Hierarchy of probabilistic systems

Falk Bartels, Ana Sokolova, Erik de Vink

falk.bartels@cwi.nl, {asokolov,evink}@win.tue.nl

CWI and TU/e

Outline

- Introduction

Outline

- Introduction
  * labelled transition systems

Outline

- Introduction
  * labelled transition systems
  * bisimulation

Outline

- Introduction
  * labelled transition systems
  * bisimulation
  * adding probabilities

Outline

- Introduction
  * labelled transition systems
  * bisimulation
  * adding probabilities
- Probabilistic system types

Outline

- Introduction
  - labelled transition systems
  - bisimulation
  - adding probabilities
- Probabilistic system types
- Comparison of system types

Outline

- Introduction
  * labelled transition systems
  * bisimulation
  * adding probabilities
- Probabilistic system types
- Comparison of system types
  * expressiveness criterion

Outline

- Introduction
  - labelled transition systems
  - bisimulation
  - adding probabilities
- Probabilistic system types
- Comparison of system types
  - expressiveness criterion
  - translation of coalgebras

Outline

- Introduction
  - * labelled transition systems
  - * bisimulation
  - * adding probabilities
- Probabilistic system types
- Comparison of system types
  - * expressiveness criterion
  - * translation of coalgebras
  - * preservation and reflection of bisimulation

Outline

- Introduction
  - * labelled transition systems
  - * bisimulation
  - * adding probabilities
- Probabilistic system types
- Comparison of system types
  - * expressiveness criterion
  - * translation of coalgebras
  - * preservation and reflection of bisimulation
- Building the hierarchy

Outline

- Introduction
  - * labelled transition systems
  - * bisimulation
  - * adding probabilities
- Probabilistic system types
- Comparison of system types
  - * expressiveness criterion
  - * translation of coalgebras
  - * preservation and reflection of bisimulation
- Building the hierarchy
- Conclusions

Labelled transition systems

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

A - a fixed set of actions

Labelled transition systems

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

A - a fixed set of actions

Hence a coalgebra

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

of the functor $\mathcal{F} = \mathcal{P}^A$

Labelled transition systems

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

A - a fixed set of actions

Example:

$$\begin{tikzcd}
& s_2 \arrow[r, "a"] \arrow[dr, "a"'] & s_4 & & \\
s_1 \arrow[ur, "b"] \arrow[dr, "a"'] & & s_5 \arrow[r, "b"] & s_6 & \cdots \\
& s_3 & & &
\end{tikzcd}$$

Labelled transition systems

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

A - a fixed set of actions

Example:

$$
\begin{tikzcd}
 & s_2 \arrow[r, "a"] \arrow[dr, "a"'] & s_4 & & \\
s_1 \arrow[ur, "b"] \arrow[dr, "a"'] & & s_5 \arrow[r, "b"] & s_6 & \cdots \\
 & s_3 & & &
\end{tikzcd}
$$

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

Concrete bisimulation for LTS

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

Concrete bisimulation for LTS

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

$R \subseteq S \times T$ is a *concrete bisimulation*
if for all $\langle s, t \rangle \in R$ and all $a \in A$

Concrete bisimulation for LTS

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

$R \subseteq S \times T$ is a *concrete bisimulation*
if for all $\langle s, t \rangle \in R$ and all $a \in A$

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

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

Concrete bisimulation for LTS

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

$R \subseteq S \times T$ is a *concrete bisimulation*
if for all $\langle s, t \rangle \in R$ and all $a \in A$

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

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

$s \approx t$ - there exists a concrete bisimulation

R with $\langle s, t \rangle \in R$

Coalgebraic bisimulation

A *bisimulation* between two $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ is a relation

$$R \subseteq S \times T$$

such that there exists an $\mathcal{F}$-coalgebra structure γ on R making

Coalgebraic bisimulation

A *bisimulation* between two $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ is a relation

$$R \subseteq S \times T$$

such that there exists an $\mathcal{F}$-coalgebra structure γ on R making

$$\begin{tikzcd}
S \arrow[d, "\alpha"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] & T \arrow[d, "\beta"] \\
\mathcal{F}S & \mathcal{F}R \arrow[l, "\mathcal{F}\pi_1"'] \arrow[r, "\mathcal{F}\pi_2"] & \mathcal{F}T
\end{tikzcd}$$

Concrete vs coalgebraic (LTS)

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

(coalgebras of type $\mathcal{F} = \mathcal{P}^A$)

$R \subseteq S \times T$ - bisimulation between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$

Concrete vs coalgebraic (LTS)

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

(coalgebras of type $\mathcal{F} = \mathcal{P}^A$)

$R \subseteq S \times T$ - bisimulation between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$

$s \sim t$ - there exists a bisimulation R with sRt

Concrete vs coalgebraic (LTS)

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - LTS

(coalgebras of type $\mathcal{F} = \mathcal{P}^A$)

$R \subseteq S \times T$ - bisimulation between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$

$s \sim t$ - there exists a bisimulation R with sRt

known: $s \approx t$ if and only if $s \sim t$

Introduction of probabilities

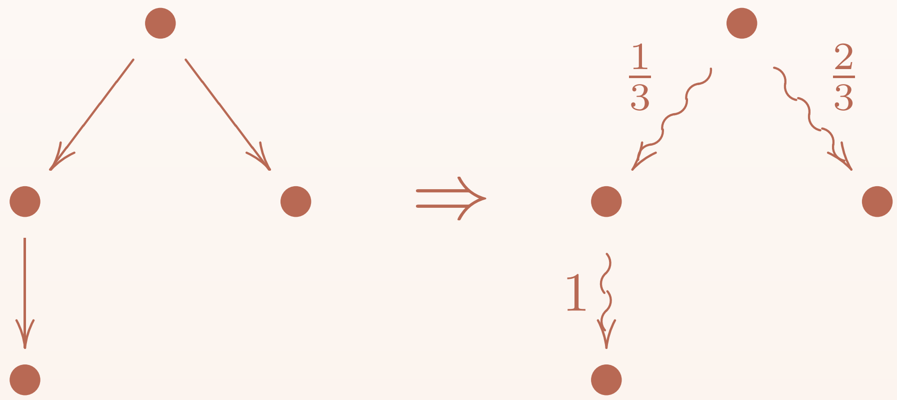
There are many ways to do it ...

Examples:

Introduction of probabilities

There are many ways to do it ...

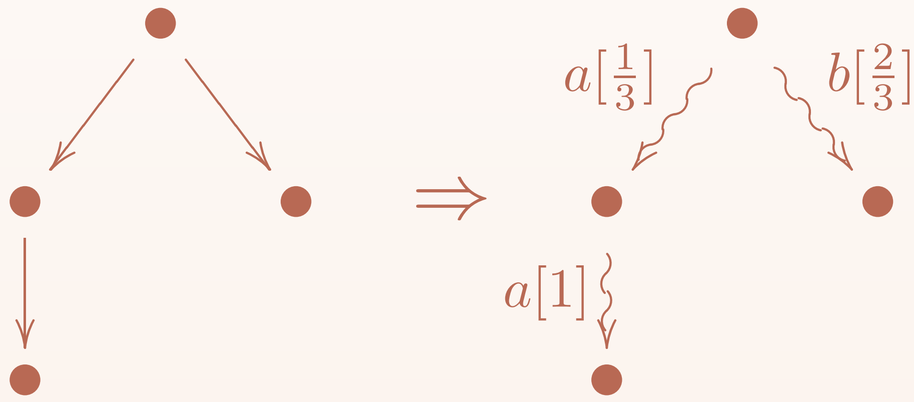
Examples:



Introduction of probabilities

There are many ways to do it ...

