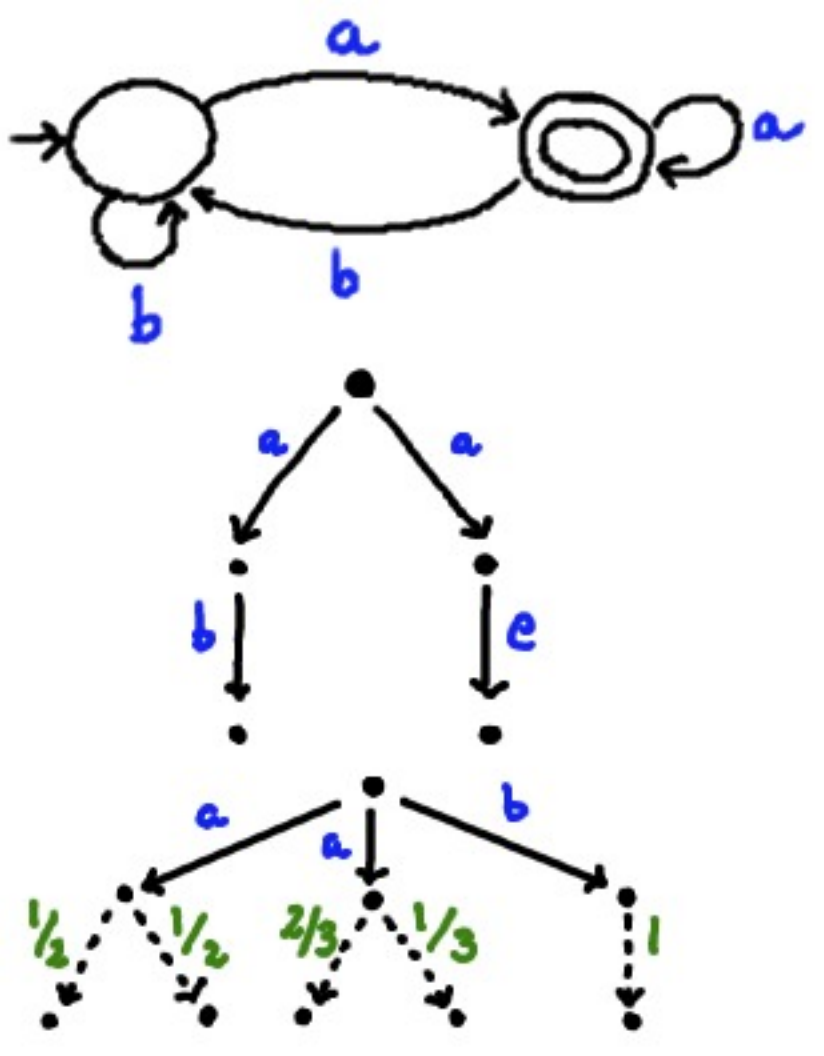
Motivation

Specify and reason about systems

Motivation

Specify and reason about systems

state-machines



Motivation

Specify and reason about systems

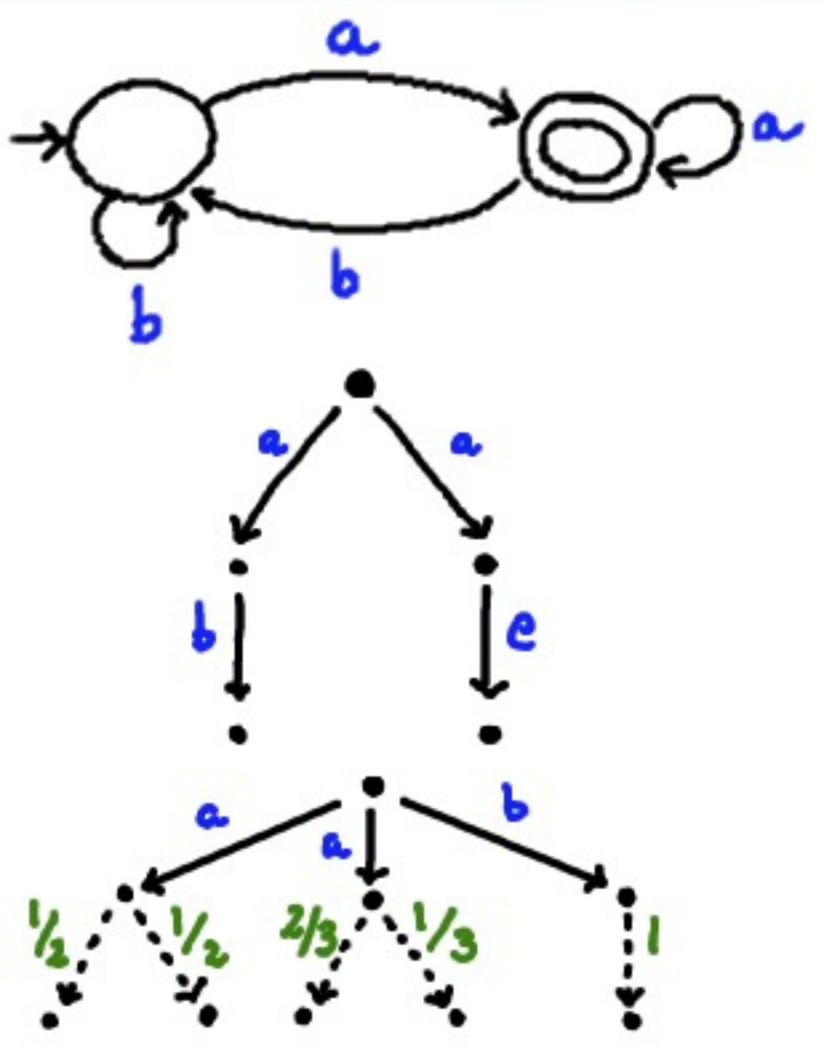
Syntax
(RE, CCS, ...)

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

$a.b.0 + a.c.0$

$a.(\frac{1}{2}\cdot 0 \oplus \frac{1}{2}\cdot 0) + \cdots$

state-machines



Motivation

Specify and reason about systems

Syntax
(RE, CCS, ...)

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

$a.b.0 + a.c.0$

$a.(\frac{1}{2} \cdot 0 \oplus \frac{1}{2} \cdot 0) + \cdots$

Axiomatization
(KA, ...)

