Probabilistic systems

a place where categories meet probability

Ana Sokolova

SOS group, Radboud University Nijmegen

Outline

- Introduction - probabilistic systems and coalgebras

Outline

- Introduction - probabilistic systems and coalgebras
- Bisimilarity - the strong end of the spectrum

Outline

- Introduction - probabilistic systems and coalgebras
- Bisimilarity - the strong end of the spectrum
- Application - expressiveness hierarchy

  (older result)

Outline

- Introduction - probabilistic systems and coalgebras
- Bisimilarity - the strong end of the spectrum
- Application - expressiveness hierarchy

  (older result)
- Trace semantics - the weak end of the spectrum

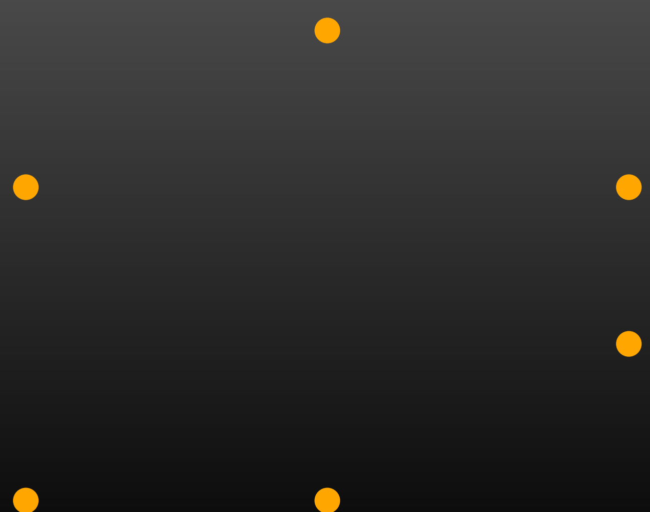
  (newer result)

Systems

are formal objects, transition systems (e.g. LTS), that serve as models of real (software, hardware,...) systems

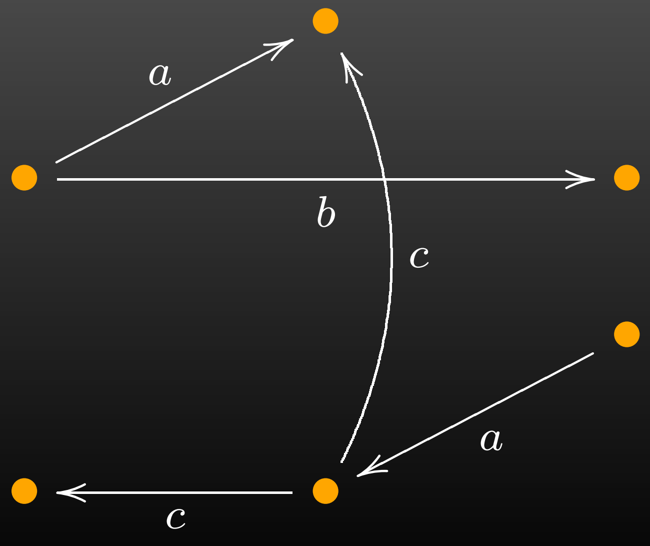
Systems

are formal objects, transition systems (e.g. LTS), that serve as models of real (software, hardware,...) systems



Systems

are formal objects, transition systems (e.g. LTS), that serve as models of real (software, hardware,...) systems



Probabilistic systems

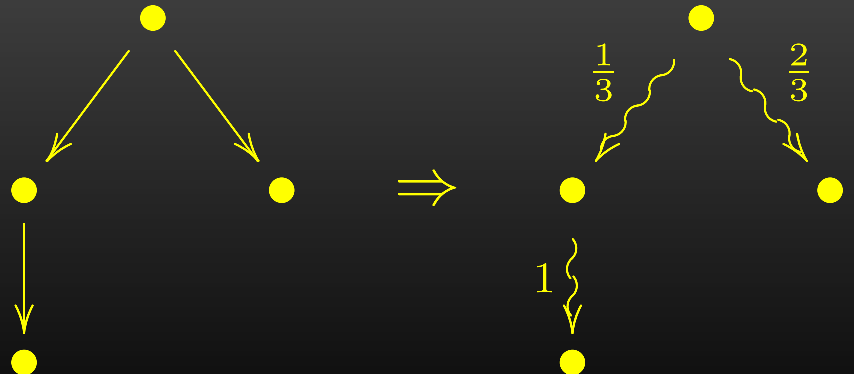
arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

Examples:

Probabilistic systems

arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

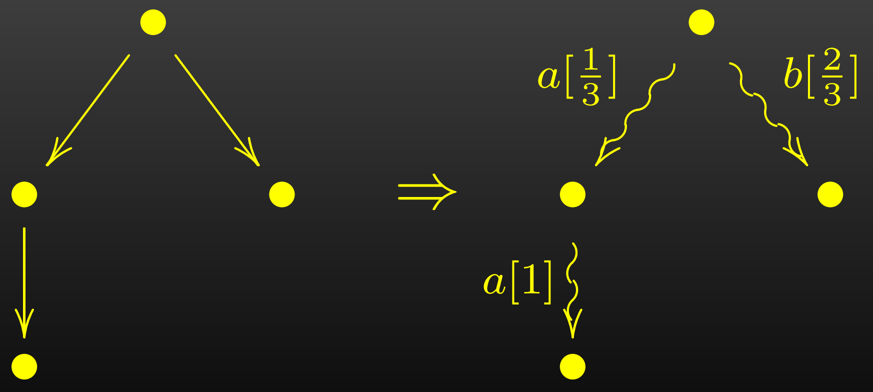
Examples:



Probabilistic systems

arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

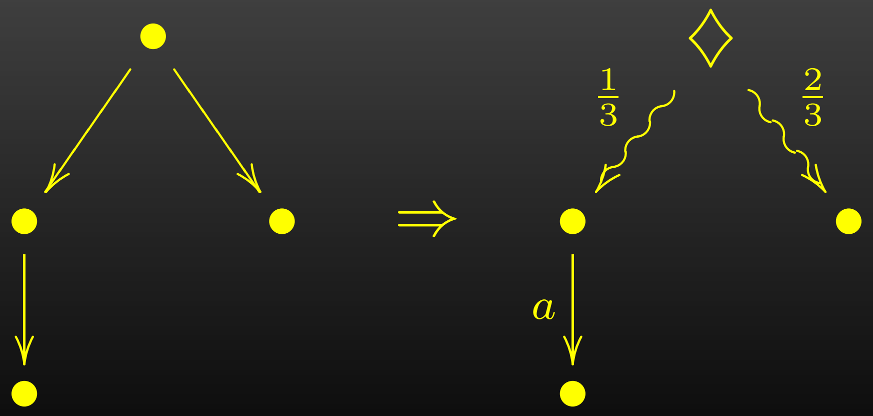
Examples:



Probabilistic systems

arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

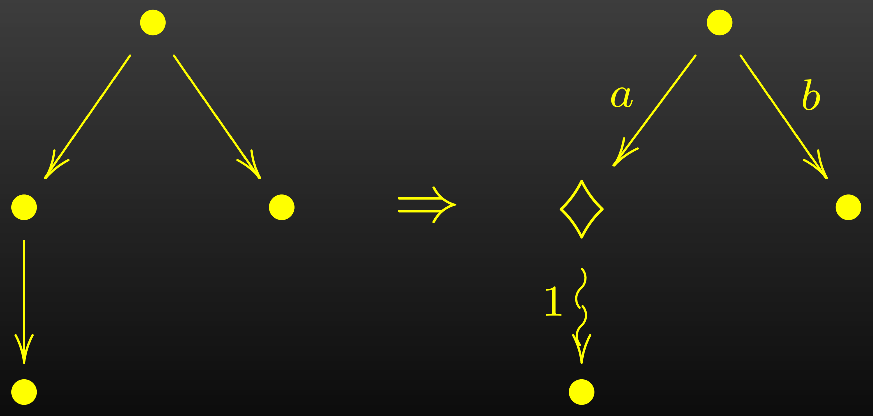
Examples:



Probabilistic systems

arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

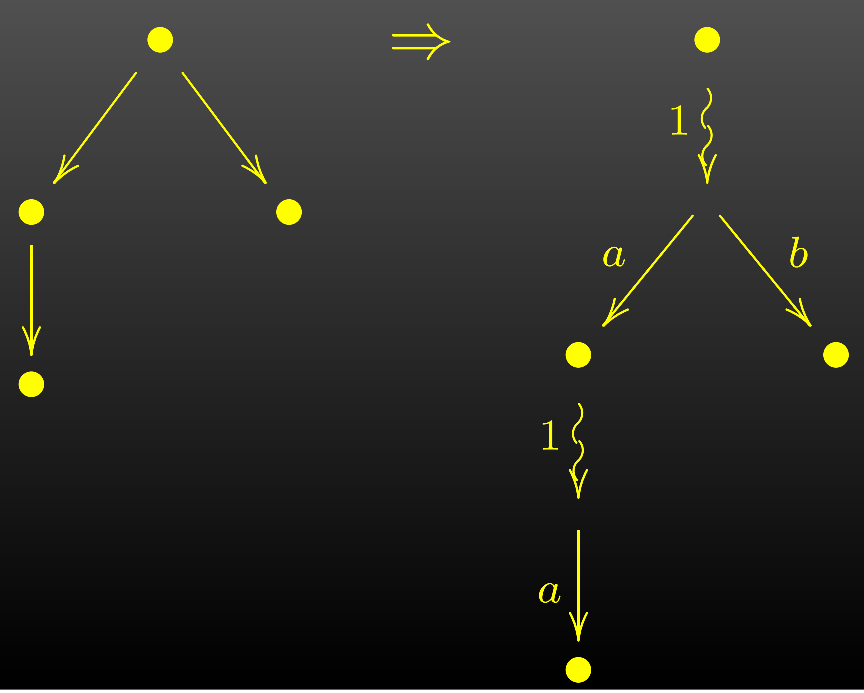
Examples:



Probabilistic systems

arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

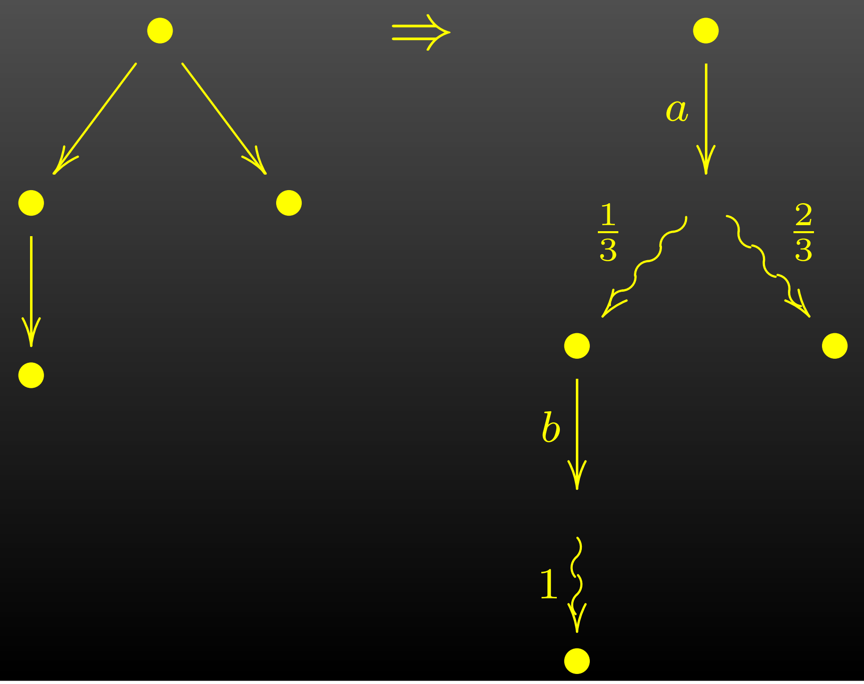
Examples:



