Probabilistic models for verification

Ana Sokolova

SOS group, Radboud University Nijmegen

Outline

- formal methods and verification
- probabilistic systems (coalgebras)
- bisimilarity - the strong end of the spectrum
- expressiveness hierarchy
- other semantics - at the weak end of the spectrum

Why formal methods?

- Every mature engineering discipline features

Why formal methods?

- Every mature engineering discipline features
  - abstraction

Why formal methods?

- Every mature engineering discipline features
  - abstraction
  - analysis

Why formal methods?

- Every mature engineering discipline features
  - abstraction
  - analysis
- In hardware and software design

Why formal methods?

- Every mature engineering discipline features
  - abstraction
  - analysis
- In hardware and software design
  - trial and error

Why formal methods?

- Every mature engineering discipline features
  - abstraction
  - analysis
- In hardware and software design
  - trial and error
  - duplication

Why formal methods?

- Every mature engineering discipline features
  - abstraction
  - analysis
- In hardware and software design
  - trial and error
  - duplication
- Formal methods aim at replacing the ♣ with the ♣

Formal methods

are mathematically based techniques for

- specification
- development
- verification

of software and hardware systems

Formal methods

In general:

- models - transition systems, automata, terms,...
  with a clear semantics
- analysis - model checking
  process algebra
  theorem proving...

Formal methods

Here, now:

- models - transition systems
- semantics - behavior equivalences

Formal methods

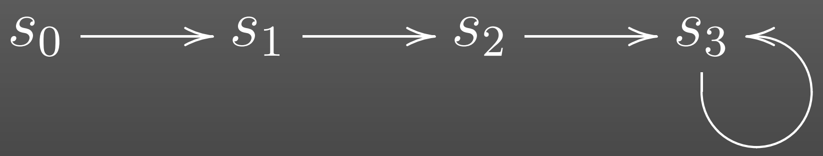
Here, now:

- models - transition systems
- semantics - behavior equivalences

Aim: one framework for many (probabilistic) models and semantics - compare expressiveness

Example models

deterministic systems



Example models

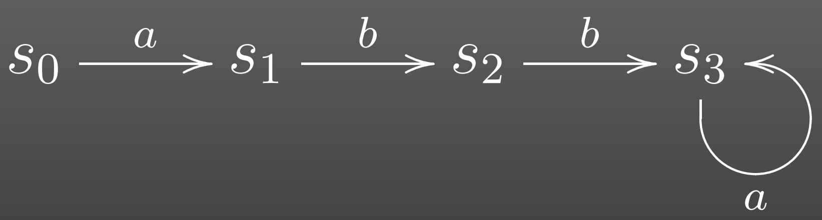
deterministic systems

states $+$ transitions $\alpha : S \to S$

$$\alpha(s_0) = s_1,\ \alpha(s_1) = s_2,\ \dots$$

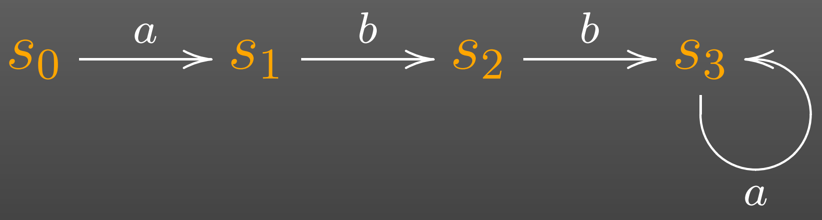
Example models

labelled deterministic systems A - labels



Example models

labelled deterministic systems A - labels

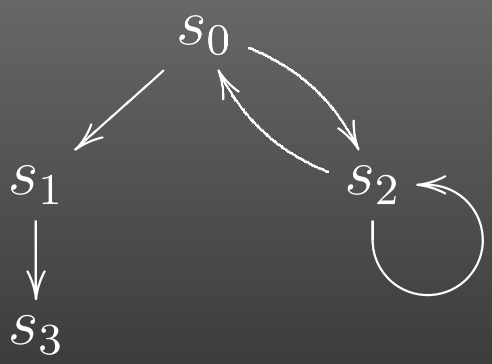


states $+$ transitions $\alpha : S \to A \times S$

$$\alpha(s_0) = \langle a, s_1 \rangle,\ \alpha(s_1) = \langle b, s_2 \rangle,\ \ldots$$