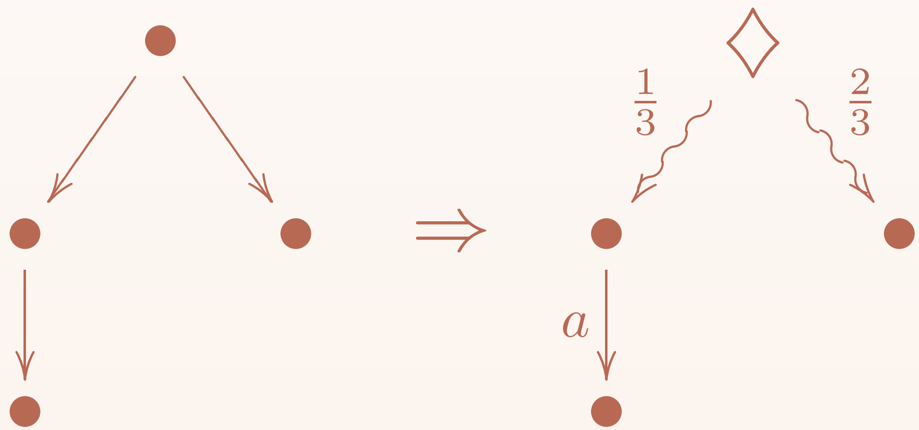
Examples:



Introduction of probabilities

There are many ways to do it ...

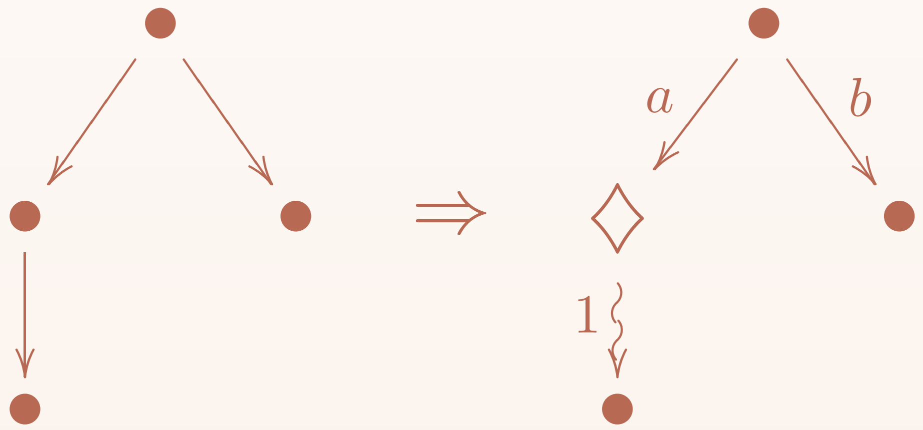
Examples:



Introduction of probabilities

There are many ways to do it ...

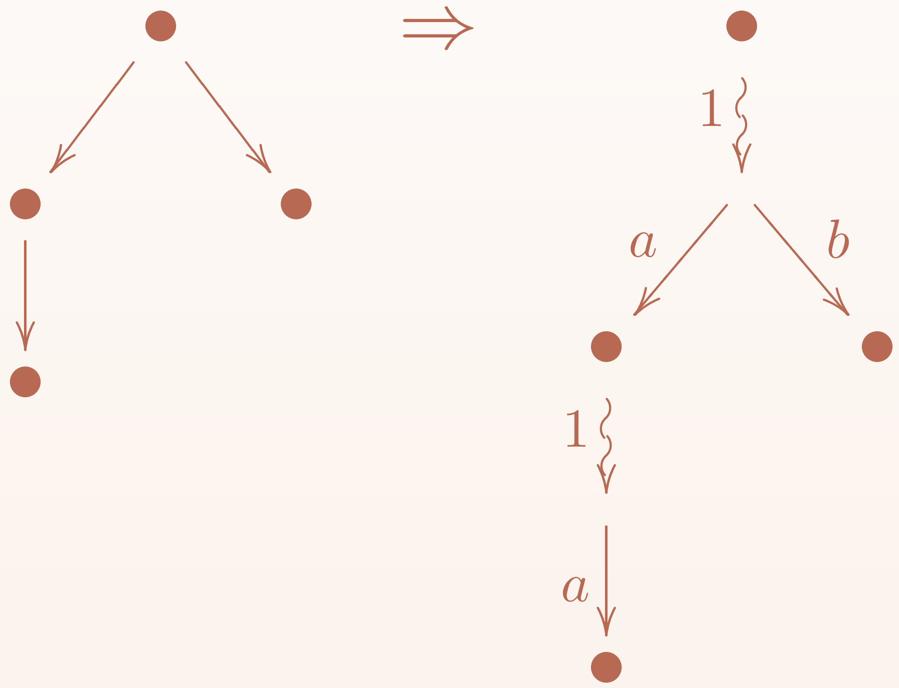
Examples:



Introduction of probabilities

There are many ways to do it ...

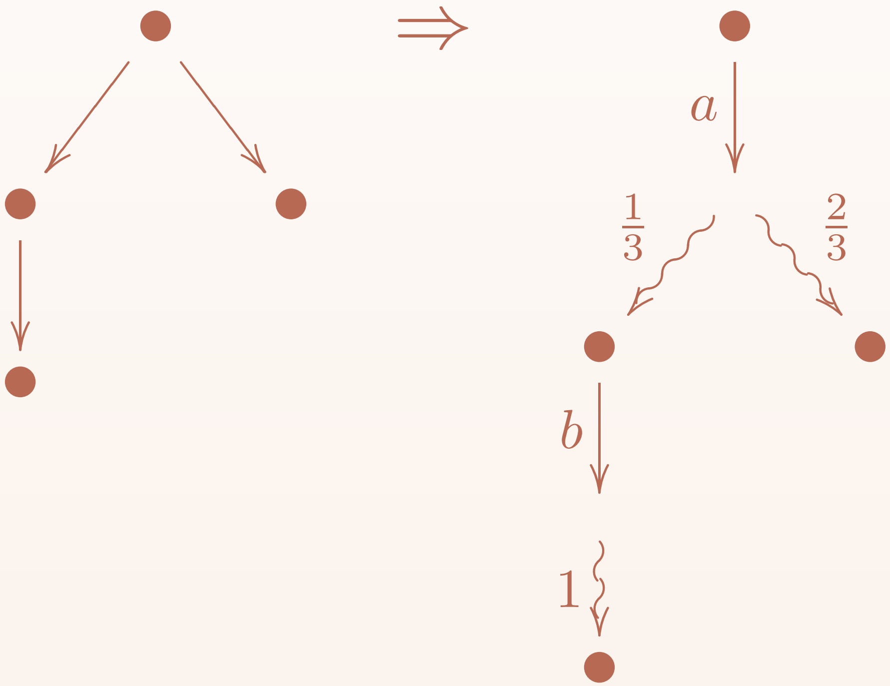
Examples:



Introduction of probabilities

There are many ways to do it ...

Examples:



Introduction of probabilities

There are many ways to do it ...

13 types of systems - from the literature

with (or without):

- action labels
- nondeterminism
- probabilities

Existing system types

MG

PZ

Alt **Seg** **Bun**

SSeg **Var**

Str **React** **NA** **Gen**

MC **DA**

System types

The (probabilistic) models of systems we consider are coalgebras

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

for a functor $\mathcal{F}$ built by the following syntax

System types

The (probabilistic) models of systems we consider are coalgebras

$$\langle S, \alpha \rangle, \;\; \alpha : S \to \mathcal{F}S$$

for a functor $\mathcal{F}$ built by the following syntax

$$\mathcal{F} ::= A \mid \mathcal{I} \mid \mathcal{P} \mid \mathcal{D}_\omega \mid \mathcal{F} + \mathcal{F} \mid \mathcal{F} \times \mathcal{F} \mid \mathcal{F}^A \mid \mathcal{F}\mathcal{F}$$

Reactive and generative systems

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

Reactive and generative systems

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

Reactive systems

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

Reactive and generative systems

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

Reactive systems

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

Reactive and generative systems

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

Reactive systems

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

Generative systems

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

Reactive and generative systems

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

Reactive systems

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

Generative systems

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

Reactive and generative systems

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

Reactive systems

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

Generative systems

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

note:

In the probabilistic case

$(\mathcal{D}_\omega + 1)^A \ncong \mathcal{D}_\omega(A \times \mathcal{I}) + 1$

Reactive and generative systems

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

Reactive systems - input type

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

Generative systems - output type

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

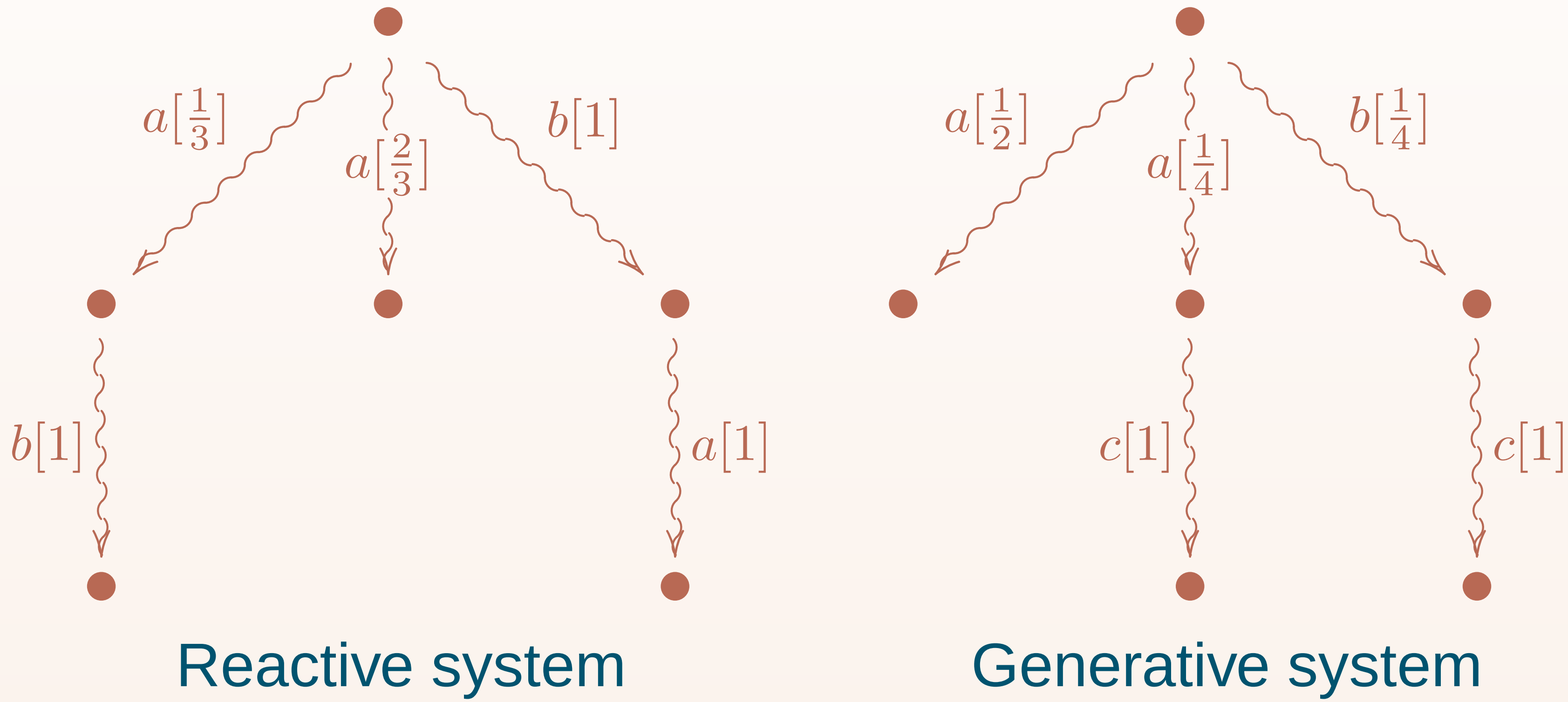
note:

In the probabilistic case

$(\mathcal{D}_\omega + 1)^A \ncong \mathcal{D}_\omega(A \times \mathcal{I}) + 1$

Reactive and generative systems

Example:



Bisimulation - generative systems

$\langle S, \alpha \rangle, \ \langle T, \beta \rangle$ - generative systems

$R \subseteq S \times T$ is a concrete bisimulation if for all $a \in A$, for all $\langle s, t \rangle \in R$ with $\mu = \alpha(s)$, $\nu = \beta(t)$ and every component C of R:

$$\mu(a, C) = \nu(a, C)$$

Bisimulation - generative systems

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - generative systems

$R \subseteq S \times T$ is a concrete bisimulation if for all $a \in A$, for all $\langle s, t \rangle \in R$ with $\mu = \alpha(s)$, $\nu = \beta(t)$ and every component C of R:

$$\mu(a, C) = \nu(a, C)$$

$$\sum_{s' \in \pi_1(C)} \mu(a, s') = \sum_{t' \in \pi_2(C)} \nu(a, t')$$

Bisimulation - generative systems

$\langle S, \alpha \rangle$, $\langle T, \beta \rangle$ - generative systems

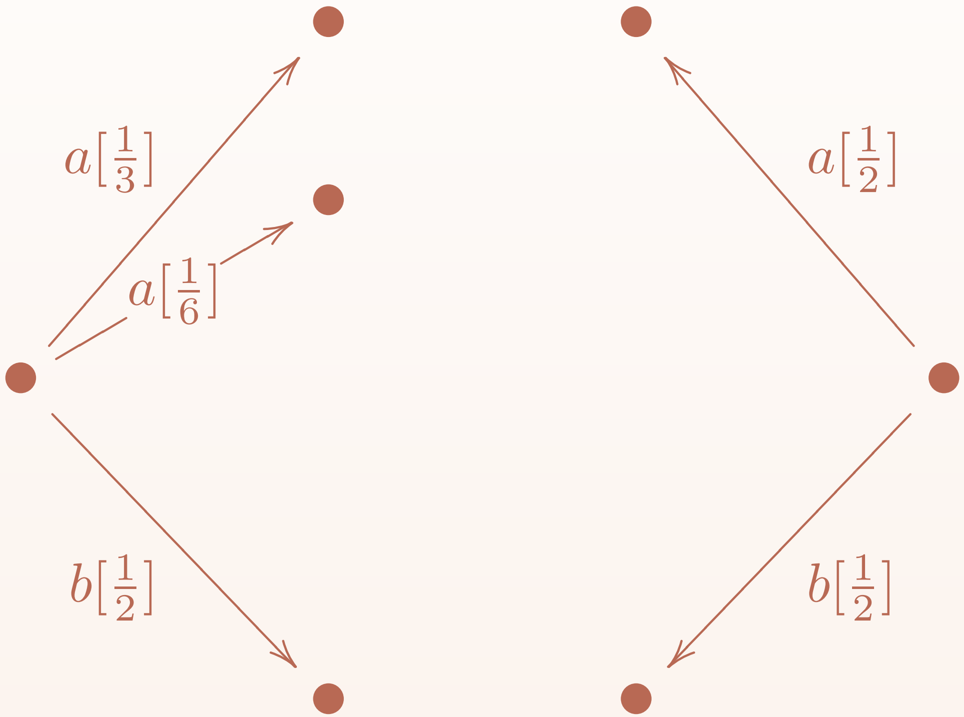
$R \subseteq S \times T$ is a concrete bisimulation if for all $a \in A$, for all $\langle s, t \rangle \in R$ with $\mu = \alpha(s)$, $\nu = \beta(t)$ and every component C of R:

$$\mu(a, C) = \nu(a, C)$$

$s \approx t$ - there exists a concrete bisimulation R with $\langle s, t \rangle \in R$