$1 + a a^* = a^*$

$\vdots$

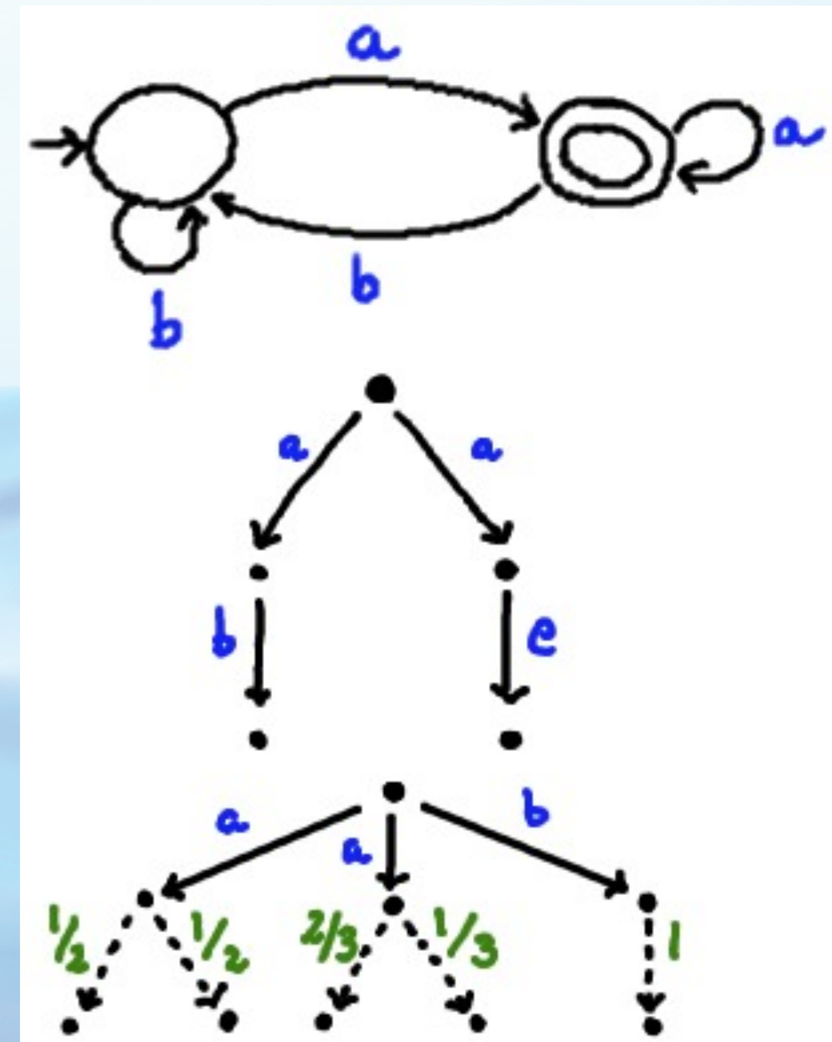
$P + 0 = P$

$\vdots$

$p.P \oplus p'.P = (p + p').P$

$\vdots$

state-machines



Motivation

Specify and reason about systems

Syntax (RE, CCS, ...)	Axiomatization (KA, ...)	state-machines

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

$a.b.0 + a.c.0$

$a.(\frac{1}{2}\cdot 0 \oplus \frac{1}{2}\cdot 0) + \cdots$

$1 + a a^* = a^*$

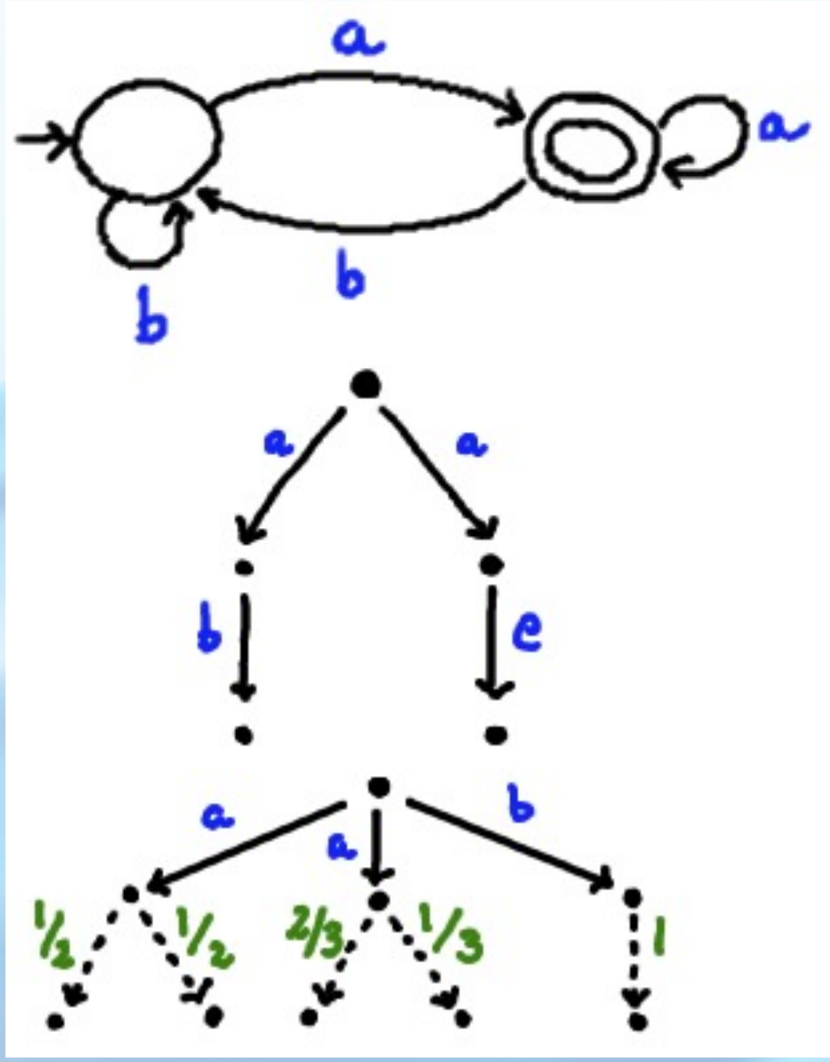