Probabilistic systems

arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

Examples:



Coalgebras

are an elegant generalization of transition systems with

states + transitions

Coalgebras

are an elegant generalization of transition systems with

states + transitions

as pairs

$\langle S, \alpha : S \to \mathcal{F}S \rangle$, for $\mathcal{F}$ a functor

Coalgebras

are an elegant generalization of transition systems with

states + transitions

as pairs

$$\langle S, \alpha : S \to \mathcal{F}S \rangle, \text{ for } \mathcal{F} \text{ a functor}$$

- based on category theory
- provide a uniform way of treating transition systems
- provide general notions and results e.g. a generic notion of bisimulation

Examples

A TS is a pair $\langle S, \alpha : S \to \mathcal{P}S \rangle$

!! coalgebra of the powerset functor $\mathcal{P}$

Examples

A TS is a pair $\langle S, \alpha : S \to \mathcal{P}S \rangle$

!! coalgebra of the powerset functor $\mathcal{P}$

An LTS is a pair $\langle S, \alpha : S \to \mathcal{P}S^A \rangle$

!!! coalgebra of the functor $\mathcal{P}^A$

Note: $\mathcal{P}^A \cong \mathcal{P}(A \times \_\,)$

More examples

Thanks to the probability distribution functor $\mathcal{D}$

$$\mathcal{D}S = \{\mu : S \to [0,1], \mu[S] = 1\}, \quad \mu[X] = \sum_{s \in X} \mu(x)$$

$$\mathcal{D}f : \mathcal{D}S \to \mathcal{D}T, \quad \mathcal{D}f(\mu)(t) = \mu[f^{-1}(\{t\})]$$

the probabilistic systems are also coalgebras

More examples

Thanks to the probability distribution functor $\mathcal{D}$

$$\mathcal{D}S = \{\mu : S \to [0,1], \mu[S] = 1\}, \quad \mu[X] = \sum_{s \in X} \mu(x)$$

$$\mathcal{D}f : \mathcal{D}S \to \mathcal{D}T, \;\; \mathcal{D}f(\mu)(t) = \mu[f^{-1}(\{t\})]$$

the probabilistic systems are also coalgebras ... of functors built by the following syntax

$$\mathcal{F} ::= \_ \mid A \mid \mathcal{P} \mid \mathcal{D} \mid \mathcal{G} + \mathcal{H} \mid \mathcal{G} \times \mathcal{H} \mid \mathcal{G}^A \mid \mathcal{G} \circ \mathcal{H}$$

reactive, generative

evolve from LTS - functor $\mathcal{P}(A \times \_) \cong \mathcal{P}^A$

reactive, generative

evolve from LTS - functor $\mathcal{P}(A \times \_) \cong \mathcal{P}^A$

reactive systems:

functor $(\mathcal{D} + 1)^A$

reactive, generative

evolve from LTS - functor $\mathcal{P}(A \times \_) \cong \mathcal{P}^A$

reactive systems:

functor $(\mathcal{D} + 1)^A$

generative systems:

functor $(\mathcal{D} + 1)(A \times \_) = \mathcal{D}(A \times \_) + 1$

reactive, generative

evolve from LTS - functor $\mathcal{P}(A \times \_) \cong \mathcal{P}^A$

reactive systems:

functor $(\mathcal{D} + 1)^A$

generative systems:

functor $(\mathcal{D} + 1)(A \times \_) = \mathcal{D}(A \times \_) + 1$

note: in the probabilistic case

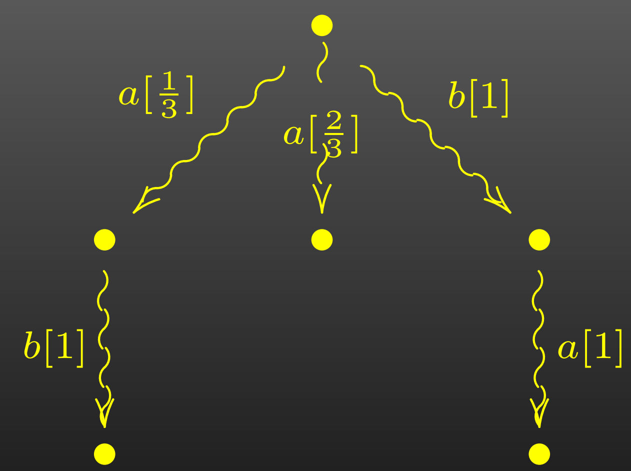
$$(\mathcal{D} + 1)^A \not\cong \mathcal{D}(A \times \_) + 1$$

Probabilistic system types

MC	$\mathcal{D}$
DLTS	$(\_ + 1)^A$
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$
React	$(\mathcal{D} + 1)^A$
Gen	$\mathcal{D}(A \times \_) + 1$
Str	$\mathcal{D} + (A \times \_) + 1$
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$
SSeg	$\mathcal{P}(A \times \mathcal{D})$
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$
$\dots$	$\dots$

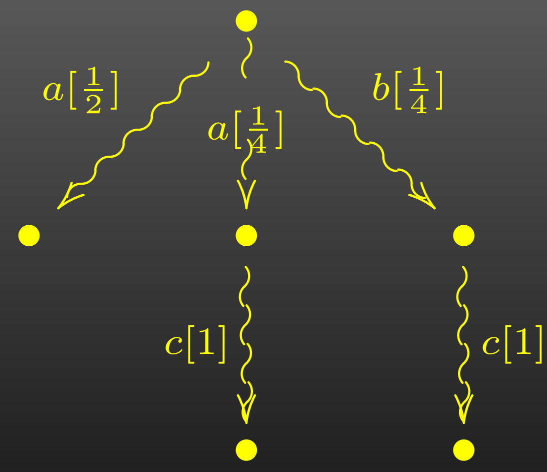
Probabilistic system types

MC	$\mathcal{D}$
DLTS	$(\_ + 1)^A$
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$
React	$(\mathcal{D} + 1)^A$
Gen	$\mathcal{D}(A \times \_) + 1$
Str	$\mathcal{D} + (A \times \_) + 1$
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$
SSeg	$\mathcal{P}(A \times \mathcal{D})$
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



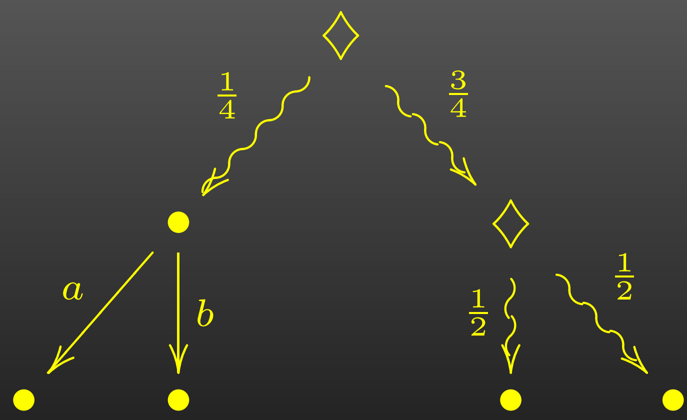
Probabilistic system types

MC	$\mathcal{D}$
DLTS	$(\_ + 1)^A$
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$
React	$(\mathcal{D} + 1)^A$
Gen	$\mathcal{D}(A \times \_) + 1$
Str	$\mathcal{D} + (A \times \_) + 1$
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$
SSeg	$\mathcal{P}(A \times \mathcal{D})$
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



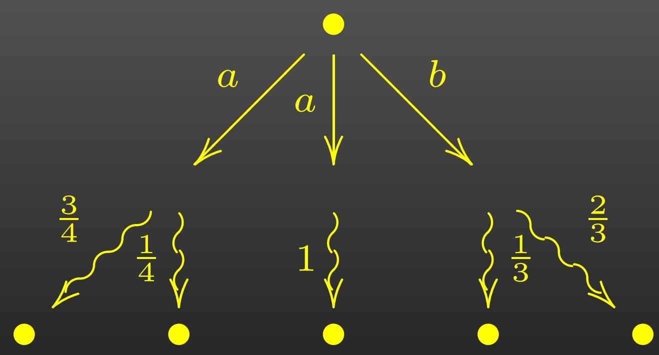
Probabilistic system types

MC	$\mathcal{D}$
DLTS	$(\_ + 1)^A$
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$
React	$(\mathcal{D} + 1)^A$
Gen	$\mathcal{D}(A \times \_) + 1$
Str	$\mathcal{D} + (A \times \_) + 1$
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$
SSeg	$\mathcal{P}(A \times \mathcal{D})$
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



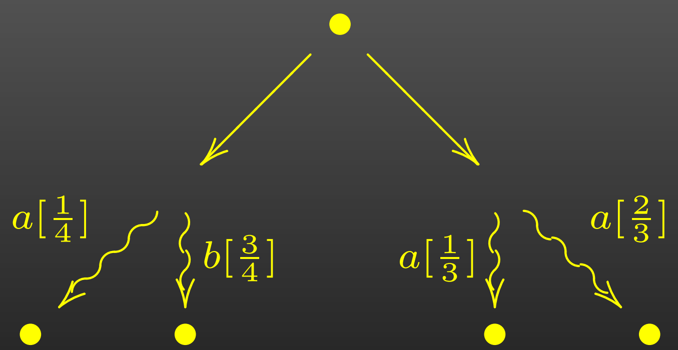
Probabilistic system types

MC	$\mathcal{D}$
DLTS	$(\_ + 1)^A$
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$
React	$(\mathcal{D} + 1)^A$
Gen	$\mathcal{D}(A \times \_) + 1$
Str	$\mathcal{D} + (A \times \_) + 1$
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$
SSeg	$\mathcal{P}(A \times \mathcal{D})$
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



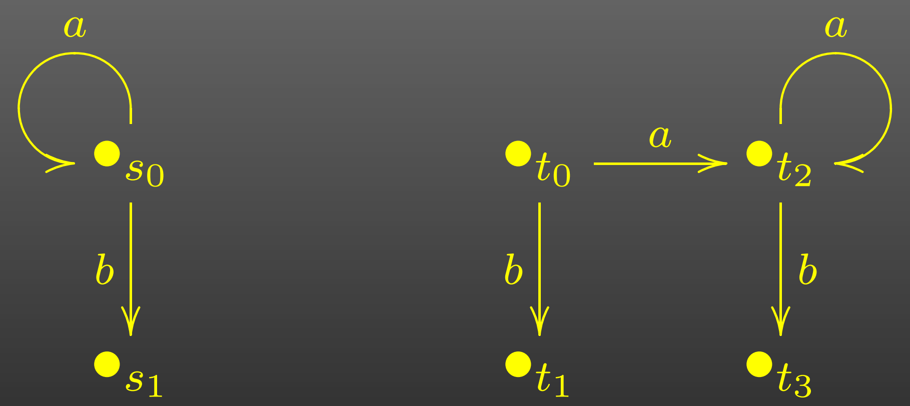
Probabilistic system types

MC	$\mathcal{D}$
DLTS	$(\_ + 1)^A$
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$
React	$(\mathcal{D} + 1)^A$
Gen	$\mathcal{D}(A \times \_) + 1$
Str	$\mathcal{D} + (A \times \_) + 1$
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$
SSeg	$\mathcal{P}(A \times \mathcal{D})$
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