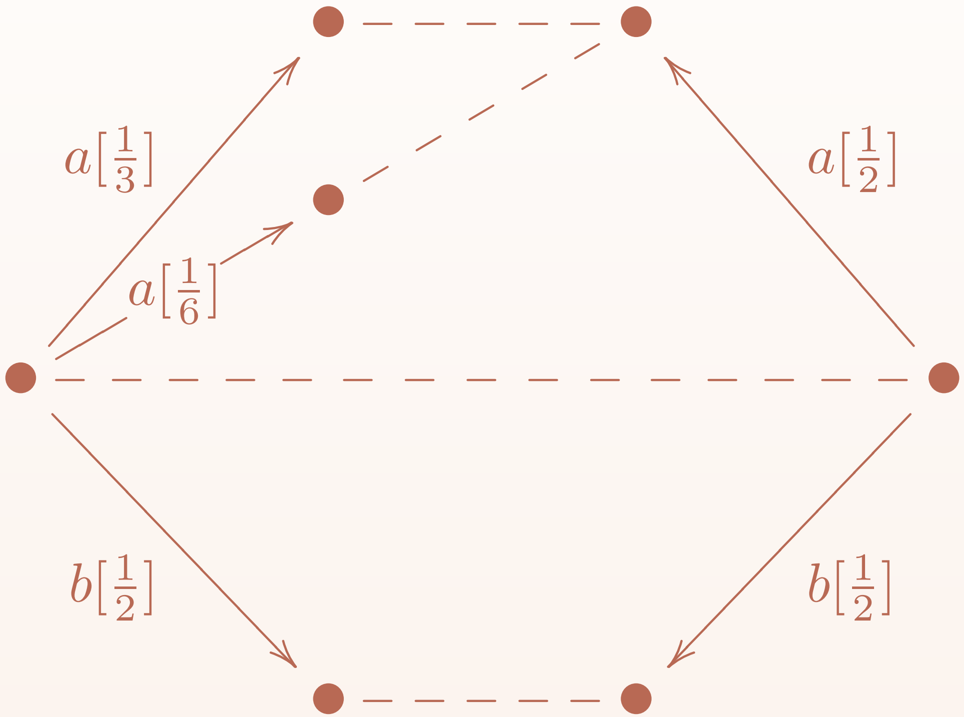
Bisimulation - generative systems

Example:



Bisimulation - generative systems

Example:



Bisimulation - generative systems

$\langle S, \alpha \rangle, \ \langle T, \beta \rangle$ - generative systems

$R \subseteq S \times T$ is a concrete bisimulation if for all $a \in A$, for all $\langle s, t \rangle \in R$ with $\mu = \alpha(s)$, $\nu = \beta(t)$ and every component C of R:

$$\mu(a, C) = \nu(a, C)$$

$s \approx t$ - there exists a concrete bisimulation R with $\langle s, t \rangle \in R$

Property:

$s \approx t$ if and only if $s \sim t$

Systems with distinction of states

Alternating systems - functor $\mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I})$

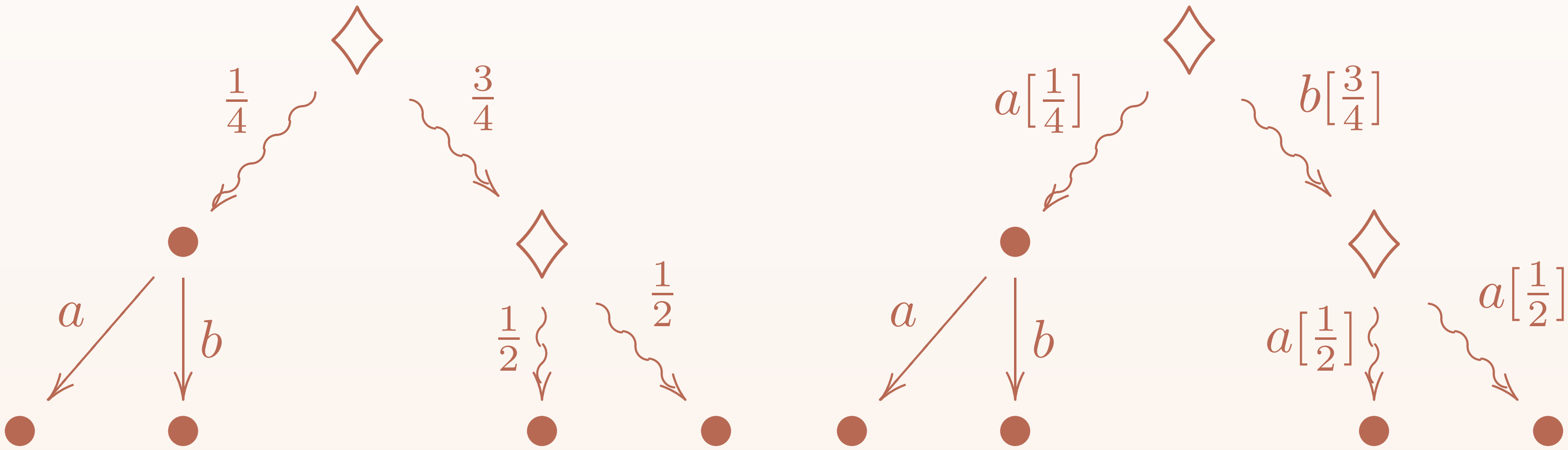
Vardi systems - functor $\mathcal{D}_\omega(A \times \mathcal{I}) + \mathcal{P}(A \times \mathcal{I})$

Systems with distinction of states

Alternating systems - functor $\mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I})$

Vardi systems - functor $\mathcal{D}_\omega(A \times \mathcal{I}) + \mathcal{P}(A \times \mathcal{I})$

Example:



Alternating system Vardi system

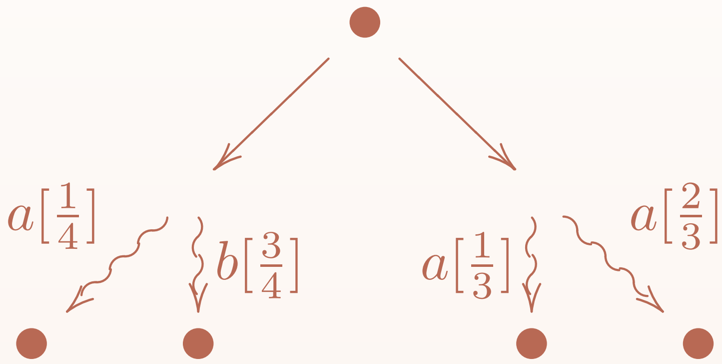
Structured transition function

Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

Structured transition function

Segala systems - functor $\mathcal{PD}_\omega(A \times \mathcal{I})$

Example:



Structured transition function

Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

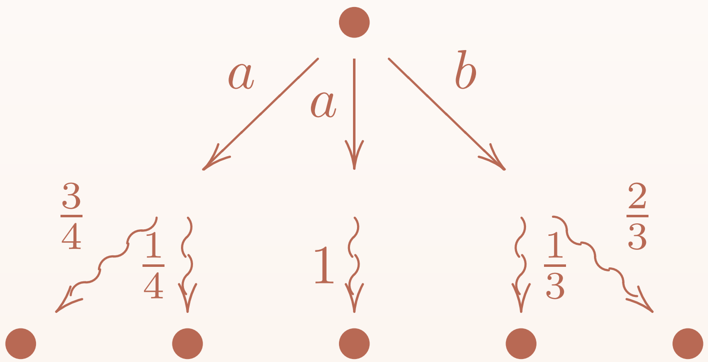
simple Segala systems - functor $\mathcal{P}(A \times \mathcal{D}_\omega)$

Structured transition function

Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

simple Segala systems - functor $\mathcal{P}(A \times \mathcal{D}_\omega)$

Example:



Structured transition function

Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

simple Segala systems - functor $\mathcal{P}(A \times \mathcal{D}_\omega)$

Structured transition function

Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

simple Segala systems - functor $\mathcal{P}(A \times \mathcal{D}_\omega)$

Bundle systems - functor $\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I})$

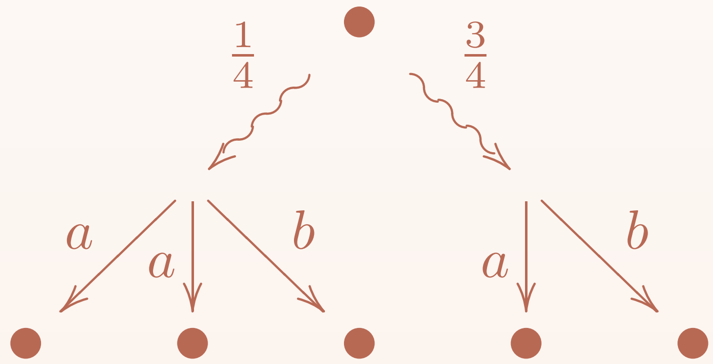
Structured transition function

Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

simple Segala systems - functor $\mathcal{P}(A \times \mathcal{D}_\omega)$

Bundle systems - functor $\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I})$

Example:



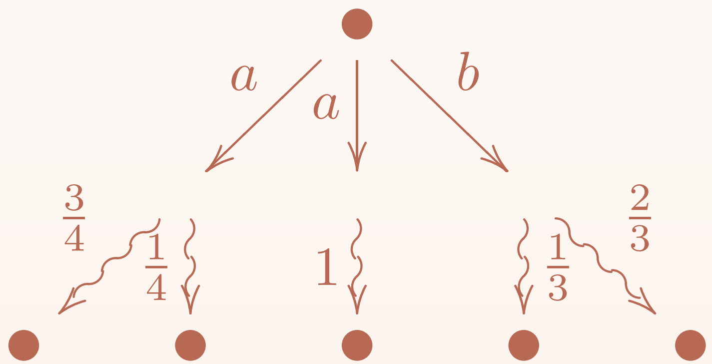
Structured transition function

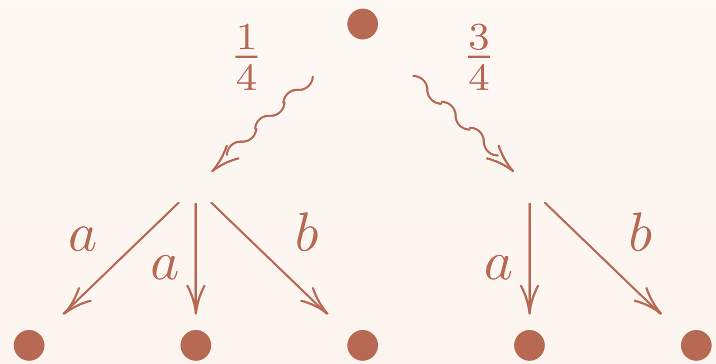
Segala systems - functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I})$

simple Segala systems - functor $\mathcal{P}(A \times \mathcal{D}_\omega)$

Bundle systems - functor $\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I})$

Simple Segala system $\quad\perp\quad$ Bundle system

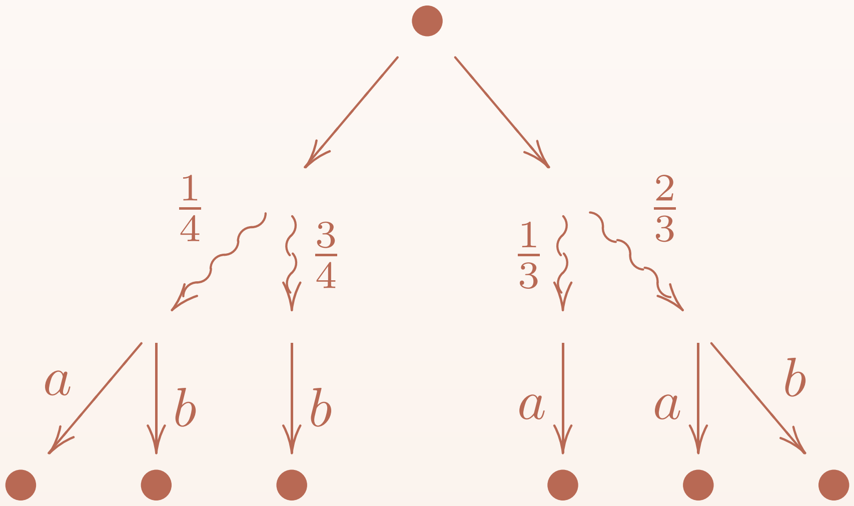




Complicated system types

Pnueli-Zuck systems - functor $\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I})$

Example:

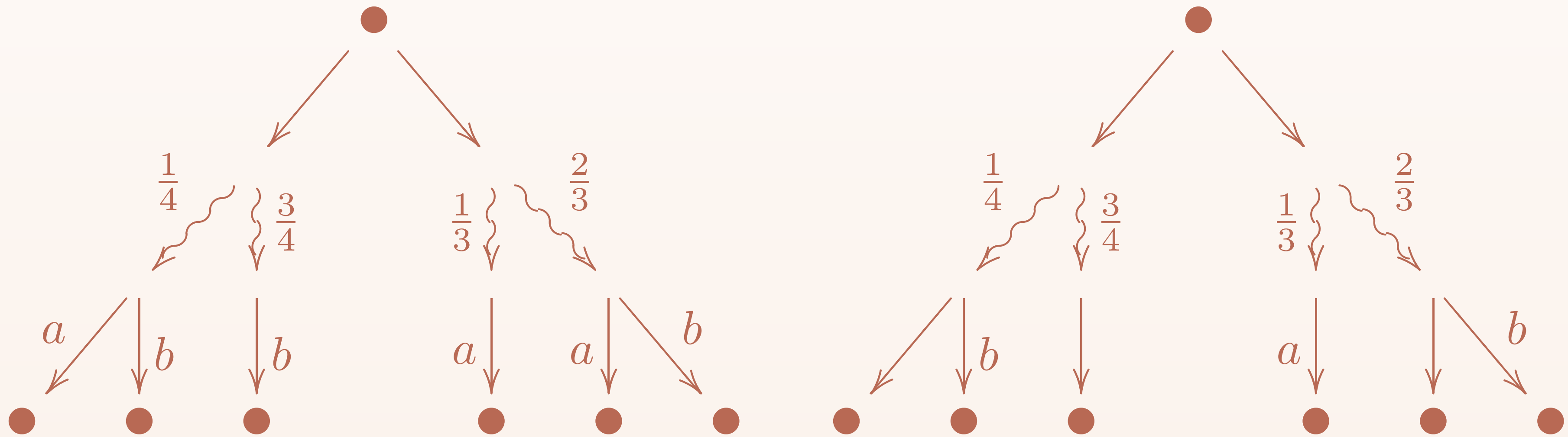


Complicated system types

Pnueli-Zuck systems - functor $\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I})$

most general systems - functor $\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}+\mathcal{I})$

Example:

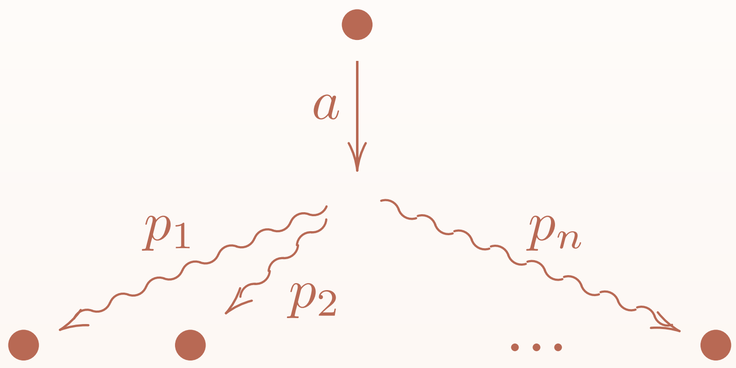


Pnueli-Zuck system

most general system

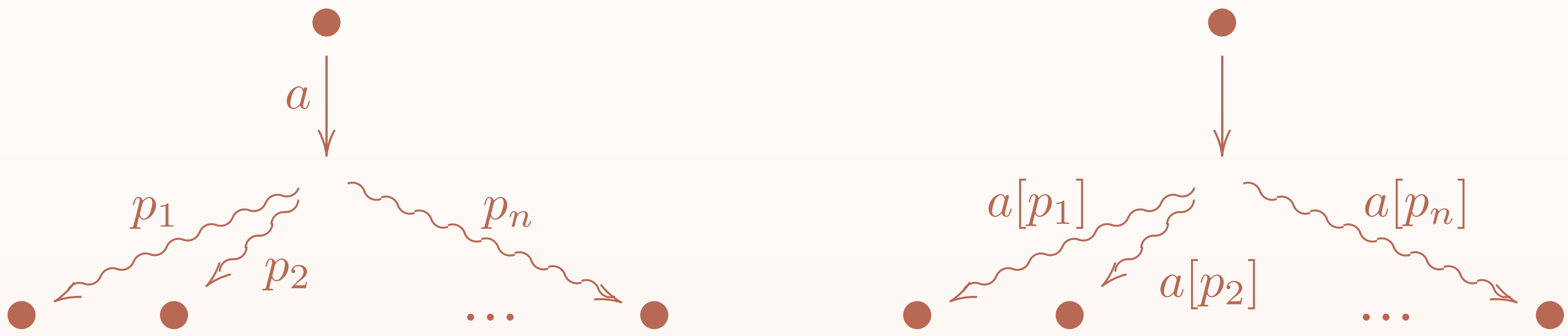
An intuitive translation

simple Segala system $\longrightarrow$ Segala system



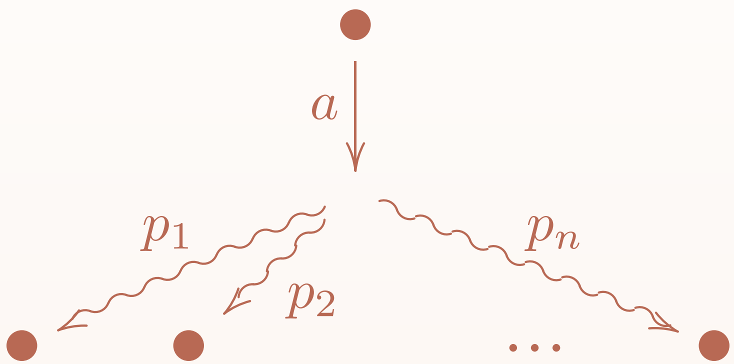
An intuitive translation

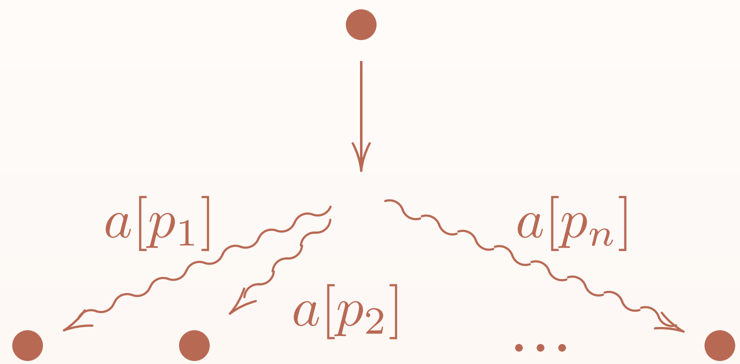
simple Segala system $\longrightarrow$ Segala system



An intuitive translation

simple Segala system $\longrightarrow$ Segala system





$$\langle S, \alpha \rangle \qquad \langle S, \alpha' \rangle$$

$$\alpha : S \to \mathcal{P}(A \times \mathcal{D}_\omega S) \qquad \alpha' : S \to \mathcal{P}\mathcal{D}_\omega(A \times S)$$

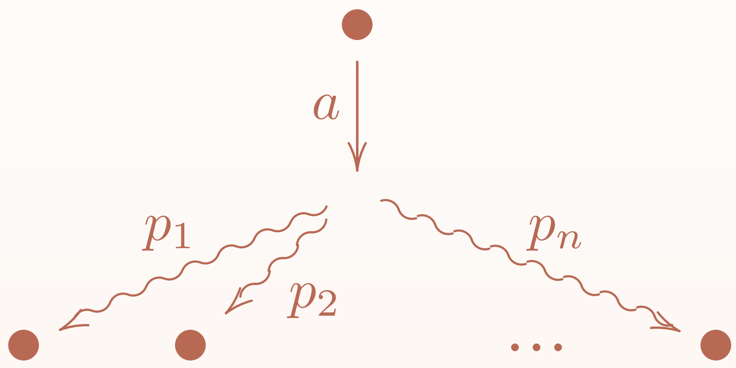
$$\alpha(s) = \{(a_i, \mu_i) \mid i \in I\} \qquad \alpha'(s) = \{\delta_{a_i} \cdot \mu_i \mid i \in I\}$$

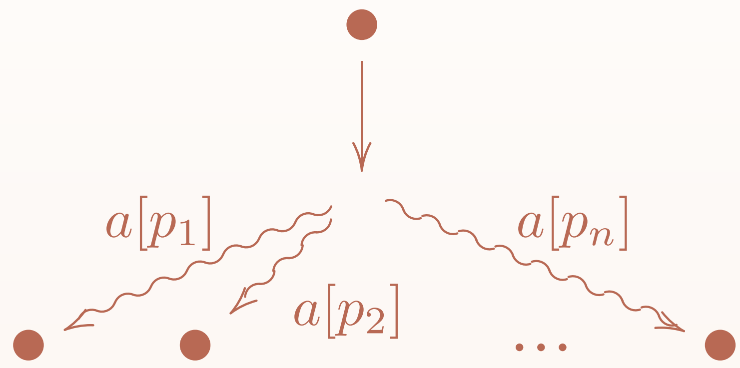
where $(\mu \cdot \mu')(x, x') = \mu(x) \cdot \mu'(x')$

and $\delta_a(b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{otherwise.} \end{cases}$

An intuitive translation

simple Segala system $\longrightarrow$ Segala system





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

Expressiveness

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

Expressiveness

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

Example:

LTS (functor: $\mathcal{P}(A \times \mathcal{I})$)
are clearly less expressive than
Alternating Systems (functor: $\mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I})$):

Any LTS can be viewed as an Alternating System that never uses the option to do a probabilistic step.

Expressiveness (2)

Our approach:

Systems of type $\mathcal{F}$ are at most as expressive as systems of type $\mathcal{G}$, if there is a mapping

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

with

$$\langle S, \alpha \rangle \overset{\mathcal{T}}{\longmapsto} \langle S, \tilde{\alpha} \rangle$$

that *preserves* and *reflects* bisimilarity:

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

Translation of coalgebras

Note that in the LTS vs. Alternating Systems example there exists a natural transformation

$$\iota_r : \mathcal{P}(A \times \mathcal{I}) \Rightarrow \mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}).$$

Translation of coalgebras

Note that in the LTS vs. Alternating Systems example there exists a natural transformation

$$\iota_r : \mathcal{P}(A \times \mathcal{I}) \Rightarrow \mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}).$$

Generally, a natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ gives rise to a translation of coalgebras $\mathcal{T}_\tau : \mathsf{Coalg}_{\mathcal{F}} \rightarrow \mathsf{Coalg}_{\mathcal{G}}$ as follows:

$$\begin{tikzcd} S \arrow[dd, "\alpha"'] & & S \arrow[d, "\alpha"] \\ & \arrow[r, phantom, "\overset{\mathcal{T}_\tau}{\Longrightarrow}"] & \mathcal{F}S \arrow[d, "\tau_S"] \\ \mathcal{F}S & & \mathcal{G}S \end{tikzcd}$$

Preservation of bisimulations

The translation $\mathcal{T}_\tau$ preserves bisimulations:

A bisimulation $R \subseteq S \times T$ between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$

$$\begin{tikzcd}
S \arrow[d, "\alpha"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] & T \arrow[d, "\beta"] \\
\mathcal{F}S & \mathcal{F}R \arrow[l, "\mathcal{F}\pi_1"'] \arrow[r, "\mathcal{F}\pi_2"] & \mathcal{F}T
\end{tikzcd}$$

Preservation of bisimulations

The translation $\mathcal{T}_\tau$ preserves bisimulations:

A bisimulation $R \subseteq S \times T$ between $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$

$$
\begin{tikzcd}
S \arrow[d, "\alpha"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] & T \arrow[d, "\beta"] \\
\mathcal{F}S \arrow[d, "\tau_S"'] \arrow[dr, phantom, "\text{nat. } \tau"] & \mathcal{F}R \arrow[l, "\mathcal{F}\pi_1"'] \arrow[r, "\mathcal{F}\pi_2"] \arrow[d, "\tau_R"] \arrow[dr, phantom, "\text{nat. } \tau"] & \mathcal{F}T \arrow[d, "\tau_T"] \\
\mathcal{G}S & \mathcal{G}R \arrow[l, "\mathcal{G}\pi_1"] \arrow[r, "\mathcal{G}\pi_2"'] & \mathcal{G}T
\end{tikzcd}
$$

is a bisimulation between $\mathcal{T}_\tau \langle S, \alpha \rangle$ and $\mathcal{T}_\tau \langle T, \beta \rangle$ as well.

Reflection of bisimilarity

But: $\mathcal{T}_\tau$ need not reflect bisimilarity.

Example:

Let τ be the natural transformation

$$\widetilde{\mathsf{supp}} : \mathcal{D}_\omega + 1 \Rightarrow \mathcal{P}$$

that *forgets* the probabilities.