$\vdots$

$P + 0 = P$

$\vdots$

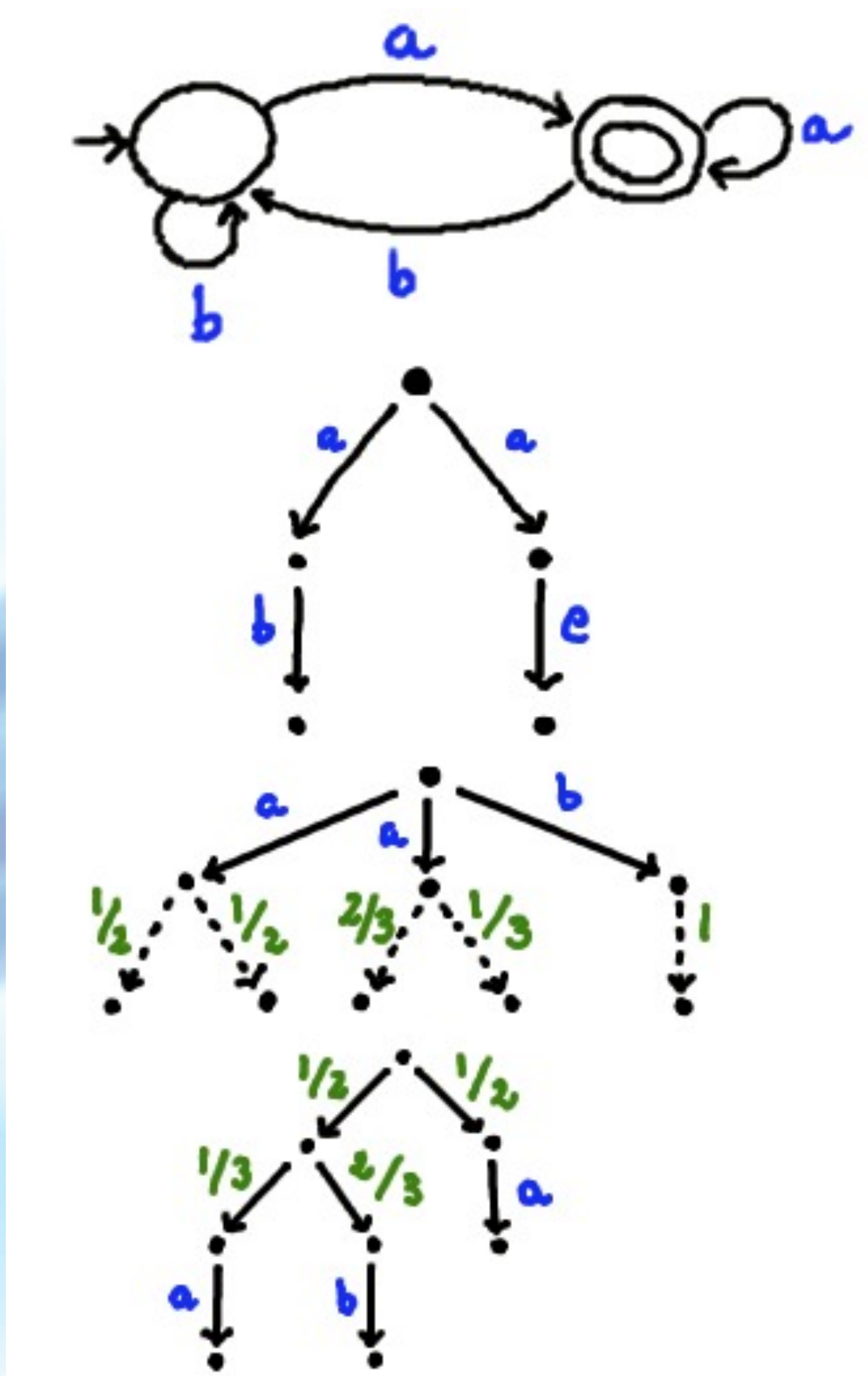
$p.P \oplus p'.P = (p+p').P$

$\vdots$

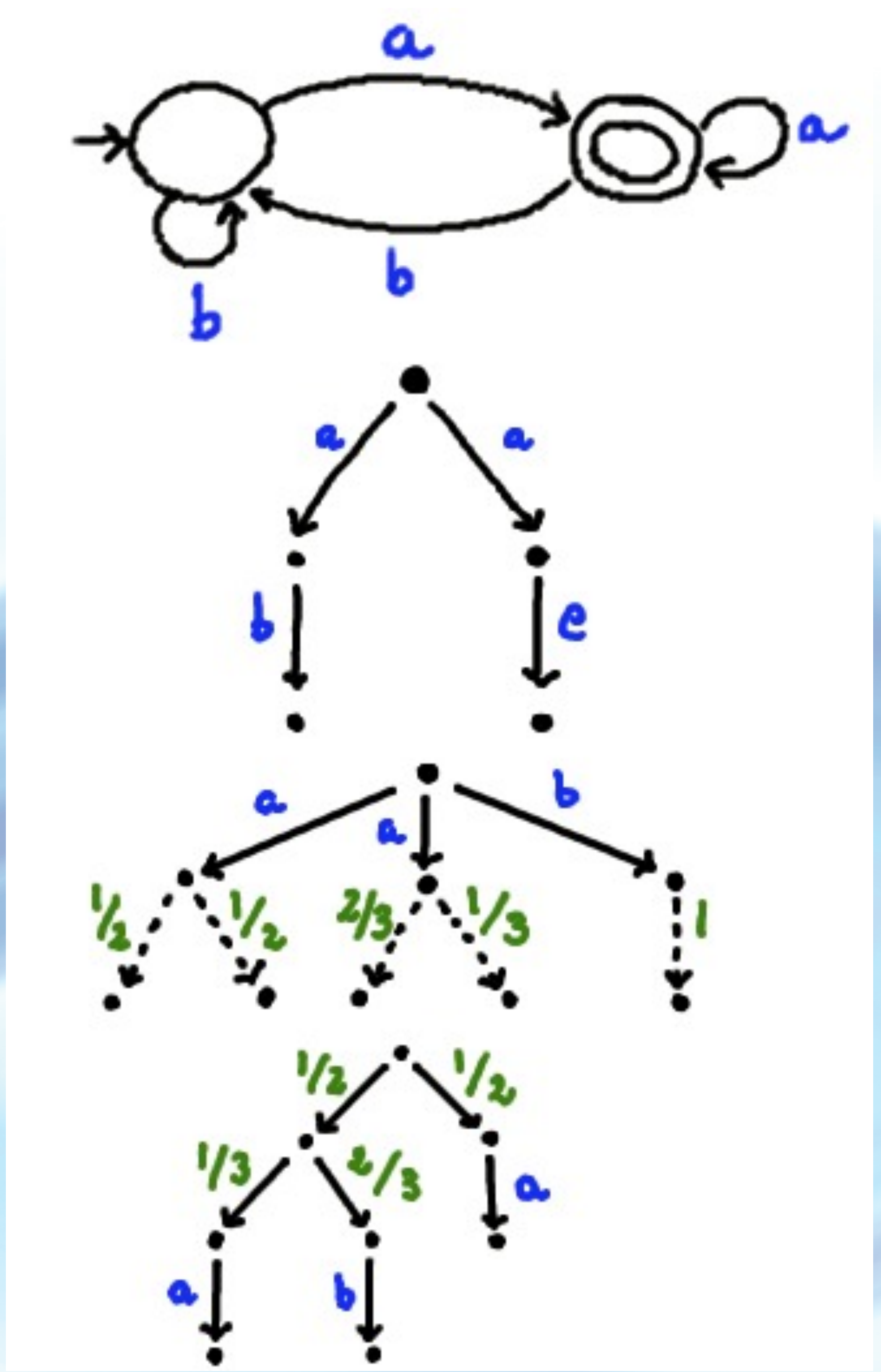


Can we do all of this uniformly in a single framework?

What do these things have in common?

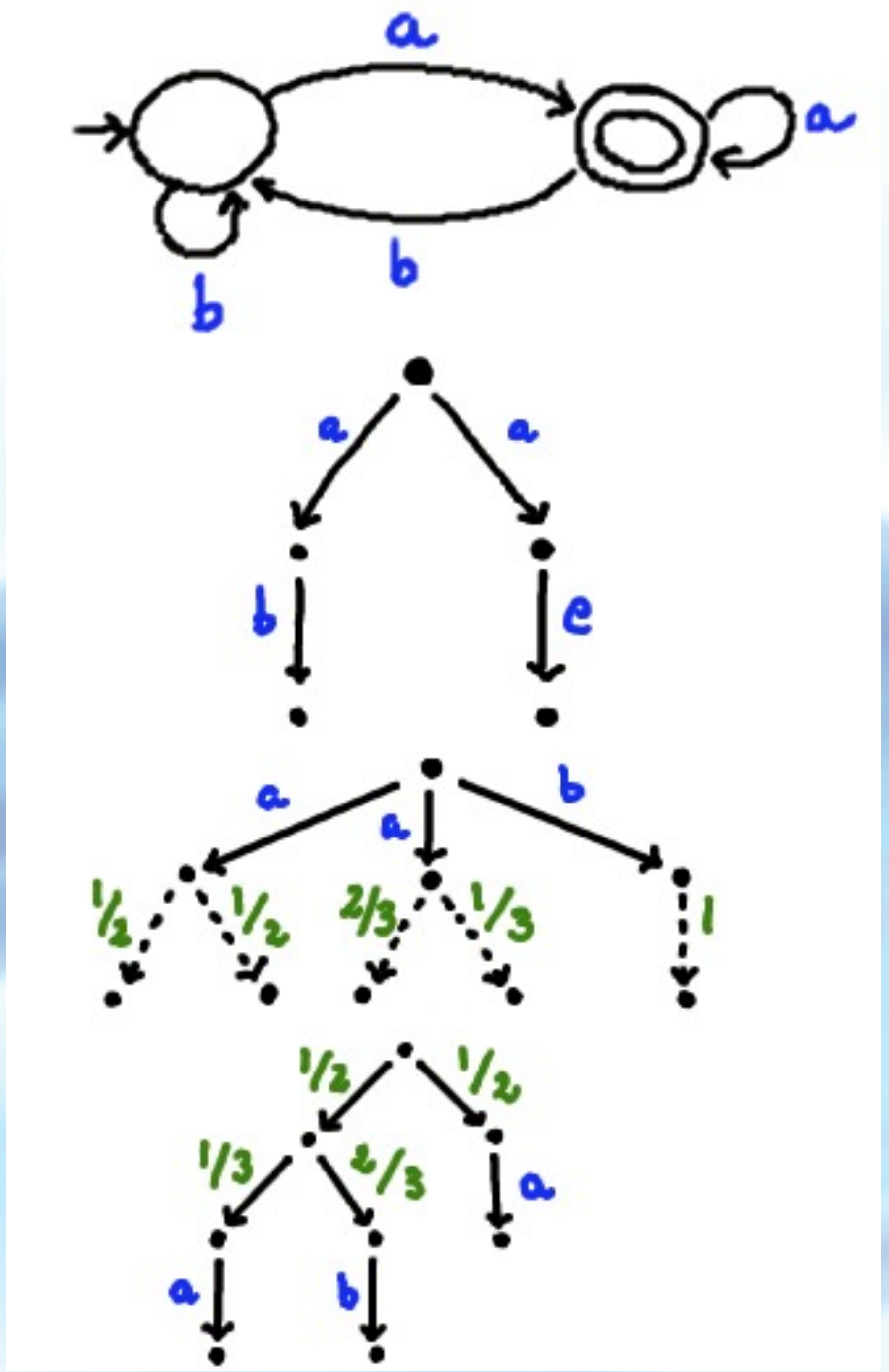


What do these things have in common?



$$(S, f : S \longrightarrow 2 \times S^A)$$

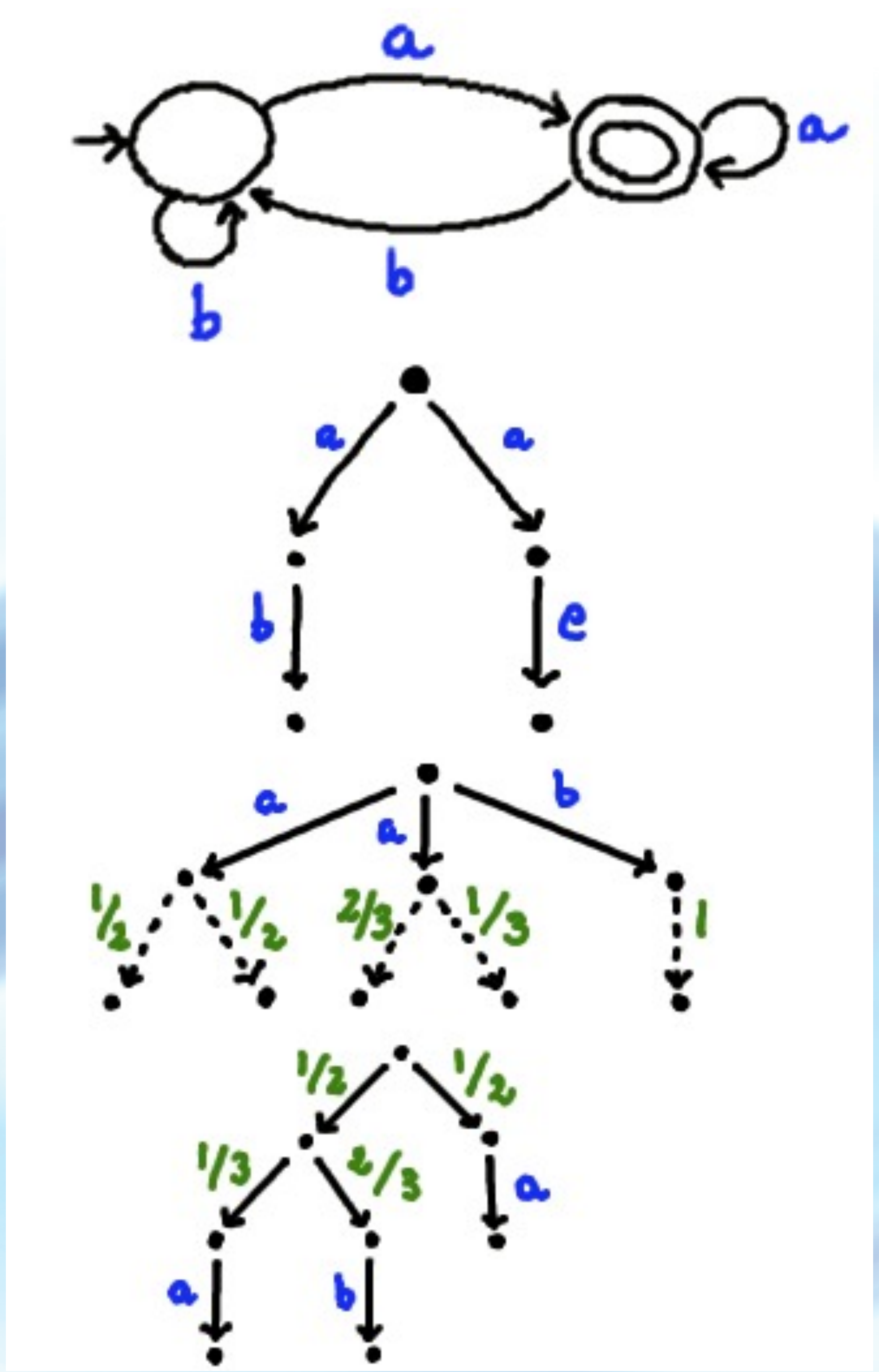
What do these things have in common?