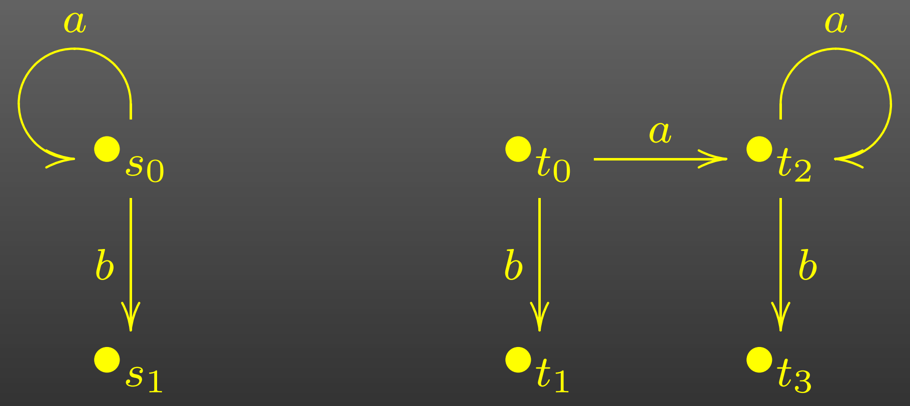
Bisimulation - LTS

Consider the LTS



Bisimulation - LTS

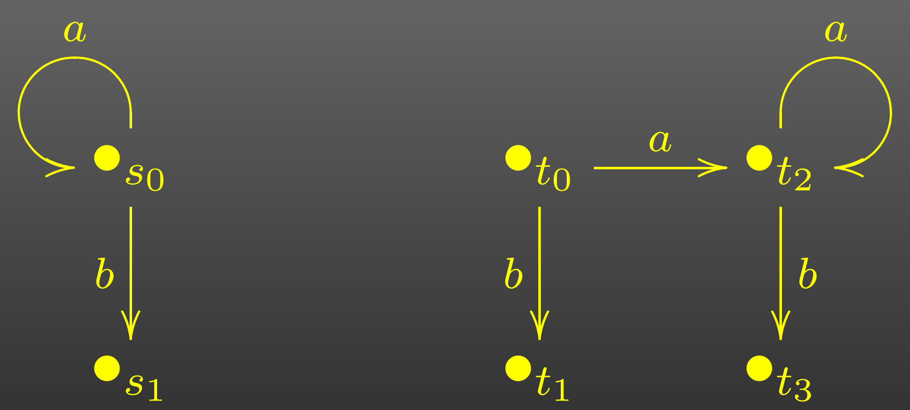
Consider the LTS



The states s_0 and t_0 are bisimilar since there is a bisimulation R relating them...

Bisimulation - LTS

Consider the LTS



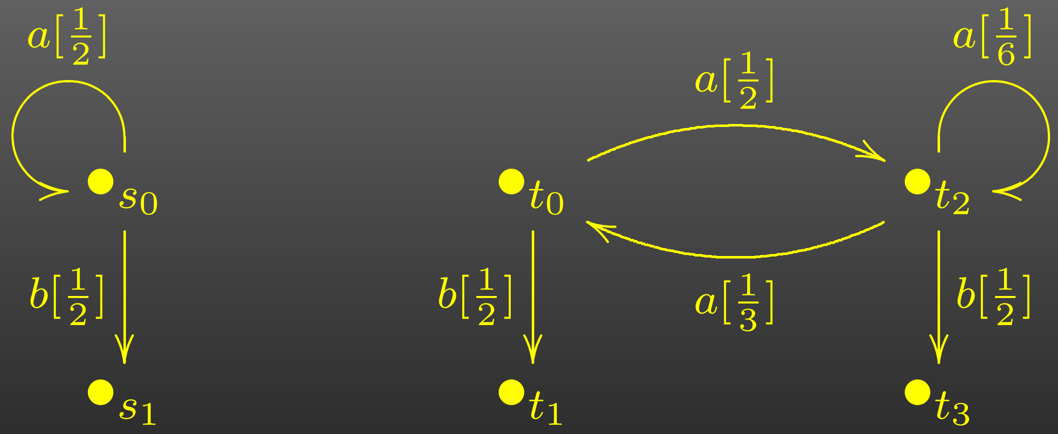
Transfer condition: $\langle s, t \rangle \in R \implies$

$$s \xrightarrow{a} s' \Rightarrow (\exists t')\ t \xrightarrow{a} t',\ \langle s', t' \rangle \in R,$$

$$t \xrightarrow{a} t' \Rightarrow (\exists s')\ s \xrightarrow{a} s',\ \langle s', t' \rangle \in R$$

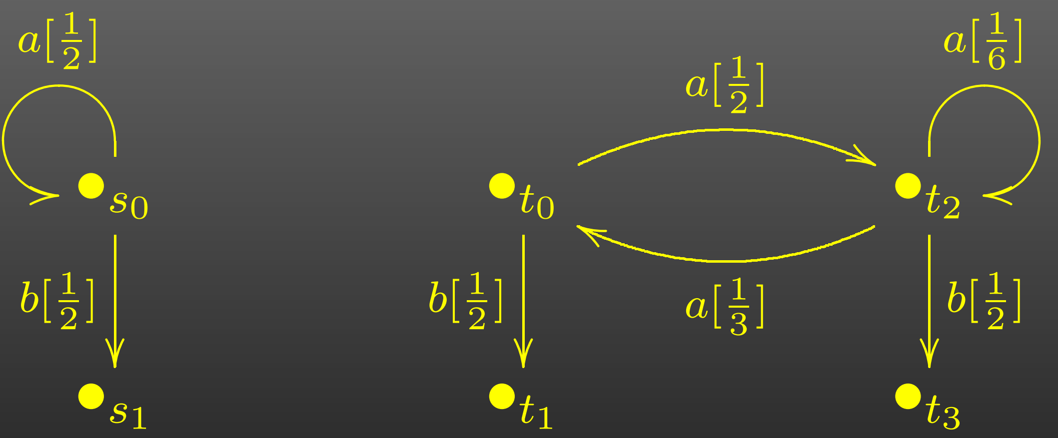
Bisimulation - generative

Consider the generative systems



Bisimulation - generative

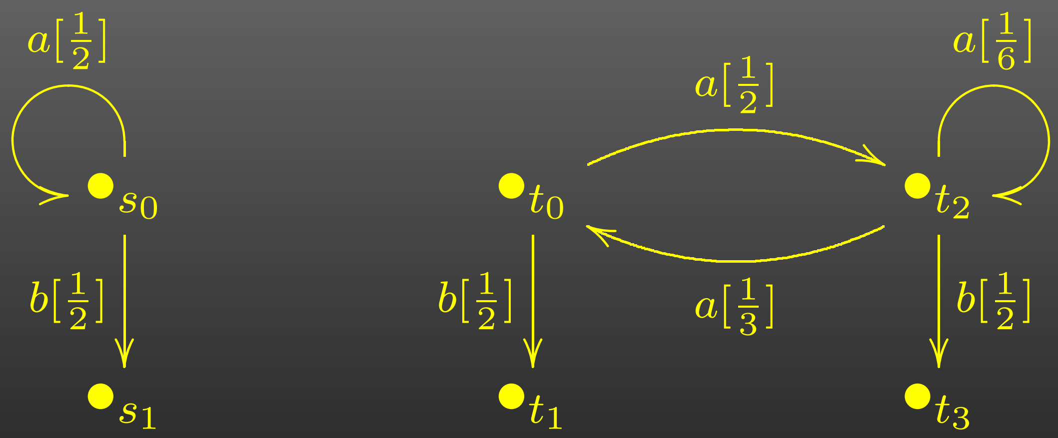
Consider the generative systems



The states s_0 and t_0 are bisimilar, and so are s_0 and t_2, since there is a bisimulation R relating them...

Bisimulation - generative

Consider the generative systems

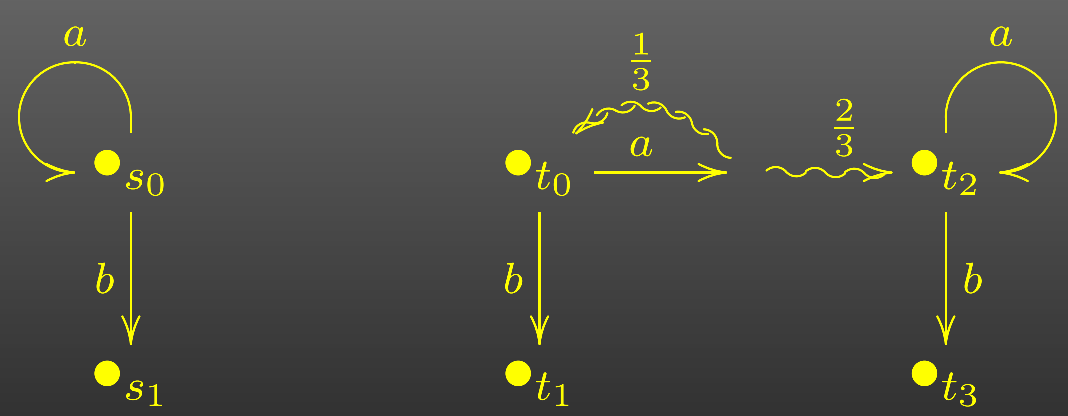


Transfer condition: $\langle s, t \rangle \in R \quad \Longrightarrow$

$$s \rightsquigarrow \mu \Rightarrow (\exists \mu')\ t \rightsquigarrow \mu',\ \mu \equiv_{R,A} \mu'$$

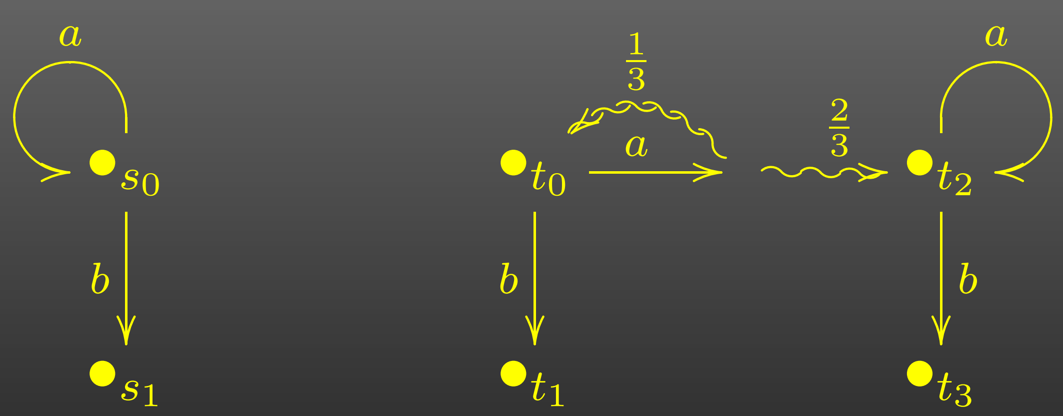
Bisimulation - simple Segala

Consider the simple Segala systems



Bisimulation - simple Segala

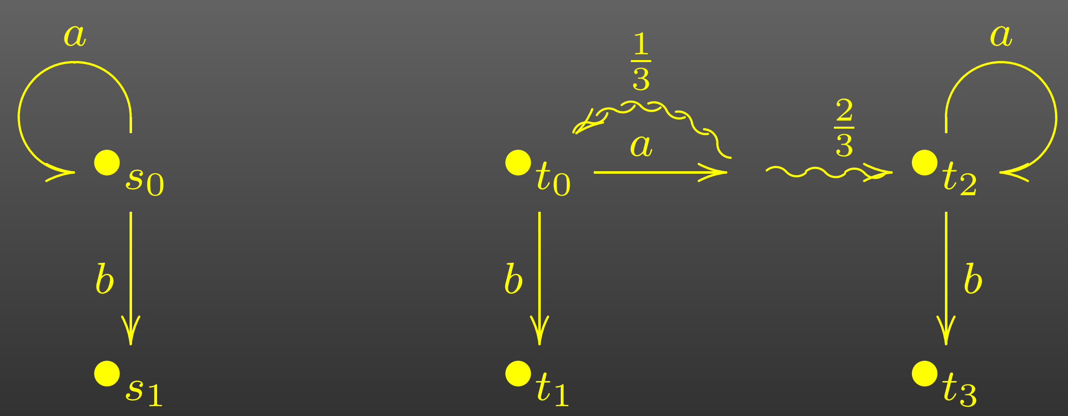
Consider the simple Segala systems



The states s_0 and t_0 are bisimilar, since there is a bisimulation R relating them...