Reflection of bisimilarity

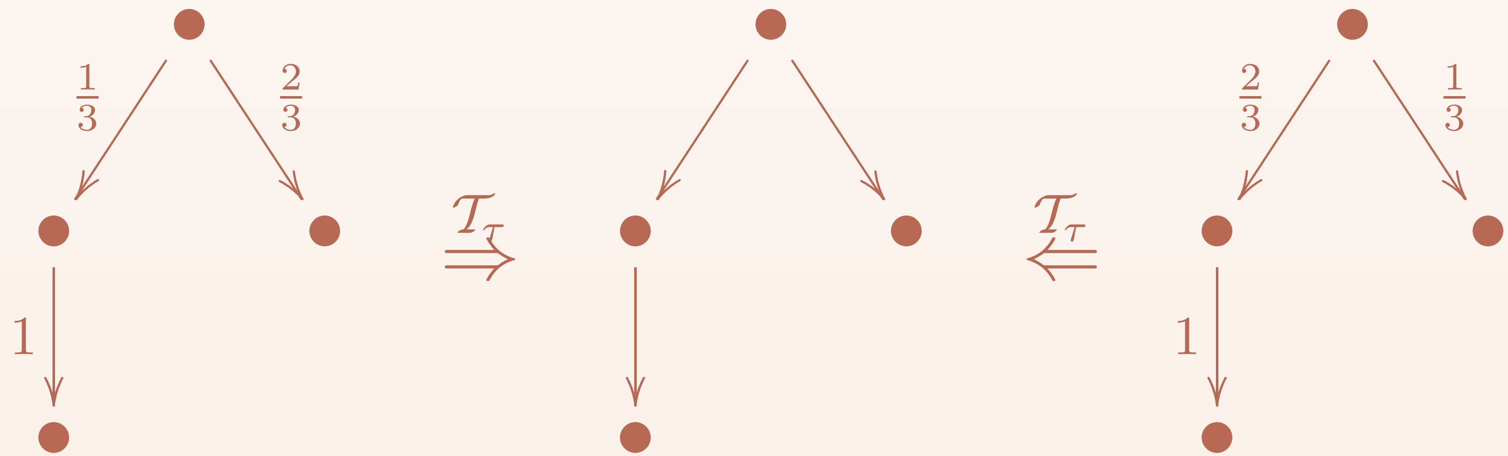
But: $\mathcal{T}_\tau$ need not reflect bisimilarity.

Example:

Let τ be the natural transformation

$$\widetilde{\mathsf{supp}} : \mathcal{D}_\omega + 1 \Rightarrow \mathcal{P}$$

that *forgets* the probabilities.



Assumption: Injectivity

Observation:
the components of $\widetilde{\text{supp}}$ are not injective.

Assumption: Injectivity

Observation:
the components of $\widetilde{\mathsf{supp}}$ are not injective.

$\Rightarrow$ to obtain reflection of bisimilarity we assume that τ is componentwise injective.

Assumption: Injectivity

Observation:
the components of $\widetilde{\mathsf{supp}}$ are not injective.

$\Rightarrow$ to obtain reflection of bisimilarity we assume that τ is componentwise injective.

Note that it is not a necessary condition.
Counter-example:
$\mathsf{supp} : \mathcal{D}_\omega \Rightarrow \mathcal{P}$

Assumption: Injectivity

Observation:
the components of $\widetilde{\mathsf{supp}}$ are not injective.

$\Rightarrow$ to obtain reflection of bisimilarity we assume that τ is componentwise injective.

Note that it is not a necessary condition.
Counter-example:
$\mathsf{supp} : \mathcal{D}_\omega \Rightarrow \mathcal{P}$

... although intuitive, sufficiency proof is not immediate...

Cocongruences

A *cocongruence* between two $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ is a *cospan*

$$\langle Q, q_1 : S \to Q, q_2 : T \to Q \rangle$$

such that there exists a $\mathcal{F}$-coalgebra structure γ on Q making the diagram below commute.

$$\begin{tikzcd} S \arrow[r, "q_1"] \arrow[d, "\alpha"'] & Q \arrow[d, "\gamma"] & T \arrow[l, "q_2"'] \arrow[d, "\beta"] \\ \mathcal{F}S \arrow[r, "\mathcal{F}q_1"'] & \mathcal{F}Q & \mathcal{F}T \arrow[l, "\mathcal{F}q_2"] \end{tikzcd}$$

Behavioural equivalence

Two states s and t in two $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ respectively are called *behavioural equivalent* if they are identified by some cocongruence.

Behavioural equivalence

Two states s and t in two $\mathcal{F}$-coalgebras $\langle S, \alpha \rangle$ and $\langle T, \beta \rangle$ respectively are called *behavioural equivalent* if they are identified by some cocongruence.

Result:
If all components of the natural transformation

$$\tau : \mathcal{F} \Rightarrow \mathcal{G}$$

are injective, then $\mathcal{T}_\tau$ reflects behavioural equivalence.

Bisimularity vs. beh. equivalence

- Generally, bisimilarity implies behavioural equivalence.

Bisimularity vs. beh. equivalence

- Generally, bisimilarity implies behavioural equivalence.
- If the functor $\mathcal{F}$ preserves weak pullbacks, then behavioural equivalence implies bisimilarity.
  $\Rightarrow$ both notions coincide.

Bisimularity vs. beh. equivalence

- Generally, bisimilarity implies behavioural equivalence.
- If the functor $\mathcal{F}$ preserves weak pullbacks, then behavioural equivalence implies bisimilarity.
  $\Rightarrow$ both notions coincide.

Corollary:
If $\mathcal{F}$ preserves weak pullbacks and all components of $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ are injective, then $\mathcal{T}_\tau$ reflects bisimilarity.

Need of w.p. preservation

The corollary is not valid without assuming that $\mathcal{F}$ preserves weak pullbacks.

Counter-example:
Consider the functors

$$\mathcal{F}X := \left\{ \langle x, y, z \rangle \in X^3 \;\middle|\; |\{x, y, z\}| \leq 2 \right\}$$

and

$$\mathcal{G}X := X^3$$

and let $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ be the set inclusion.

Weak pullback preservation here

- The functors A, $\mathcal{I}$, $\mathcal{P}$, and $\mathcal{D}_\omega$ preserve weak pullbacks.

Weak pullback preservation here

- The functors $\mathbb{A}$, $\mathcal{I}$, $\mathcal{P}$, and $\mathcal{D}_\omega$ preserve weak pullbacks.
- If $\mathcal{F}$ and $\mathcal{G}$ preserve weak pullbacks, so do $\mathcal{F}+\mathcal{G}$, $\mathcal{F}\times\mathcal{G}$, $\mathcal{F}^C$, and $\mathcal{F}\mathcal{G}$.

Weak pullback preservation here

- The functors A, $\mathcal{I}$, $\mathcal{P}$, and $\mathcal{D}_\omega$ preserve weak pullbacks.
- If $\mathcal{F}$ and $\mathcal{G}$ preserve weak pullbacks, so do $\mathcal{F} + \mathcal{G}$, $\mathcal{F} \times \mathcal{G}$, $\mathcal{F}^C$, and $\mathcal{F}\mathcal{G}$.

$\Rightarrow$ All functors used to define the different probabilistic system types preserve weak pullbacks.