$(S, f : S \longrightarrow 2 \times S^A)$

$(S, f : S \longrightarrow P(S)^A)$

What do these things have in common?

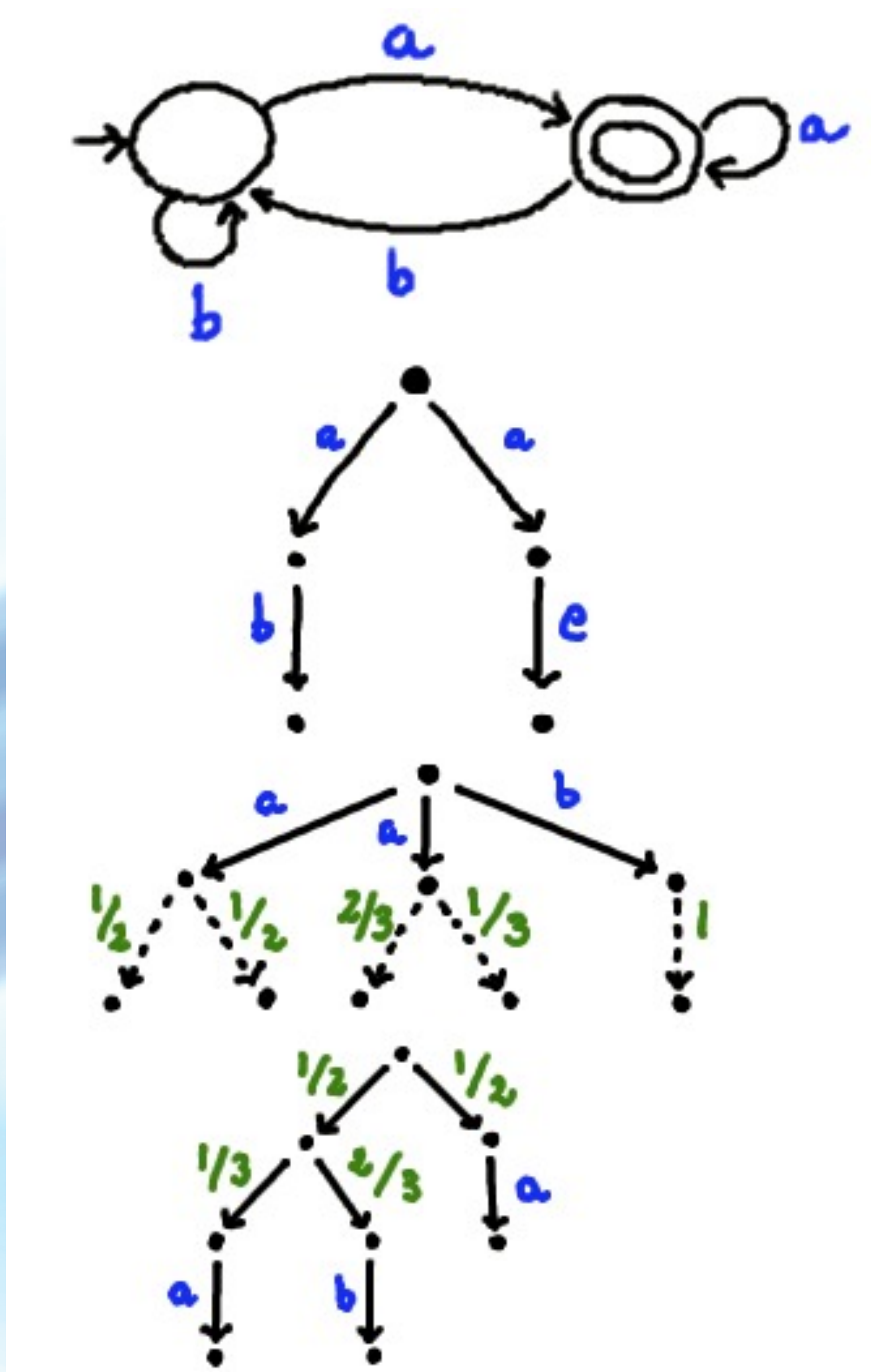


$(S, f : S \longrightarrow 2 \times S^A)$

$(S, f : S \longrightarrow P(S)^A)$

$(S, f : S \longrightarrow PD(S)^A)$

What do these things have in common?



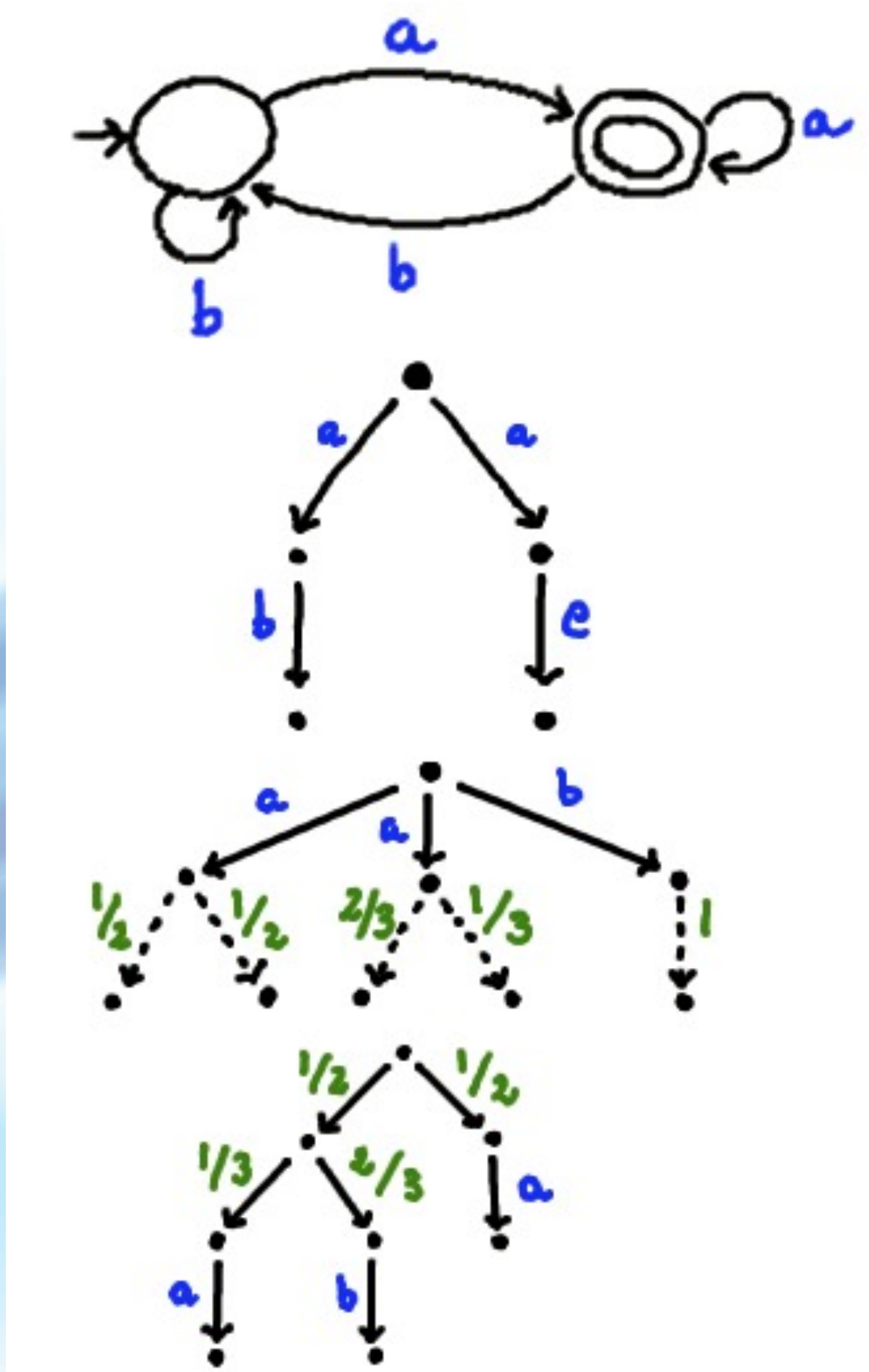
$$(S, f : S \longrightarrow 2 \times S^A)$$

$$(S, f : S \longrightarrow P(S)^A)$$

$$(S, f : S \longrightarrow PD(S)^A)$$

$$(S, f : S \longrightarrow 1 + A \times S + D(S))$$

What do these things have in common?