Bisimulation - simple Segala

Consider the simple Segala systems



Transfer condition: $\langle s, t\rangle \in R \quad \Longrightarrow$

$$s \xrightarrow{a} \mu \Rightarrow (\exists \mu')\ t \xrightarrow{a} \mu',\ \ \mu \equiv_R \mu'$$

Coalgebraic bisimulation

A bisimulation between

$$\langle S, \alpha : S \to \mathcal{F}S \rangle \text{ and } \langle T, \beta : S \to \mathcal{F}S \rangle$$

is $R \subseteq S \times T$ such that $\exists\, \gamma$:

Coalgebraic bisimulation

A bisimulation between

$$\langle S, \alpha : S \to \mathcal{F}S \rangle \text{ and } \langle T, \beta : S \to \mathcal{F}S \rangle$$

is $R \subseteq S \times T$ such that $\exists\, \gamma$:

$$\begin{array}{ccccc}
S & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & T \\
\downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \gamma} & & \downarrow{\scriptstyle \beta} \\
\mathcal{F}S & \xleftarrow[\mathcal{F}\pi_1]{} & \mathcal{F}R & \xrightarrow[\mathcal{F}\pi_2]{} & \mathcal{F}T
\end{array}$$

Coalgebraic bisimulation

A bisimulation between

$$\langle S, \alpha : S \to \mathcal{F}S \rangle \text{ and } \langle T, \beta : S \to \mathcal{F}S \rangle$$

is $R \subseteq S \times T$ such that $\exists\, \gamma$:

$$\begin{array}{ccccc}
S & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & T \\
\downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \gamma} & & \downarrow{\scriptstyle \beta} \\
\mathcal{F}S & \xleftarrow[\mathcal{F}\pi_1]{} & \mathcal{F}R & \xrightarrow[\mathcal{F}\pi_2]{} & \mathcal{F}T
\end{array}$$

Transfer condition: $\langle s, t \rangle \in R \quad \Longrightarrow$

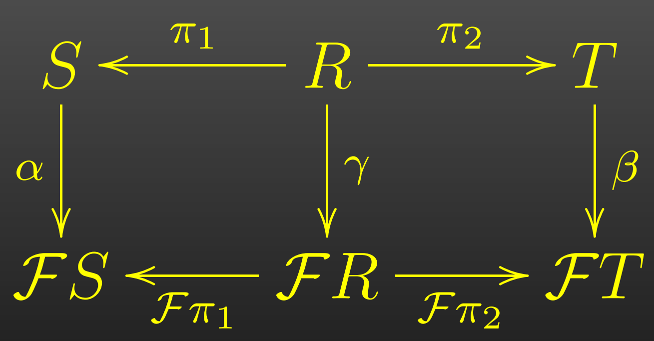
$$\langle \alpha(s), \beta(t) \rangle \in \mathrm{Rel}(\mathcal{F})(R)$$

Coalgebraic bisimulation

A bisimulation between

$$\langle S, \alpha : S \to \mathcal{F}S \rangle \text{ and } \langle T, \beta : S \to \mathcal{F}S \rangle$$

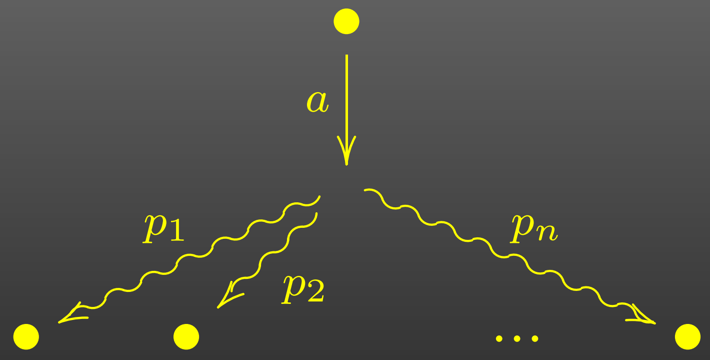
is $R \subseteq S \times T$ such that $\exists\, \gamma$:



Theorem: Coalgebraic and concrete bisimilarity coincide !

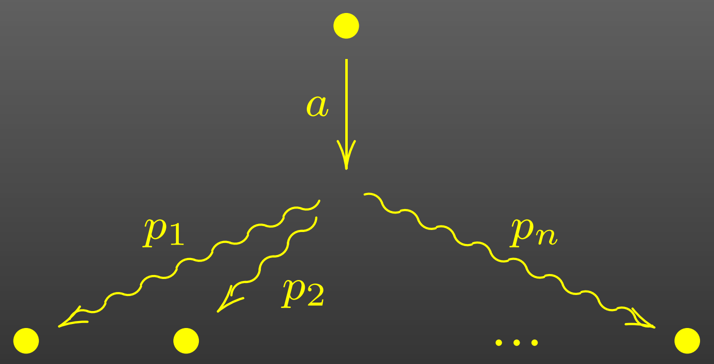
Expressiveness

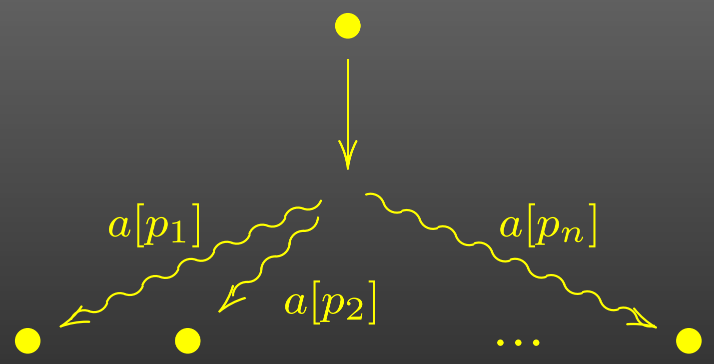
simple Segala system $\longrightarrow$ Segala system



Expressiveness

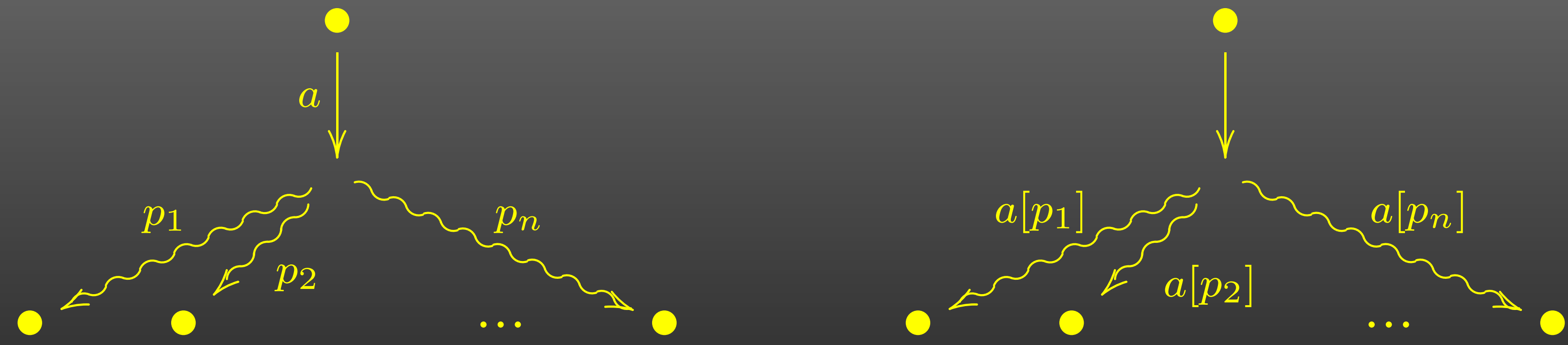
simple Segala system $\longrightarrow$ Segala system





Expressiveness

simple Segala system $\longrightarrow$ Segala system



When do we consider one type of systems more expressive than another?

Comparison criterion

$\mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$

if there is a mapping $\langle S, \alpha : S \to \mathcal{F}S \rangle \overset{\mathcal{T}}{\mapsto} \langle S, \tilde{\alpha} : S \to \mathcal{G}S \rangle$ that preserves and reflects bisimilarity

Comparison criterion

$\mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$

if there is a mapping $\langle S, \alpha : S \to \mathcal{F}S \rangle \overset{\mathcal{T}}{\mapsto} \langle S, \tilde{\alpha} : S \to \mathcal{G}S \rangle$ that preserves and reflects bisimilarity

$$s_{\langle S,\alpha \rangle} \sim t_{\langle T,\beta \rangle} \iff s_{\mathcal{T}\langle S,\alpha \rangle} \sim t_{\mathcal{T}\langle T,\beta \rangle}$$

Comparison criterion

$\mathsf{Coalg}_{\mathcal{F}} \longrightarrow \mathsf{Coalg}_{\mathcal{G}}$

if there is a mapping $\langle S, \alpha : S \to \mathcal{F}S\rangle \overset{\mathcal{T}}{\mapsto} \langle S, \tilde{\alpha} : S \to \mathcal{G}S\rangle$ that preserves and reflects bisimilarity

Theorem: An injective natural transformation $\mathcal{F} \Rightarrow \mathcal{G}$ is sufficient for $\mathsf{Coalg}_{\mathcal{F}} \longrightarrow \mathsf{Coalg}_{\mathcal{G}}$

Comparison criterion

$\mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$

if there is a mapping $\langle S, \alpha : S \to \mathcal{F}S \rangle \overset{\mathcal{T}}{\mapsto} \langle S, \tilde{\alpha} : S \to \mathcal{G}S \rangle$ that preserves and reflects bisimilarity

Theorem: An injective natural transformation $\mathcal{F} \Rightarrow \mathcal{G}$ is sufficient for $\mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$

proof via cocongruences - behavioral equivalence

Example

Indeed $\mathbf{SSeg} \to \mathbf{Seg}$ since $\mathcal{P}(A \times \mathcal{D}) \overset{\mathcal{P}\tau}{\Rightarrow} \mathcal{P}\mathcal{D}(A \times \_)$ is injective for

Example

Indeed $\mathbf{SSeg} \to \mathbf{Seg}$ since $\mathcal{P}(A \times \mathcal{D}) \overset{\mathcal{P}\mathcal{T}}{\Rightarrow} \mathcal{P}\mathcal{D}(A \times \_\,)$ is injective for

$$(A \times \mathcal{D}) \overset{\mathcal{T}}{\Rightarrow} \mathcal{D}(A \times \_\,)$$

given by

Example

Indeed $\mathbf{SSeg} \to \mathbf{Seg}$ since $\mathcal{P}(A \times \mathcal{D}) \stackrel{\mathcal{P}\tau}{\Rightarrow} \mathcal{P}\mathcal{D}(A \times \_)$ is injective for

$$(A \times \mathcal{D}) \stackrel{\tau}{\Rightarrow} \mathcal{D}(A \times \_)$$

given by $\tau_X(\langle a, \mu\rangle) = \delta_a \times \mu$, where

Example

Indeed $\mathbf{SSeg} \to \mathbf{Seg}$ since $\mathcal{P}(A \times \mathcal{D}) \stackrel{\mathcal{P}\tau}{\Rightarrow} \mathcal{P}\mathcal{D}(A \times \_)$ is injective for

$$(A \times \mathcal{D}) \stackrel{\tau}{\Rightarrow} \mathcal{D}(A \times \_)$$

given by $\tau_X(\langle a, \mu \rangle) = \delta_a \times \mu$, where

$$\mu \times \mu'(\langle x, x' \rangle) = \mu(x) \cdot \mu'(x')$$

Example

Indeed $\mathbf{SSeg} \to \mathbf{Seg}$ since $\mathcal{P}(A \times \mathcal{D}) \stackrel{\mathcal{P}\tau}{\Rightarrow} \mathcal{P}\mathcal{D}(A \times \_)$ is injective for