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,
- $\sigma : \mathcal{I} \Rightarrow \mathcal{P}$ with $\sigma_X(x) := \{x\}$,

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,
- $\sigma : \mathcal{I} \Rightarrow \mathcal{P}$ with $\sigma_X(x) := \{x\}$,
- $\delta : \mathcal{I} \Rightarrow \mathcal{D}_\omega$ with $\delta_X(x) := \delta_x$ (*Dirac*),

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,
- $\sigma : \mathcal{I} \Rightarrow \mathcal{P}$ with $\sigma_X(x) := \{x\}$,
- $\delta : \mathcal{I} \Rightarrow \mathcal{D}_\omega$ with $\delta_X(x) := \delta_x$ (*Dirac*),
- $\iota_l : \mathcal{F} \Rightarrow \mathcal{F} + \mathcal{G}$ and $\iota_r : \mathcal{G} \Rightarrow \mathcal{F} + \mathcal{G}$,

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,
- $\sigma : \mathcal{I} \Rightarrow \mathcal{P}$ with $\sigma_X(x) := \{x\}$,
- $\delta : \mathcal{I} \Rightarrow \mathcal{D}_\omega$ with $\delta_X(x) := \delta_x$ (*Dirac*),
- $\iota_l : \mathcal{F} \Rightarrow \mathcal{F} + \mathcal{G}$ and $\iota_r : \mathcal{G} \Rightarrow \mathcal{F} + \mathcal{G}$,
- $\phi + \psi : \mathcal{F} + \mathcal{G} \Rightarrow \mathcal{F}' + \mathcal{G}'$ for $\phi : \mathcal{F} \Rightarrow \mathcal{F}'$ and $\psi : \mathcal{G} \Rightarrow \mathcal{G}'$ (both with i.c.),

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,
- $\sigma : \mathcal{I} \Rightarrow \mathcal{P}$ with $\sigma_X(x) := \{x\}$,
- $\delta : \mathcal{I} \Rightarrow \mathcal{D}_\omega$ with $\delta_X(x) := \delta_x$ (*Dirac*),
- $\iota_l : \mathcal{F} \Rightarrow \mathcal{F} + \mathcal{G}$ and $\iota_r : \mathcal{G} \Rightarrow \mathcal{F} + \mathcal{G}$,
- $\phi + \psi : \mathcal{F} + \mathcal{G} \Rightarrow \mathcal{F}' + \mathcal{G}'$ for $\phi : \mathcal{F} \Rightarrow \mathcal{F}'$ and $\psi : \mathcal{G} \Rightarrow \mathcal{G}'$ (both with i.c.),
- $\kappa : \mathrm{A} \times \mathcal{P} \Rightarrow \mathcal{P}(\mathrm{A} \times \mathcal{I})$ with
$$\kappa_X(a, M) := \{\langle a, x\rangle \mid x \in M\},$$

Some basic transformations

Examples of natural transformations with injective components:

- $\eta : 1 \Rightarrow \mathcal{P}$ with $\eta_X(*) := \emptyset$,
- $\sigma : \mathcal{I} \Rightarrow \mathcal{P}$ with $\sigma_X(x) := \{x\}$,
- $\delta : \mathcal{I} \Rightarrow \mathcal{D}_\omega$ with $\delta_X(x) := \delta_x$ (*Dirac*),
- $\iota_l : \mathcal{F} \Rightarrow \mathcal{F} + \mathcal{G}$ and $\iota_r : \mathcal{G} \Rightarrow \mathcal{F} + \mathcal{G}$,
- $\phi + \psi : \mathcal{F} + \mathcal{G} \Rightarrow \mathcal{F}' + \mathcal{G}'$ for
  $\phi : \mathcal{F} \Rightarrow \mathcal{F}'$ and $\psi : \mathcal{G} \Rightarrow \mathcal{G}'$ (both with i.c.),
- $\kappa : \mathrm{A} \times \mathcal{P} \Rightarrow \mathcal{P}(\mathrm{A} \times \mathcal{I})$ with
  $$\kappa_X(a, M) := \{\langle a, x\rangle \mid x \in M\},$$
- $\ldots$

One expressiveness statement

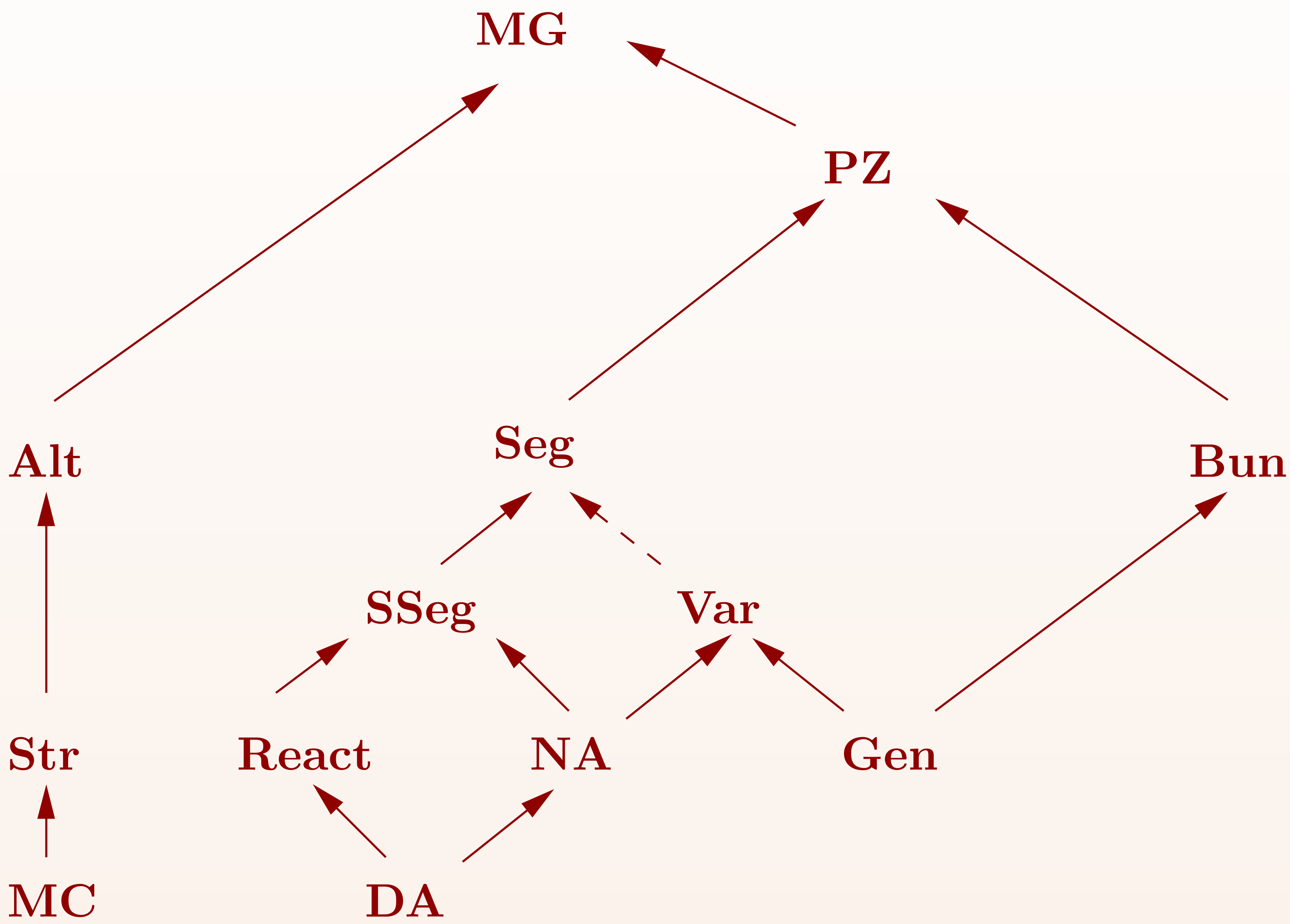
To show for instance that
Generative systems
(functor: $\mathcal{F} := \mathcal{D}_\omega(\mathrm{A} \times \mathcal{I}) + 1$)

are at most as expressive as
Vardi systems
(functor: $\mathcal{G} := \mathcal{D}_\omega(\mathrm{A} \times \mathcal{I}) + \mathcal{P}(\mathrm{A} \times \mathcal{I})$)

we employ the natural transformation

$$\mathcal{D}_\omega(\mathrm{A} \times \mathcal{I}) + \eta(\mathrm{A} \times \mathcal{I}) : \mathcal{F} \Rightarrow \mathcal{G}.$$

The hierarchy of system types



Conclusion

- Various probabilistic system types were compared

Conclusion

- Various probabilistic system types were compared
- The coalgebraic approach proved useful for:

Conclusion

- Various probabilistic system types were compared
- The coalgebraic approach proved useful for:
  - providing a uniform framework

Conclusion

- Various probabilistic system types were compared
- The coalgebraic approach proved useful for:
  * providing a uniform framework
  * a general notion of bisimulation

Conclusion

- Various probabilistic system types were compared
- The coalgebraic approach proved useful for:
  * providing a uniform framework
  * a general notion of bisimulation
  * proving a comparison result