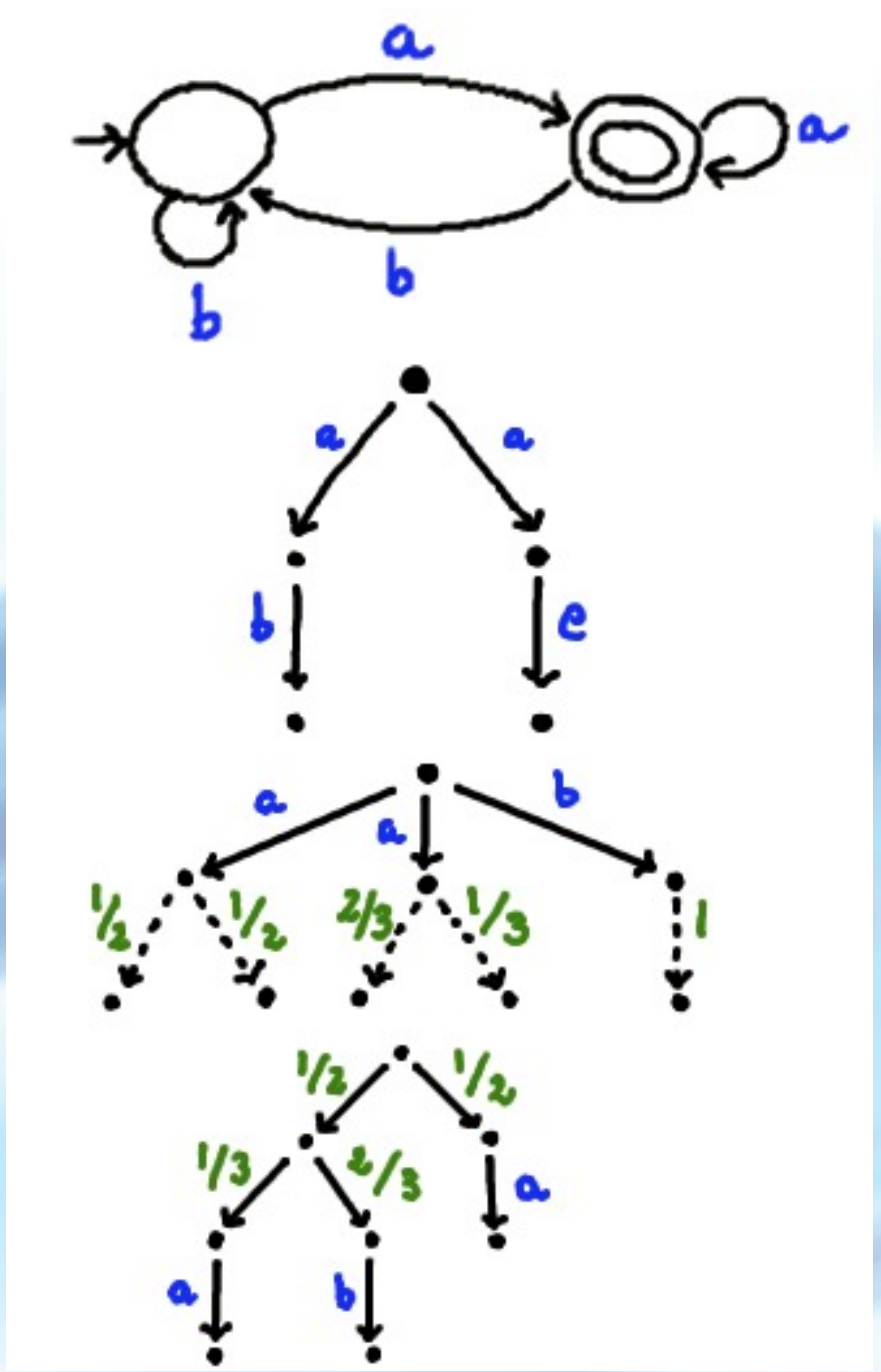
$$(S, f : S \longrightarrow 2 \times S^A)$$

$$(S, f : S \longrightarrow P(S)^A)$$

$$(S, f : S \longrightarrow PD(S)^A)$$

$$(S, f : S \longrightarrow 1 + A \times S + D(S))$$

What do these things have in common?



$(S, f : S \longrightarrow 2 \times S^A)$

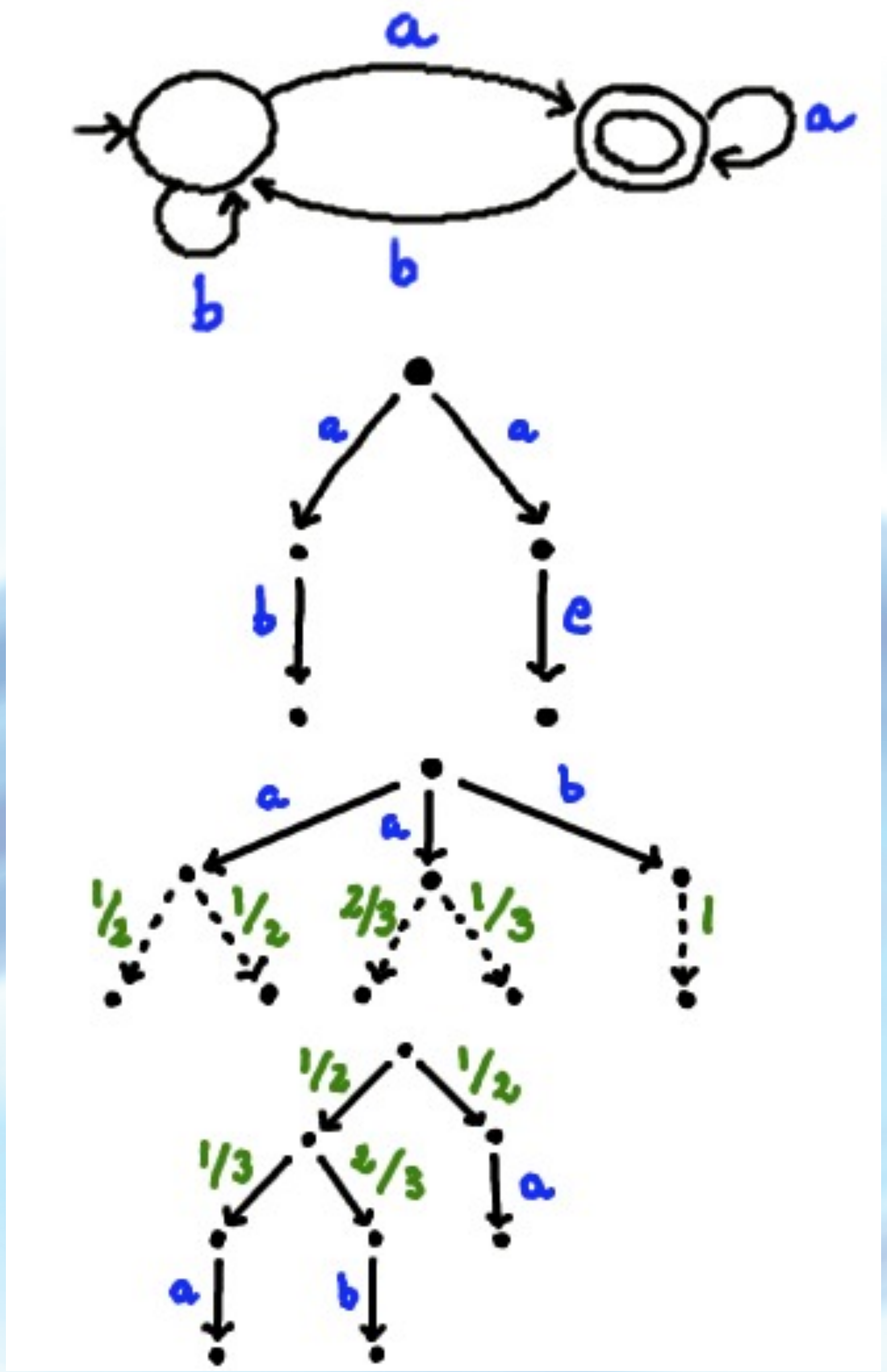
$(S, f : S \longrightarrow P(S)^A)$

$(S, f : S \longrightarrow PD(S)^A)$

$(S, f : S \longrightarrow 1 + A \times S + D(S))$

$(S, f : S \longrightarrow T(S))$

What do these things have in common?