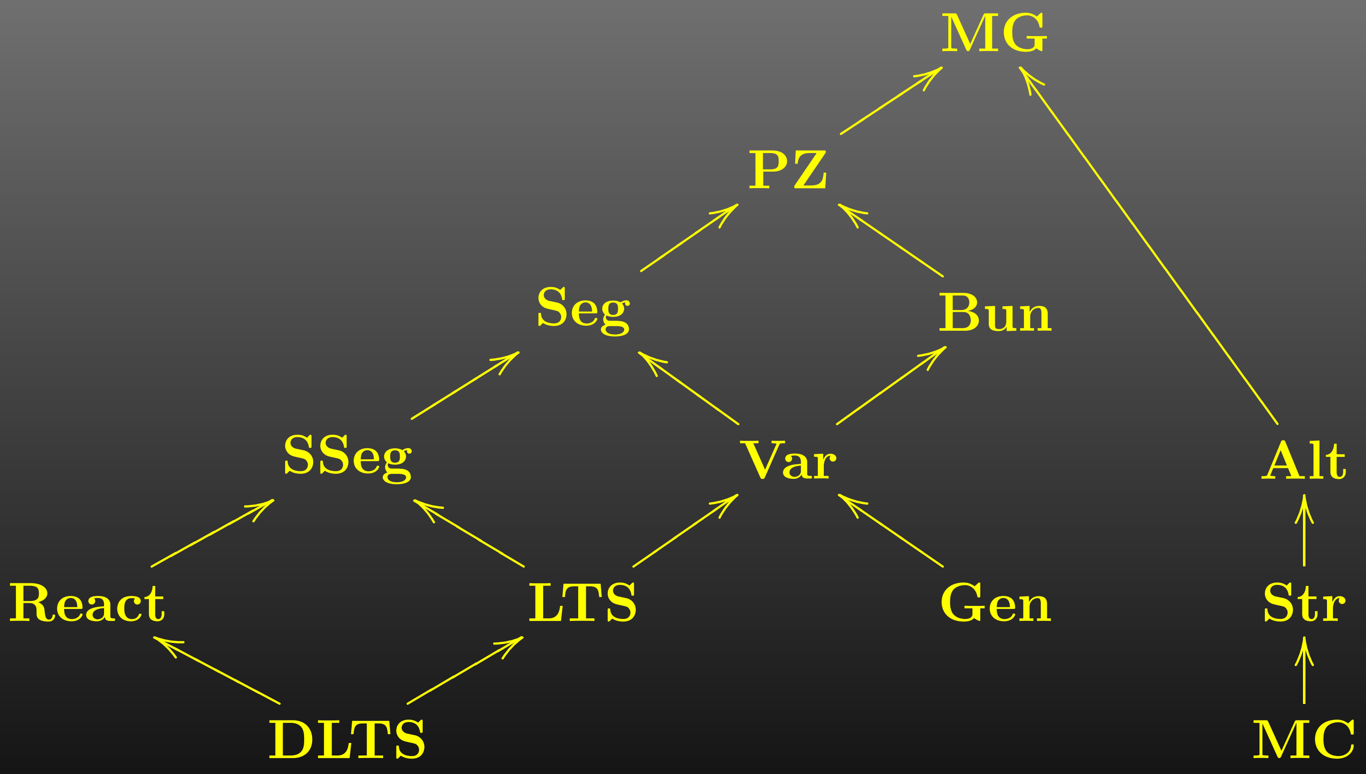
$$(A \times \mathcal{D}) \stackrel{\tau}{\Rightarrow} \mathcal{D}(A \times \_)$$

given by $\tau_X(\langle a, \mu \rangle) = \delta_a \times \mu$, where

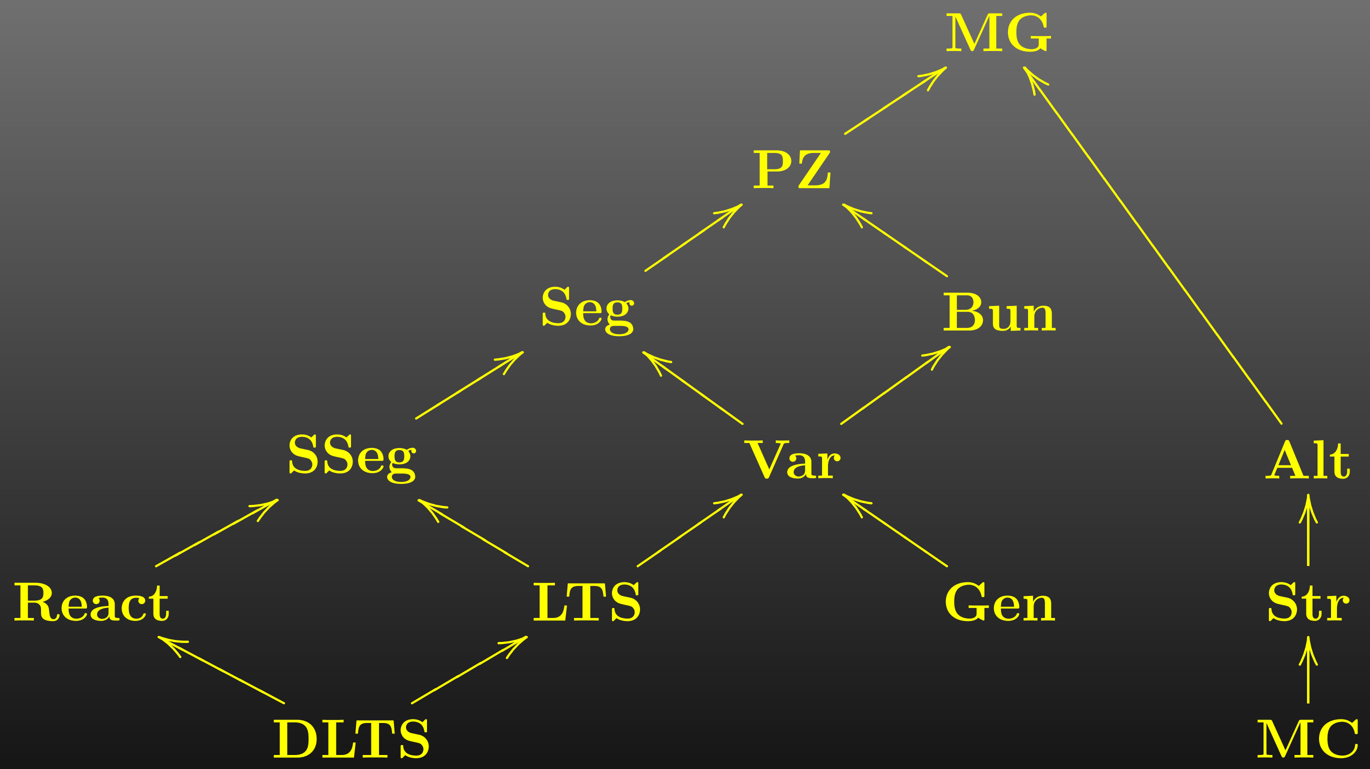
$$\mu \times \mu'(\langle x, x' \rangle) = \mu(x) \cdot \mu'(x')$$

and δ_a is Dirac distribution for a

The hierarchy...



The hierarchy...



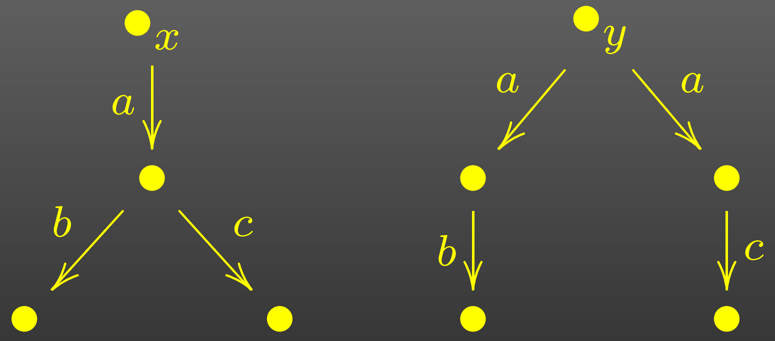
* Falk Bartels, AS, Erik de Vink

LT/BT spectrum

Bisimilarity is not the only semantics...

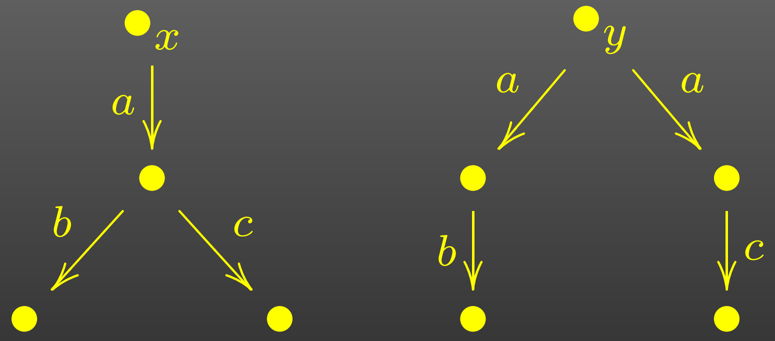
LT/BT spectrum

Are these non-deterministic systems equal ?



LT/BT spectrum

Are these non-deterministic systems equal ?