$(S, f : S \longrightarrow 2 \times S^A)$

$(S, f : S \longrightarrow P(S)^A)$

$(S, f : S \longrightarrow PD(S)^A)$

$(S, f : S \longrightarrow 1 + A \times S + D(S))$

$(S, f : S \longrightarrow T(S))$ T- coalgebras

Research Questions

Research Questions

(1) Can we define a language (*syntax*) to denote behaviors and ...

Research Questions

(1) Can we define a language (*syntax*) to denote behaviors and ...

(2)... *algebraic laws* to reason about equivalence ...

Research Questions

(1) Can we define a language (*syntax*) to denote behaviors and ...

(2)... *algebraic laws* to reason about equivalence ...

(3)... for a large class of systems *uniformly?*

Research Questions

generalizing Kleene

(1) Can we define a language (*syntax*) to denote behaviors and ...

(2)... *algebraic laws* to reason about equivalence ...

(3)... for a large class of systems *uniformly?*

Research Questions

generalizing Kleene

generalizing Kozen

(1) Can we define a language (*syntax*) to denote behaviors and ...

(2)... *algebraic laws* to reason about equivalence ...

(3)... for a large class of systems *uniformly?*

Research Questions

generalizing Kleene

generalizing Kozen

(1) Can we define a language (*syntax*) to denote behaviors and ...

(2)... *algebraic laws* to reason about equivalence ...

(3)... for a large class of systems *uniformly?*

(4) Can coalgebraic methods help: is the type T of the system also enough to derive (1)-(3) above?

Research Questions

generalizing Kleene

generalizing Kozen

(1) Can we define a language (*syntax*) to denote behaviors and ...

(2)... *algebraic laws* to reason about equivalence ...

(3)... for a large class of systems *uniformly?*

(4) Can coalgebraic methods help: is the type T of the system also enough to derive (1)-(3) above?

YES WE CAN!

Coalgebraic methods

- Mathematical framework to reason about *state based systems*

Coalgebraic methods

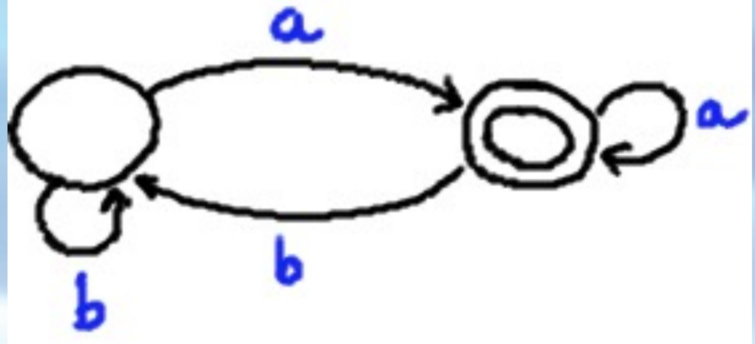
- Mathematical framework to reason about *state based systems*

Coalgebraic methods

- Mathematical framework to reason about *state based systems*
- Strenghts: the type of the system is enough to derive a canonical notion of *behaviour* and *equivalence*

languages

language equivalence



The power of T

$(S, f : S \qquad T(S))$

The power of T

$$(S, f : S \longrightarrow T(S))$$

The power of T

$$(S, f : S \longrightarrow T(S))$$

The power of T

$$(S, f : S \longrightarrow T(S))$$

The functor T determines:

The power of T

$$(S, f : S \longrightarrow T(S))$$

The functor T determines:

1) notion of observational equivalence (coalg. bisimulation)

The power of T

$$(S, f : S \longrightarrow T(S))$$

intuition: language equiv.

The functor T determines:

1) notion of observational equivalence (coalg. bisimulation)
2) behavior (final coalgebra)

intuition: languages

The power of T

$$(S, f : S \longrightarrow T(S))$$

The functor T determines:

1) notion of observational equivalence (coalg. bisimulation)
2) behavior (final coalgebra)
3) set of expressions describing finite systems

The power of T

$$(S, f : S \longrightarrow T(S))$$

The functor T determines:

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

The power of T

$$(S, f : S \longrightarrow T(S))$$

The functor T determines:

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

1 + 2 are classic coalgebra; 3 + 4 are my thesis

Coalgebras

Quantitative coalgebras

- Generalizations of deterministic automata
- Quantitative coalgebras: set of states S and $t : S \to TS$

$$T ::= Id \mid B \mid T \times T \mid T + T \mid T^A \mid \mathbb{M}^T$$

$\mathbb{M}$ is a monoid. $\mathcal{P} = 2^{Id}$ and $\mathcal{D}_\omega = \mathbb{R}^{Id}$

Coalgebras

Quantitative coalgebras

- Generalizations of deterministic automata
- Quantitative coalgebras: set of states S and $t : S \to TS$

$$T ::= Id \mid B \mid T \times T \mid T + T \mid T^A \mid \mathbb{M}^T$$

$\mathbb{M}$ is a monoid. $\mathcal{P} = 2^{Id}$ and $\mathcal{D}_\omega = \mathbb{R}^{Id}$

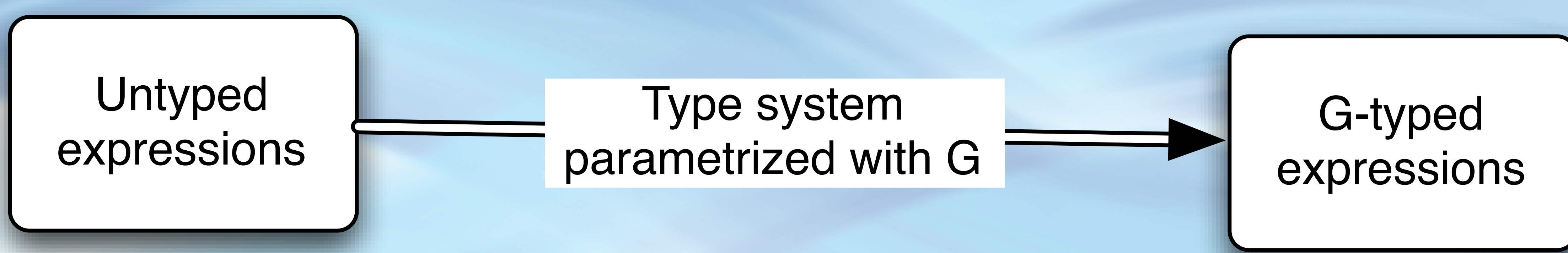
Examples

- $T = 2 \times Id^A$ — Deterministic automata
- $T = (B \times Id)^A$ — Mealy machines
- $T = (\mathcal{P} Id)^A$ — LTS
- $T = \mathcal{P}\mathcal{D}(S)^A$ — Simple Segala systems
- ...

T- expressions

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

$$E_T ::= ?$$



T- expressions

$$\mathit{Exp} \ni \varepsilon \quad ::= \quad \emptyset \mid \varepsilon \bigoplus \varepsilon \quad \mid \mu x.\gamma$$

T- expressions

$$Exp \ni \varepsilon \quad ::= \quad \emptyset \mid \varepsilon \oplus \varepsilon \quad \mid \mu x.\gamma$$

$$\mid b \qquad\qquad B$$

T- expressions

$$
\begin{aligned}
Exp \ni \varepsilon \quad ::= \quad & \emptyset \mid \varepsilon \oplus \varepsilon \quad \mid \mu x.\gamma \\
& \mid b \qquad\qquad\qquad\quad {\color{blue} B} \\
& \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle \quad {\color{blue} T_1 \times T_2}
\end{aligned}
$$

T- expressions

$$
\begin{array}{llll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon & \mid \mu x.\gamma \\
 & & & \mid b \qquad\qquad\qquad B \\
 & & & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle \qquad T_1 \times T_2 \\
 & & & \mid l[\varepsilon] \mid r[\varepsilon] \qquad T_1 + T_2
\end{array}
$$

T- expressions

$$
\begin{array}{lll}
Exp \ni \varepsilon & ::= & \emptyset \mid \varepsilon \oplus \varepsilon \quad \mid \mu x.\gamma & \\
& & \mid b & B \\
& & \mid l\langle\varepsilon\rangle \mid r\langle\varepsilon\rangle & T_1 \times T_2 \\
& & \mid l[\varepsilon] \mid r[\varepsilon] & T_1 + T_2 \\
& & \mid a(\varepsilon) & T^A \\
& & \mid m \cdot \varepsilon & \mathbb{M}^T
\end{array}
$$

Examples

LTS expressions – $T = (\mathcal{P} Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid$$

Examples

LTS expressions – $T = (\mathcal{P}Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid a(\underbrace{\qquad\quad}_{\_A})$$

Examples

LTS expressions – $T = (\mathcal{P}Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{a(\underbrace{1 \cdot \varepsilon}_{2^{Id}})}_{\_A} \mid a(\underbrace{0 \cdot \varepsilon}_{2^{Id}})$$

Examples

LTS expressions – $T = (\mathcal{P}Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{a(\underbrace{1 \cdot \varepsilon}_{2^{Id}})}_{\_^A} \mid a(\underbrace{0 \cdot \varepsilon}_{2^{Id}})$$

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

Examples

LTS expressions – $T = (\mathcal{P} Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{a(\underbrace{1 \cdot \varepsilon}_{2^{Id}})}_{\_^A} \mid a(\underbrace{0 \cdot \varepsilon}_{2^{Id}})$$

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

Markov Chain expressions – $T = \mathcal{D}_\omega(Id) = \mathbb{R}^{Id}$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{p \cdot \varepsilon}_{2^{Id}}$$

Examples

LTS expressions – $T = (\mathcal{P}Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{a(\underbrace{1 \cdot \varepsilon}_{2^{Id}})}_{\_^A} \mid a(\underbrace{0 \cdot \varepsilon}_{2^{Id}})$$

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

Markov Chain expressions – $T = \mathcal{D}_\omega(Id) = \mathbb{R}^{Id}$ $\quad \sum_i p_i = 1$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{p \cdot \varepsilon}_{2^{Id}}$$

Examples

LTS expressions – $T = (\mathcal{P}Id)^A = (2^{Id})^A$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{a(\underbrace{1 \cdot \varepsilon}_{2^{Id}})}_{\_A} \mid a(\underbrace{0 \cdot \varepsilon}_{2^{Id}})$$

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

Markov Chain expressions – $T = \mathcal{D}_\omega(Id) = \mathbb{R}^{Id}$ $\sum_i p_i = 1$

$$\varepsilon ::= \underbrace{\emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon}_{T} \mid \underbrace{p \cdot \varepsilon}_{2^{Id}}$$

$$\varepsilon ::= \mu x.\varepsilon \mid \bigoplus_{i \in 1 \dots n} p_i \cdot \varepsilon \quad \text{for } p_i \in (0,1] \text{ such that } \sum_{i \in 1 \dots n} p_i = 1$$

A generalized Kleene Theorem

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.

A generalized Kleene Theorem

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.

The goal is:

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

What does it mean **correspond**?

A generalized Kleene Theorem

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.

The goal is:

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

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

A generalized Kleene Theorem

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.

The goal is:

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

What does it mean **correspond**?

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

$\equiv$ **bisimilar**

intuition: language equiv.

intuition: languages

A generalized Kleene Theorem

Theorem

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

The proof provides algorithms to construct an expression from a system and vice-versa.

Axiomatization

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1, \quad \textit{T polynomial} \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} T$$

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1, \quad \textit{T polynomial} \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} T$$

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

Axiomatization

$$
\left.\begin{array}{lcl}
\varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\
\varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\
\varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1, \quad \textit{T polynomial} \\
\varepsilon \oplus \emptyset & \equiv & \varepsilon
\end{array}\right\} T
$$

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

$$
\left.\begin{array}{lcl}
\emptyset & \equiv & 0 \\
m_1 \cdot \varepsilon \oplus m_2 \cdot \varepsilon & \equiv & (m_1 + m_2) \cdot \varepsilon
\end{array}\right\} \mathbb{M}^T
$$

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1, \quad T \textit{ polynomial} \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} T$$

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

$$\left.\begin{array}{lcl} \emptyset & \equiv & 0 \\ m_1 \cdot \varepsilon \oplus m_2 \cdot \varepsilon & \equiv & (m_1 + m_2) \cdot \varepsilon \end{array}\right\} \mathbb{M}^T$$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} T_1 \times T_2$$

Similar for $T_1 + T_2$ and T^A

Axiomatization

$$\left.\begin{array}{lcl} \varepsilon_1 \oplus \varepsilon_2 & \equiv & \varepsilon_2 \oplus \varepsilon_1 \\ \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) & \equiv & (\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \\ \varepsilon_1 \oplus \varepsilon_1 & \equiv & \varepsilon_1, \quad \textit{T polynomial} \\ \varepsilon \oplus \emptyset & \equiv & \varepsilon \end{array}\right\} T$$

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

$$\left.\begin{array}{lcl} \emptyset & \equiv & 0 \\ m_1 \cdot \varepsilon \oplus m_2 \cdot \varepsilon & \equiv & (m_1 + m_2) \cdot \varepsilon \end{array}\right\} \mathbb{M}^T$$

$$\left.\begin{array}{lcl} l(\emptyset) & \equiv & \emptyset \\ l(\varepsilon_1) \oplus l(\varepsilon_2) & \equiv & l(\varepsilon_1 \oplus \varepsilon_2) \\ r(\emptyset) & \equiv & \emptyset \\ r(\varepsilon_1) \oplus r(\varepsilon_2) & \equiv & r(\varepsilon_1 \oplus \varepsilon_2) \end{array}\right\} T_1 \times T_2$$

Similar for $T_1 + T_2$ and T^A

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

Results I: Stratified systems

$$\varepsilon ::= \mu x.\varepsilon \mid x \mid \langle b, \varepsilon \rangle \mid \bigoplus_{i \in 1 \ldots n} p_i \cdot \varepsilon_i \mid \downarrow$$

where $b \in B$, $p_i \in (0, 1]$ and $\sum_{i \in 1 \ldots n} p_i = 1$

$$(\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \equiv \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3)$$
$$\varepsilon_1 \oplus \varepsilon_2 \equiv \varepsilon_2 \oplus \varepsilon_1$$
$$(p_1 \cdot \varepsilon) \oplus (p_2 \cdot \varepsilon) \equiv (p_1 + p_2) \cdot \varepsilon$$
$$\varepsilon[\mu x.\varepsilon / x] \equiv \mu x.\varepsilon$$
$$\gamma[\varepsilon / x] \equiv \varepsilon \Rightarrow \mu x.\gamma \equiv \varepsilon$$

Same syntax as in [van Glabbeek, Smolka and Steffen'95] and new axiomatization (inexistent).

Results II: Segala systems

$$\varepsilon ::= \emptyset \mid \varepsilon \boxplus \varepsilon \mid \mu x.\varepsilon \mid x \mid a(\{\varepsilon'\})$$
$$\varepsilon' ::= \bigoplus_{i \in 1 \cdots n} p_i \cdot \varepsilon_i$$

where $a \in A$, $p_i \in (0,1]$ and $\sum_{i \in 1 \ldots n} p_i = 1$

$$(\varepsilon_1 \boxplus \varepsilon_2) \boxplus \varepsilon_3 \equiv \varepsilon_1 \boxplus (\varepsilon_2 \boxplus \varepsilon_3)$$
$$\varepsilon_1 \boxplus \varepsilon_2 \equiv \varepsilon_2 \boxplus \varepsilon_1$$
$$\varepsilon \boxplus \emptyset \equiv \varepsilon$$
$$\varepsilon \boxplus \varepsilon \equiv \varepsilon$$
$$(\varepsilon'_1 \oplus \varepsilon'_2) \oplus \varepsilon'_3 \equiv \varepsilon'_1 \oplus (\varepsilon'_2 \oplus \varepsilon'_3)$$
$$\varepsilon'_1 \oplus \varepsilon'_2 \equiv \varepsilon'_2 \oplus \varepsilon'_1$$
$$(p_1 \cdot \varepsilon) \oplus (p_2 \cdot \varepsilon) \equiv (p_1 + p_2) \cdot \varepsilon$$
$$\varepsilon[\mu x.\varepsilon / x] \equiv \mu x.\varepsilon$$
$$\gamma[\varepsilon / x] \equiv \varepsilon \Rightarrow \mu x.\gamma \equiv \varepsilon$$

Same syntax and axioms as in [Deng and Palamidessi'05]

Results III: Pnueli-Zuck systems

$$\varepsilon ::= \emptyset \mid \varepsilon \boxplus \varepsilon \mid \mu x.\varepsilon \mid x \mid \{\varepsilon'\}$$
$$\varepsilon' ::= \bigoplus_{i \in 1 \cdots n} p_i \cdot \varepsilon''_i$$
$$\varepsilon'' ::= \emptyset \mid \varepsilon'' \boxplus \varepsilon'' \mid a(\{\varepsilon\})$$

where $a \in A$, $p_i \in (0, 1]$ and $\sum_{i \in 1 \ldots n} p_i = 1$

$$(\varepsilon_1 \boxplus \varepsilon_2) \boxplus \varepsilon_3 \equiv \varepsilon_1 \boxplus (\varepsilon_2 \boxplus \varepsilon_3)$$
$$\varepsilon_1 \boxplus \varepsilon_2 \equiv \varepsilon_2 \boxplus \varepsilon_1$$
$$\varepsilon \boxplus \emptyset \equiv \varepsilon$$
$$\varepsilon \boxplus \varepsilon \equiv \varepsilon$$
$$(\varepsilon'_1 \oplus \varepsilon'_2) \oplus \varepsilon'_3 \equiv \varepsilon'_1 \oplus (\varepsilon'_2 \oplus \varepsilon'_3) \qquad \varepsilon'_1 \oplus \varepsilon'_2 \equiv \varepsilon'_2 \oplus \varepsilon'_1$$
$$(p_1 \cdot \varepsilon'') \oplus (p_2 \cdot \varepsilon'') \equiv (p_1 + p_2) \cdot \varepsilon''$$
$$\varepsilon[\mu x.\varepsilon / x] \equiv \mu x.\varepsilon$$
$$\gamma[\varepsilon / x] \equiv \varepsilon \Rightarrow \mu x.\gamma \equiv \varepsilon$$

New syntax and axiomatization.