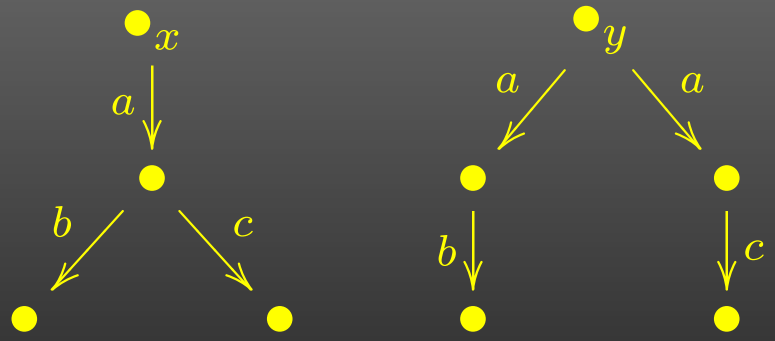


x and y are:

- different wrt. bisimilarity

LT/BT spectrum

Are these non-deterministic systems equal ?



x and y are:

- different wrt. bisimilarity, but
- equivalent wrt. trace semantics
  $\mathrm{tr}(x) = \mathrm{tr}(y) = \{ab, ac\}$

Traces - LTS

For LTS with explicit termination (NA)

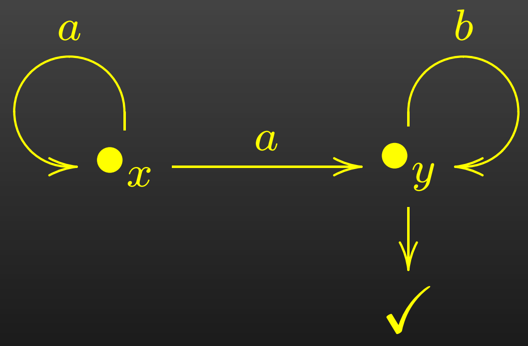
trace = the set of all possible linear behaviors

Traces - LTS

For LTS with explicit termination (NA)

trace = the set of all possible linear behaviors

Example:



$$\mathrm{tr}(y) \;=\; b^*, \qquad \mathrm{tr}(x) \;=\; a^+ \cdot \mathrm{tr}(y) \;=\; a^+ \cdot b^*$$

Traces - generative

For generative probabilistic systems with ex. termination

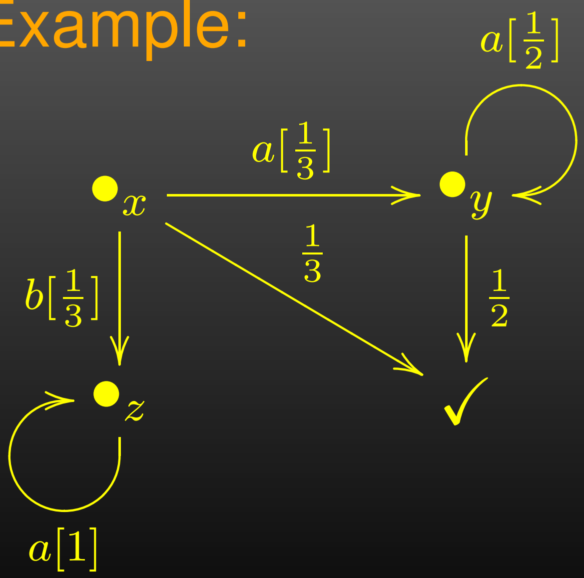
trace = sub-probability distribution over possible linear behaviors

Traces - generative

For generative probabilistic systems with ex. termination

trace = sub-probability distribution over possible linear behaviors

Example:



$$\mathrm{tr}(x): \quad \langle\rangle \mapsto \frac{1}{3}$$

$$a \mapsto \frac{1}{3} \cdot \frac{1}{2}$$

$$a^2 \mapsto \frac{1}{3} \cdot \frac{1}{2} \cdot \frac{1}{2}$$

$$\cdots$$

Trace of a coalgebra ?

Trace of a coalgebra ?

- Power&Turi '99
- Jacobs '04
- Hasuo& Jacobs '05
- Ichiro Hasuo, Bart Jacobs, AS:
  Generic Trace Theory, CMCS'06

Trace of a coalgebra ?

- Power&Turi ’99
- Jacobs ’04
- Hasuo& Jacobs ’05
- Ichiro Hasuo, Bart Jacobs, AS:
  Generic Trace Theory, CMCS’06

main idea: coinduction in a Kleisli category

Coinduction

$$\begin{array}{ccc}
\mathcal{F}X & \overset{\mathcal{F}(\mathrm{beh})}{- - - - \dashrightarrow} & \mathcal{F}Z \\
\uparrow{\scriptstyle \alpha} & & \uparrow{\scriptstyle \cong} \\
X & \underset{\mathrm{beh}}{- - - - \dashrightarrow} & Z
\end{array}$$

system final coalgebra

Coinduction

$$
\begin{array}{ccc}
\mathcal{F}X & \xrightarrow{\mathcal{F}(\mathrm{beh})} & \mathcal{F}Z \\
\uparrow \alpha & & \uparrow \cong \\
X & \xrightarrow[\mathrm{beh}]{} & Z
\end{array}
$$

system final coalgebra

- finality = $\exists!$(morphism for any $\mathcal{F}$- coalgebra)
- beh gives the behavior of the system
- this yields final coalgebra semantics

Coinduction

$$\begin{array}{ccc} \mathcal{F}X & \xrightarrow{\mathcal{F}(\text{beh})} & \mathcal{F}Z \\ \uparrow{\scriptstyle \alpha} & & \uparrow{\scriptstyle \cong} \\ X & \xrightarrow[\text{beh}]{} & Z \end{array}$$

system final coalgebra

- f.c.s. in **Sets** = bisimilarity
- f.c.s. in a Kleisli category = trace semantics

Types of systems

For trace semantics systems are suitably modelled as coalgebras in $\mathbf{Sets}$

$$X \xrightarrow{c} \mathcal{T}\,\mathcal{F}X$$

Types of systems

For trace semantics systems are suitably modelled as coalgebras in $\mathbf{Sets}$

$$X \xrightarrow{c} \mathcal{T}\mathcal{F}X$$

monad - branching type

Types of systems

For trace semantics systems are suitably modelled as coalgebras in $\mathbf{Sets}$

$$X \xrightarrow{c} \mathcal{T}\,\mathcal{F}\,X$$

monad - branching type

functor - linear i/o type

Types of systems

For trace semantics systems are suitably modelled as coalgebras in $\mathbf{Sets}$

$$X \xrightarrow{c} \mathcal{T}\mathcal{F}X$$

monad - branching type

functor - linear i/o type

needed: distributive law $\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$

Distributive law

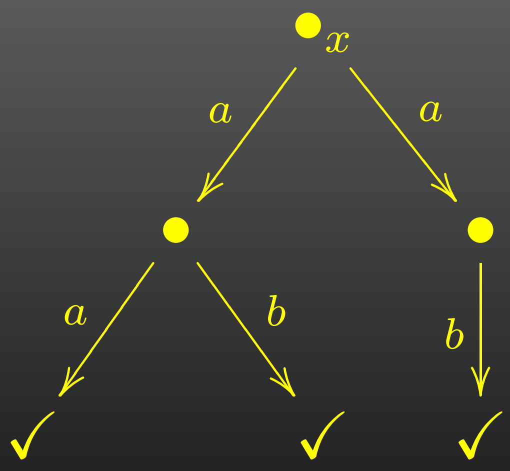
is needed since branching is irrelevant:

LTS with $\checkmark$ - $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$

Distributive law

is needed since branching is irrelevant:

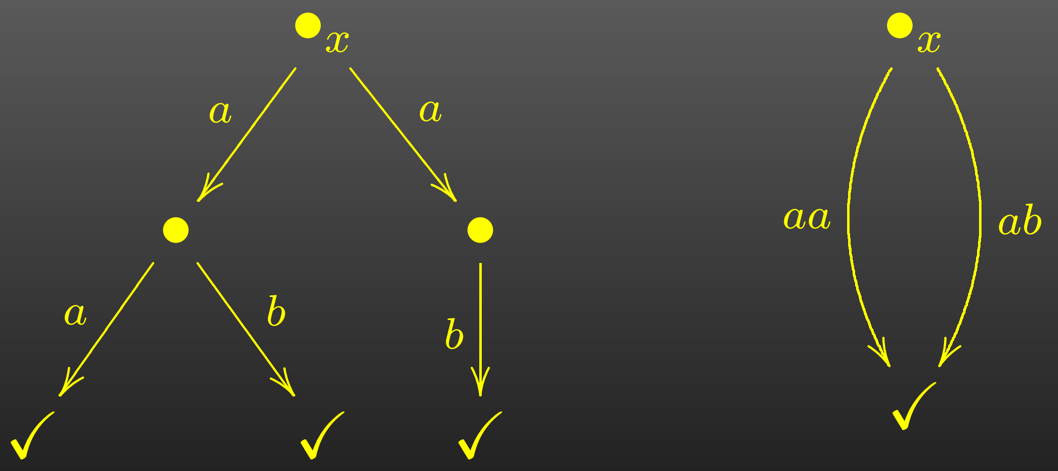
LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$



Distributive law

is needed since branching is irrelevant:

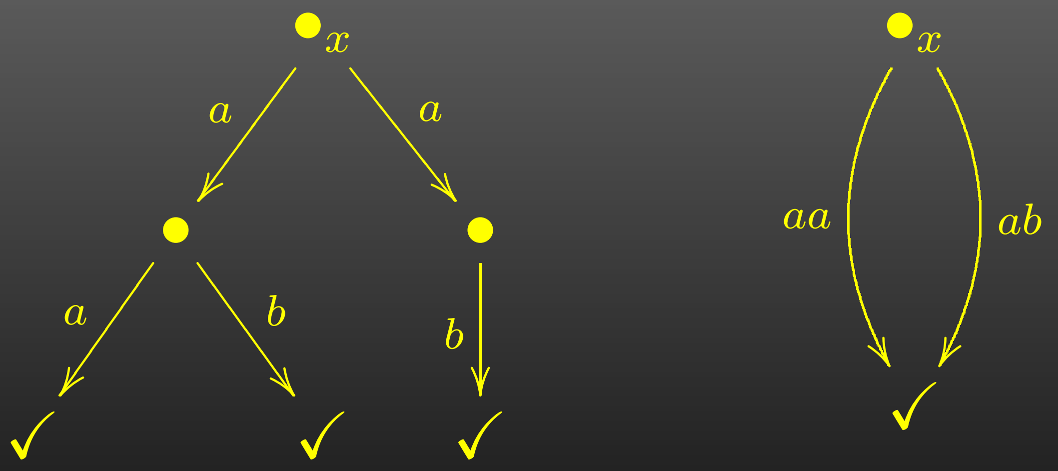
LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$



Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$

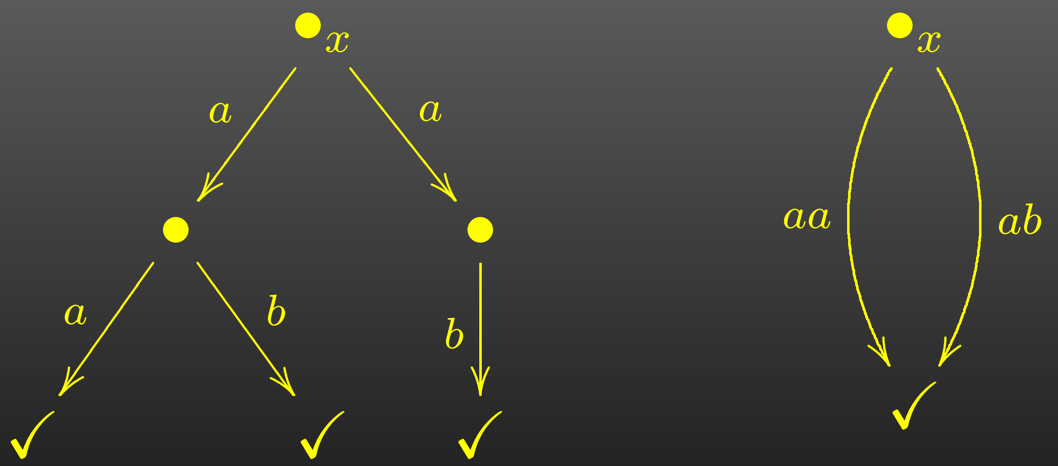


$X \xrightarrow{c} \mathcal{PF}X$

Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$

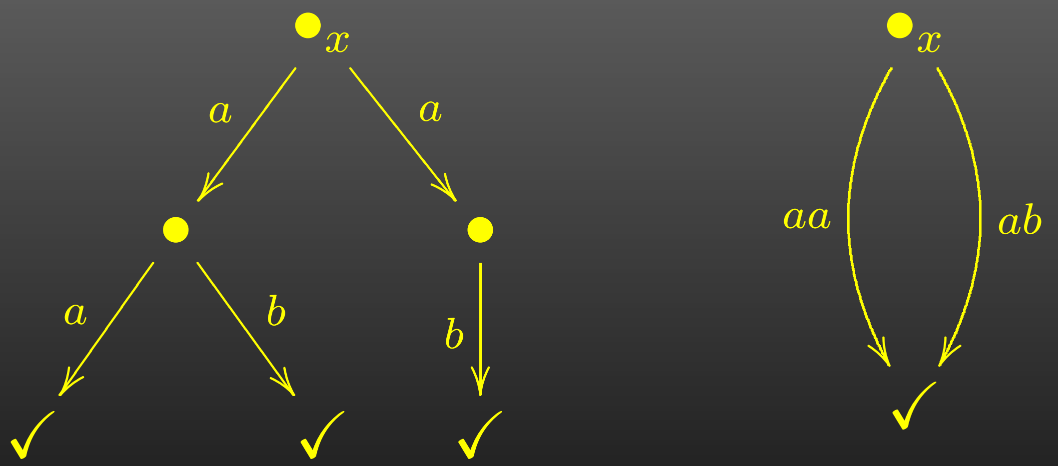


$X \xrightarrow{c} \mathcal{PF}X$

Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$

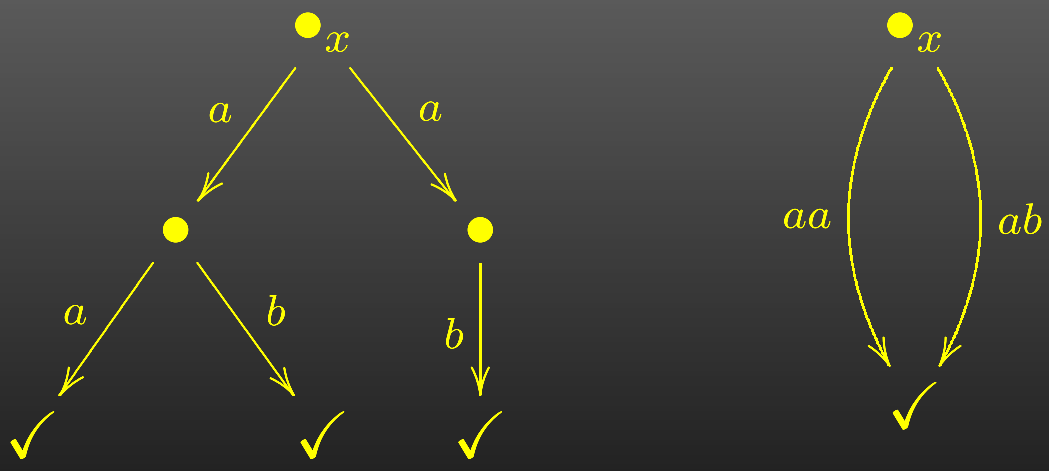


$$X \stackrel{c}{\to} \mathcal{PF}X \stackrel{\mathcal{PF}c}{\to} \mathcal{PFPF}X$$

Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{P}\mathcal{F} = \mathcal{P}(1 + A \times \_)$

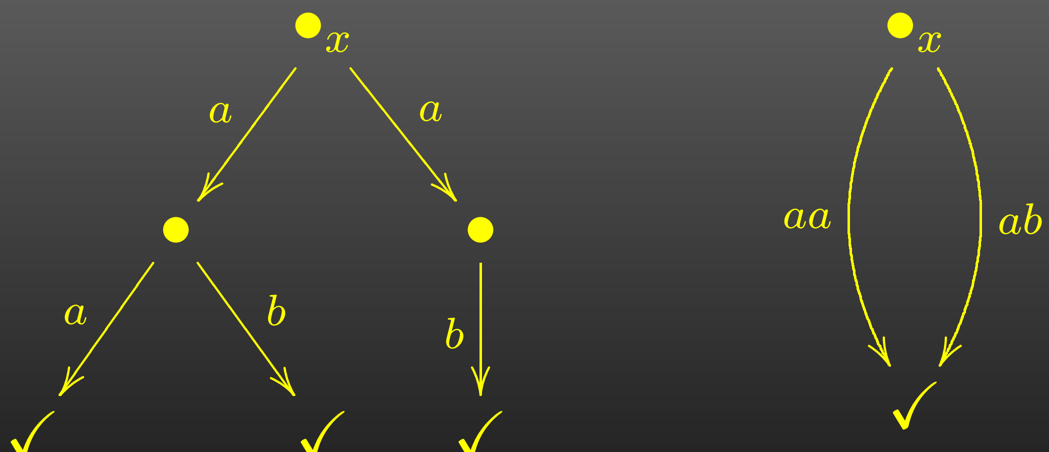


$$X \xrightarrow{c} \mathcal{P}\mathcal{F}X \xrightarrow{\mathcal{P}\mathcal{F}c} \mathcal{P}\mathcal{F}\mathcal{P}\mathcal{F}X$$

Distributive law

is needed since branching is irrelevant:

LTS with ✓ - $\mathcal{P}\mathcal{F} = \mathcal{P}(1 + A \times \_)$

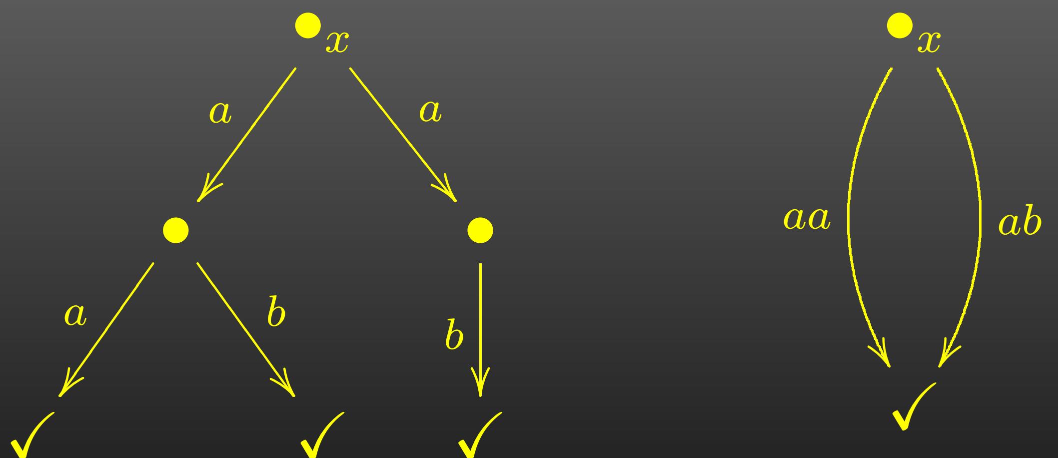


$$X \xrightarrow{c} \mathcal{P}\mathcal{F}X \xrightarrow{\mathcal{P}\mathcal{F}c} \mathcal{P}\mathcal{F}\mathcal{P}\mathcal{F}X \xrightarrow{\text{d.l.}} \mathcal{P}\mathcal{P}\mathcal{F}\mathcal{F}X$$

Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$

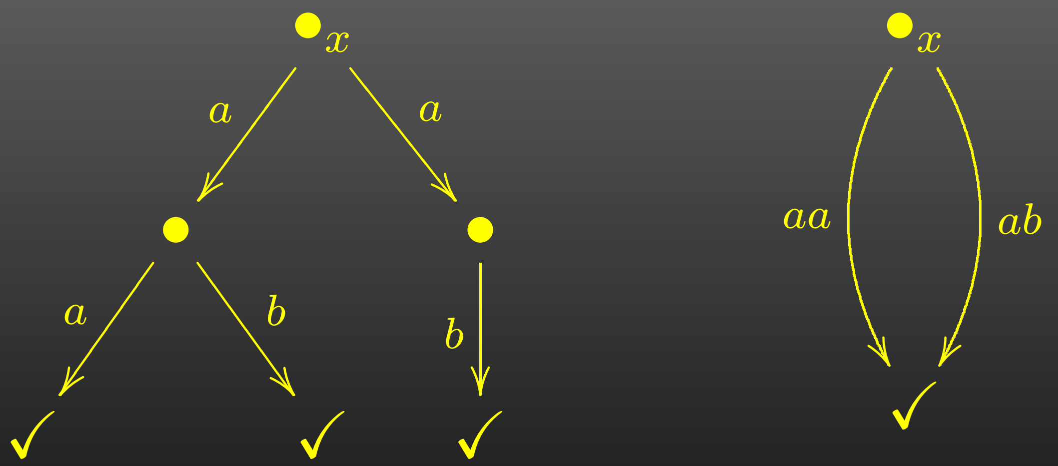


$$X \stackrel{c}{\to} \mathcal{PF}X \stackrel{\mathcal{PF}c}{\to} \mathcal{PFPF}X \stackrel{\text{d.l.}}{\to} \mathcal{PPFF}X$$

Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{P}\mathcal{F} = \mathcal{P}(1 + A \times \_)$

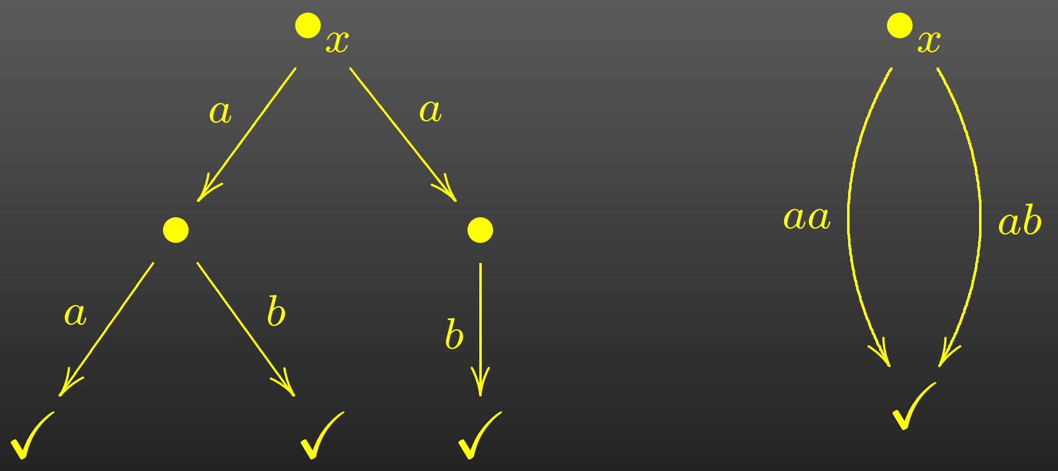


$$X \stackrel{c}{\to} \mathcal{P}\mathcal{F}X \stackrel{\mathcal{P}\mathcal{F}c}{\to} \mathcal{P}\mathcal{F}\mathcal{P}\mathcal{F}X \stackrel{\text{d.l.}}{\to} \mathcal{P}\mathcal{P}\mathcal{F}\mathcal{F}X \stackrel{\text{m.m.}}{\to} \mathcal{P}\mathcal{F}\mathcal{F}X$$

Distributive law

is needed since branching is irrelevant:

LTS with ✓- $\mathcal{PF} = \mathcal{P}(1 + A \times \_)$



$$X \xrightarrow{c} \mathcal{PF}X \xrightarrow{\mathcal{PF}c} \mathcal{PFPF}X \xrightarrow{\text{d.l.}} \mathcal{PPFF}X \xrightarrow{\text{m.m.}} \mathcal{PFF}X$$

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

- objects - sets
- arrows - $X \xrightarrow{f} Y$ are functions $f : X \to \mathcal{T}Y$

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

$\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: $\mathcal{F}$ lifts to $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ on $\mathcal{K}\ell(\mathcal{T})$.

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

$\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: $\mathcal{F}$ lifts to $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ on $\mathcal{K}\ell(\mathcal{T})$.

Hence: coalgebra $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$!!!

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

$\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: $\mathcal{F}$ lifts to $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ on $\mathcal{K}\ell(\mathcal{T})$.

Hence: coalgebra $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$!!!

in $\mathcal{K}\ell(\mathcal{T})$: $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

$\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: $\mathcal{F}$ lifts to $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ on $\mathcal{K}\ell(\mathcal{T})$.

Hence: coalgebra $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$!!!

in $\mathcal{K}\ell(\mathcal{T})$: $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

$\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: $\mathcal{F}$ lifts to $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ on $\mathcal{K}\ell(\mathcal{T})$.

Hence: coalgebra $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$!!!

in $\mathcal{K}\ell(\mathcal{T})$: $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X \xrightarrow{\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$

Distributive law

is needed for $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ to be a coalgebra in the Kleisli category $\mathcal{K}\ell(\mathcal{T})$..

$\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: $\mathcal{F}$ lifts to $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ on $\mathcal{K}\ell(\mathcal{T})$.

Hence: coalgebra $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$!!!

in $\mathcal{K}\ell(\mathcal{T})$: $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X \xrightarrow{\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X \to \cdots$

Main theorem - traces

If ♣, then

$$\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \xrightarrow[\cong]{\eta_I \circ \alpha} I \qquad \text{is initial} \qquad\qquad \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \xleftarrow[\cong]{\eta_{\mathcal{F}I} \circ \alpha^{-1}} I \qquad \text{is final} \qquad \text{in } \mathcal{K}\ell(\mathcal{T})$$

Main theorem - traces

If ♣, then

$$\begin{array}{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \\ {\scriptstyle \eta_I \circ \alpha}\ \big\downarrow\ {\scriptstyle \cong} \\ I \end{array} \qquad\qquad \begin{array}{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \\ {\scriptstyle \eta_{\mathcal{F}I} \circ \alpha^{-1}}\ \big\uparrow\ {\scriptstyle \cong} \\ I \end{array}$$

is initial   is final   in $\mathcal{K}\ell(\mathcal{T})$

[$\alpha : \mathcal{F}I \stackrel{\cong}{\to} I$ denotes the initial $\mathcal{F}$-algebra in $\mathbf{Sets}$]

Main theorem - traces

If ♣, then

$$\begin{array}{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \\ {\scriptstyle \eta_I \circ \alpha} \Big\downarrow {\scriptstyle \cong} \\ I \end{array} \qquad\qquad \begin{array}{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \\ {\scriptstyle \eta_{\mathcal{F}I} \circ \alpha^{-1}} \Big\uparrow {\scriptstyle \cong} \\ I \end{array}$$

is initial is final in $\mathcal{K}\ell(\mathcal{T})$

[$\alpha : \mathcal{F}I \stackrel{\cong}{\to} I$ denotes the initial $\mathcal{F}$-algebra in $\mathbf{Sets}$]

proof: via limit-colimit coincidence Smyth&Plotkin ’82

The assumptions ♣:

- A monad $\mathcal{T}$ s.t. $\mathcal{K}\ell(\mathcal{T})$ is $\mathbf{DCpo}_\perp$-enriched
  left-strict composition

The assumptions ♣:

- A monad $\mathcal{T}$ s.t. $\mathcal{K}\ell(\mathcal{T})$ is $\mathbf{DCpo}_\perp$-enriched left-strict composition
- A functor $\mathcal{F}$ that preserves ω-colimits

The assumptions ♣:

- A monad $\mathcal{T}$ s.t. $\mathcal{K}\ell(\mathcal{T})$ is $\mathbf{DCpo}_{\perp}$-enriched left-strict composition
- A functor $\mathcal{F}$ that preserves ω-colimits
- A distributive law $\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: lifting $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$

The assumptions ♣:

- A monad $\mathcal{T}$ s.t. $\mathcal{K}\ell(\mathcal{T})$ is $\mathbf{DCpo}_\perp$-enriched left-strict composition
- A functor $\mathcal{F}$ that preserves ω-colimits
- A distributive law $\mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$: lifting $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$
- $\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}$ should be locally monotone

Corollary (♣)

For $X \stackrel{c}{\to} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$

Corollary (♣)

For $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} X$ in $\mathcal{K}\ell(\mathcal{T})$ ... $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ in $\mathbf{Sets}$

Corollary (♣)

For $X \stackrel{c}{\to} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}X$ in $\mathcal{K}\ell(\mathcal{T})$ ... $X \stackrel{c}{\to} \mathcal{T}\mathcal{F}X$ in $\mathbf{Sets}$

$\exists!$ finite trace map $\mathrm{tr}_c : X \to \mathcal{T}I$ in $\mathbf{Sets}$:

Corollary (♣)

For $X \xrightarrow{c} \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} X$ in $\mathcal{K}\ell(\mathcal{T})$ ... $X \xrightarrow{c} \mathcal{T}\mathcal{F}X$ in **Sets**

$\exists!$ finite trace map $\mathrm{tr}_c : X \to \mathcal{T}I$ in **Sets**:

in $\mathcal{K}\ell(\mathcal{T})$

$$\begin{array}{ccc}
\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} X & \overset{\mathcal{F}_{\mathcal{K}\ell(\mathcal{T})}(\mathrm{tr}_c)}{- - - - - \to} & \mathcal{F}_{\mathcal{K}\ell(\mathcal{T})} I \\
{\scriptstyle c}\uparrow & & \uparrow{\scriptstyle \cong} \\
X & \underset{\mathrm{tr}_c}{- - - - - \to} & I
\end{array}$$

It works for...

- lift, powerset, sub-distribution monad

It works for...

- lift, powerset, sub-distribution monad
- shapely functors - almost all polynomial

It works for...

- lift, powerset, sub-distribution monad
- shapely functors - almost all polynomial
- Hence:

It works for...

- lift, powerset, sub-distribution monad
- shapely functors - almost all polynomial
- Hence:
  * for LTS with explicit termination $\mathcal{P}(1 + A \times \_)$

It works for...

- lift, powerset, sub-distribution monad
- shapely functors - almost all polynomial
- Hence:
  - for LTS with explicit termination $\mathcal{P}(1 + A \times \_)$
  - for generative systems with explicit termination $\mathcal{D}(1 + A \times \_)$

It works for...

- lift, powerset, sub-distribution monad
- shapely functors - almost all polynomial
- Hence:
  - * for LTS with explicit termination $\mathcal{P}(1 + A \times \_\,)$
  - * for generative systems with explicit termination $\mathcal{D}(1 + A \times \_\,)$

Note: Initial $1 + A \times \_$ - algebra is

$$A^* \xrightarrow[\cong]{[\text{nil},\text{cons}]} 1 + A \times A^*$$

Finite traces - LTS with ✓

the finality diagram in $\mathcal{K}\ell(\mathcal{P})$

$$\begin{array}{ccc}
\mathcal{F}_{\mathcal{K}\ell(\mathcal{P})}X & \xrightarrow{\mathcal{F}_{\mathcal{K}\ell(\mathcal{P})}(\mathrm{tr}_c)} & \mathcal{F}_{\mathcal{K}\ell(\mathcal{P})}A^* \\
c \uparrow & & \uparrow \cong \\
X & \xrightarrow[\mathrm{tr}_c]{} & A^*
\end{array}$$

Finite traces - LTS with ✓

the finality diagram in $\mathcal{K}\ell(\mathcal{P})$

$$\begin{array}{ccc} 1 + A \times X & \xrightarrow{(1+A\times\_)_{\mathcal{K}\ell(\mathcal{P})}(\mathrm{tr}_c)} & 1 + A \times A^* \\ c \uparrow & & \uparrow \cong \\ X & \xrightarrow[\mathrm{tr}_c]{} & A^* \end{array}$$

Finite traces - LTS with ✓

the finality diagram in $\mathcal{K}\ell(\mathcal{P})$

$$\begin{array}{ccc}
1 + A \times X & \xrightarrow{(1+A\times_)_{\mathcal{K}\ell(\mathcal{P})}(\mathrm{tr}_c)} & 1 + A \times A^* \\
\uparrow c & & \uparrow \cong \\
X & \xrightarrow[\mathrm{tr}_c]{} & A^*
\end{array}$$

amounts to

- $\langle\rangle \in \mathrm{tr}_c(x) \iff \checkmark \in c(x)$
- $a \cdot w \in \mathrm{tr}_c(x) \iff (\exists x')\langle a, x'\rangle \in c(x),\ w \in \mathrm{tr}_c(x')$

Finite traces - generative ✓

the finality diagram in $\mathcal{K}\ell(\mathcal{D})$

$$\begin{array}{ccc}
\mathcal{F}_{\mathcal{K}\ell(\mathcal{D})} X & \overset{\mathcal{F}_{\mathcal{K}\ell(\mathcal{D})}(\mathrm{tr}_c)}{- - - - - \rightarrow} & \mathcal{F}_{\mathcal{K}\ell(\mathcal{D})} A^* \\
c \uparrow & & \uparrow \cong \\
X & \underset{\mathrm{tr}_c}{- - - - - \rightarrow} & A^*
\end{array}$$

Finite traces - generative ✓

the finality diagram in $\mathcal{K}\ell(\mathcal{D})$

$$\begin{array}{ccc}
1 + A \times X & \xrightarrow{(1+A\times_)_{\mathcal{K}\ell(\mathcal{D})}(\mathrm{tr}_c)} & 1 + A \times A^* \\
\uparrow c & & \uparrow \cong \\
X & \xrightarrow[\mathrm{tr}_c]{} & A^*
\end{array}$$

Finite traces - generative ✓

the finality diagram in $\mathcal{K}\ell(\mathcal{D})$

$$
\begin{array}{ccc}
1 + A \times X & \xrightarrow{(1+A\times\_)_{\mathcal{K}\ell(\mathcal{D})}(\mathrm{tr}_c)} & 1 + A \times A^* \\
\uparrow{\scriptstyle c} & & \uparrow{\scriptstyle \cong} \\
X & \xrightarrow[\mathrm{tr}_c]{} & A^*
\end{array}
$$

amounts to $\mathrm{tr}_c(x)$:

- $\langle\rangle \mapsto c(x)(\checkmark)$
- $a \cdot w \mapsto \sum_{y \in X} c(x)(a, y) \cdot c(y)(w)$

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS
- Coinduction gives us semantic relations:

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS
- Coinduction gives us semantic relations:
  - bisimilarity for $\mathcal{F}$-systems in $\mathbf{Sets}$

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS
- Coinduction gives us semantic relations:
  - * bisimilarity for $\mathcal{F}$-systems in $\mathbf{Sets}$
  - * trace semantics for $\mathcal{T}\mathcal{F}$-systems in $\mathcal{K}\ell(T)$

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS
- Coinduction gives us semantic relations:
  - * bisimilarity for $\mathcal{F}$-systems in $\mathbf{Sets}$
  - * trace semantics for $\mathcal{T}\mathcal{F}$-systems in $\mathcal{K}\ell(T)$
- Other monads e.g. $\mathcal{P}\mathcal{D}$? ... suitable monad/order structure yet to be found (Varacca&Winskel)

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS
- Coinduction gives us semantic relations:
  - bisimilarity for $\mathcal{F}$-systems in $\mathbf{Sets}$
  - trace semantics for $\mathcal{T}\mathcal{F}$-systems in $\mathcal{K}\ell(T)$
- Other monads e.g. $\mathcal{P}\mathcal{D}$? ... suitable monad/order structure yet to be found (Varacca&Winskel)
- Other semantics ? .. between bisimilarity and trace in the spectrum

Conclusions & future work

- Coalgebras allow for a unified treatment and expressiveness study of (P)TS
- Coinduction gives us semantic relations:
  - bisimilarity for $\mathcal{F}$-systems in $\mathbf{Sets}$
  - trace semantics for $\mathcal{T}\mathcal{F}$-systems in $\mathcal{K}\ell(T)$
- Other monads e.g. $\mathcal{P}\mathcal{D}$? ... suitable monad/order structure yet to be found (Varacca&Winskel)
- Other semantics ? .. between bisimilarity and trace in the spectrum