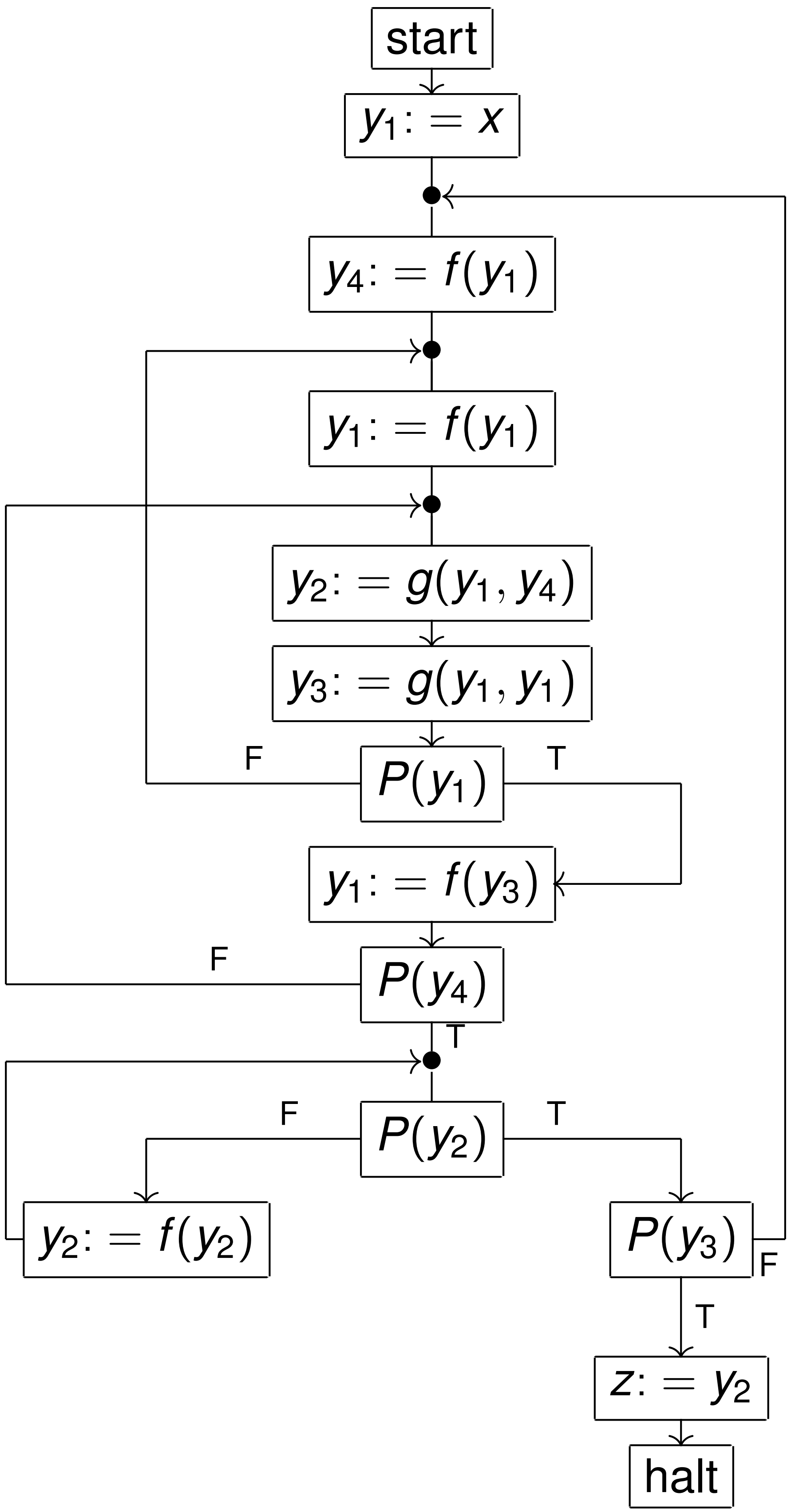
Conclusions

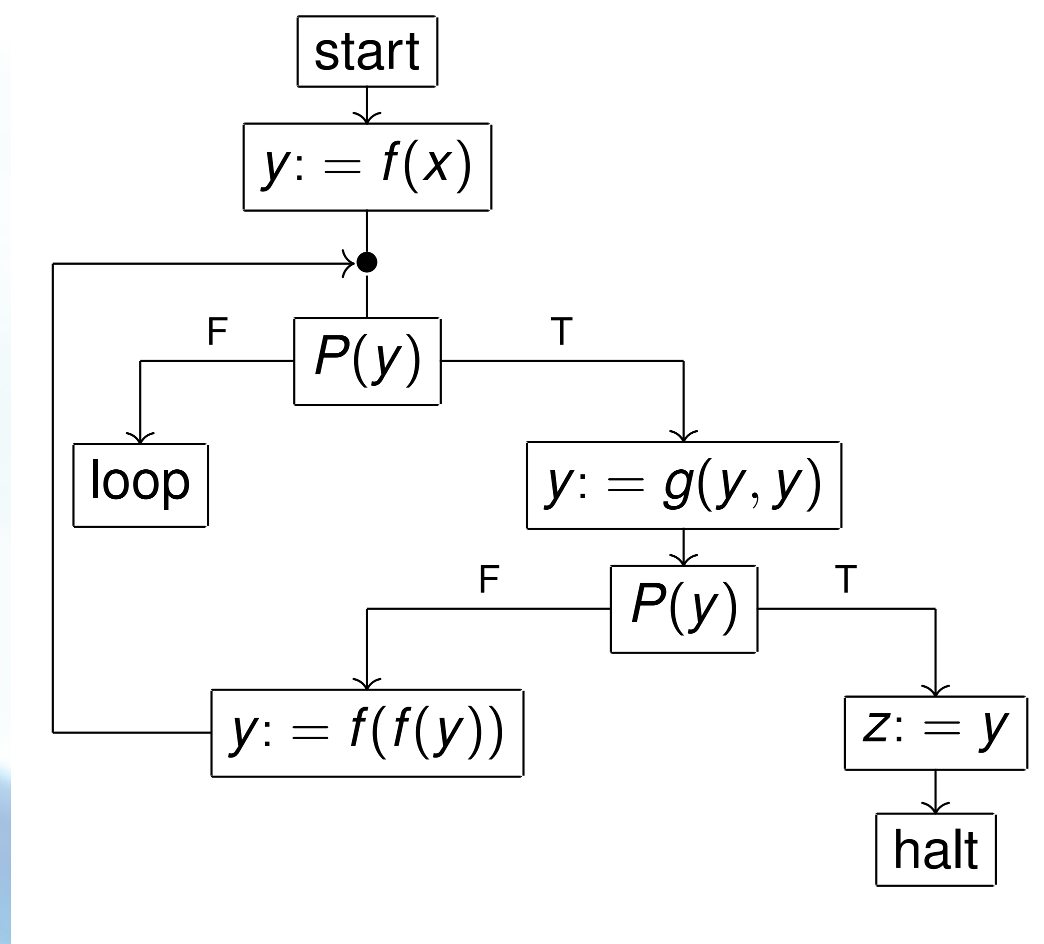
- Framework to uniformly derive language and axioms for quantitative coalgebras (weighted automata, probabilistic automata, etc)
- Examples show the effectiveness of the framework: known syntaxes recovered, new ones derived.

Future work

- Extend the syntax with new operators (paralell composition, etc)
- Coalgebraic context-free counterpart
- Automation: `Circ`

Why should we care about coalgebra?





Original proof: complex graph transformation
Algebraic proof: beautiful, but long and requires (Kozen) ingenuity

Coinductive proof fully automatic
(uses Kozen's coinductive KAT)