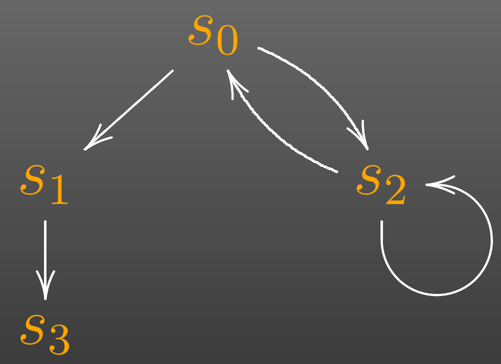
Example models

transition systems



Example models

transition systems

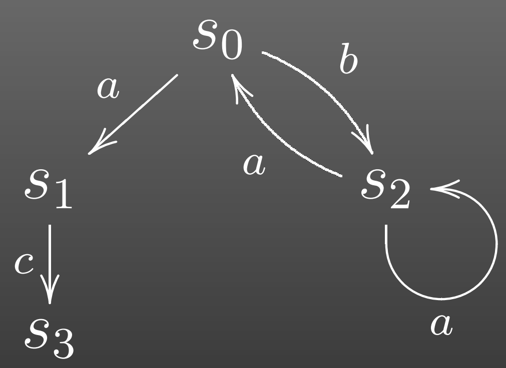


states $+$ transitions $\alpha : S \to \mathcal{P}(S)$

$$\alpha(s_0) = \{s_1, s_2\},\ \alpha(s_1) = \{s_3\},\ \ldots$$

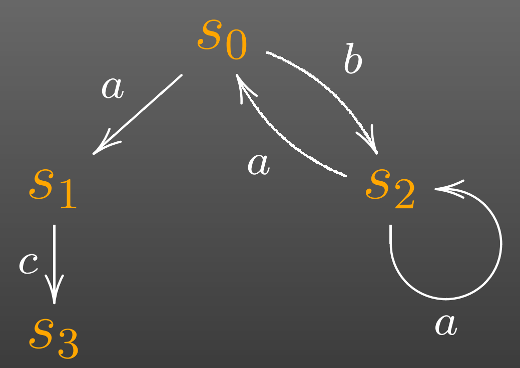
Example models

labelled transition systems A - labels



Example models

labelled transition systems A - labels



states + transitions $\alpha : S \to \mathcal{P}(A \times S)$

$$\alpha(s_0) = \{\langle a, s_1\rangle, \langle b, s_2\rangle\},\ \alpha(s_1) = \{\langle c, s_3\rangle\},\ \ldots$$

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

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

- rich mathematical structure
- a uniform way for treating transition systems
- general notions and results, generic notion of bisimulation

Probabilistic systems

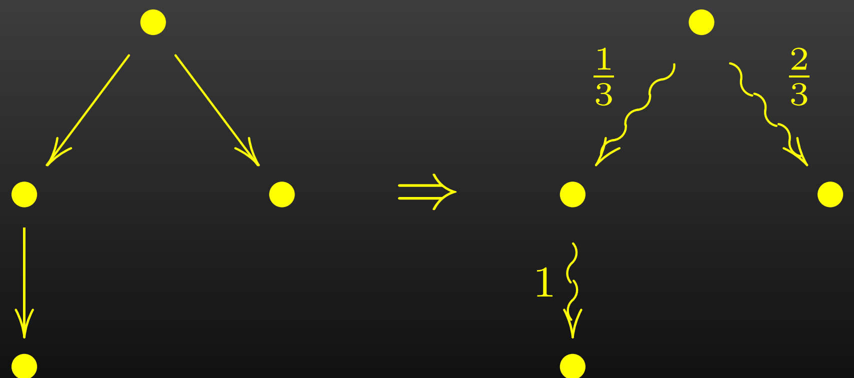
arise by enriching transition systems with (discrete) probabilities as labels on the transitions.

Examples:

Probabilistic systems

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

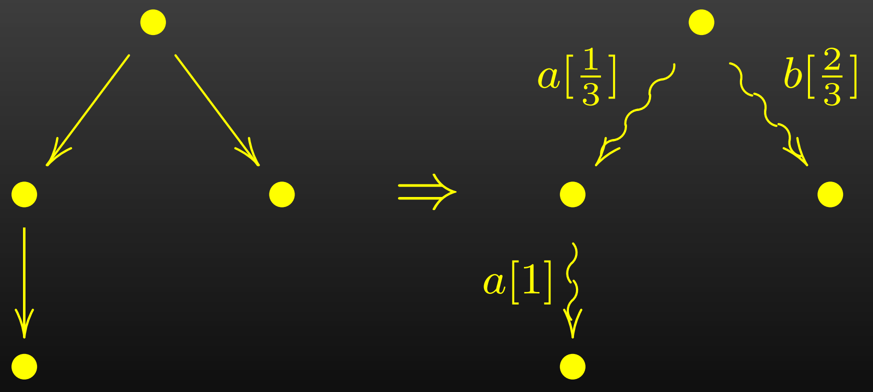
Examples:



Probabilistic systems

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

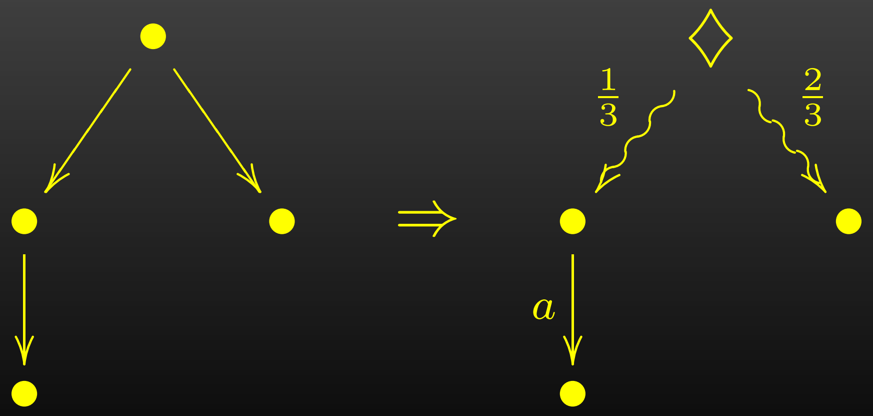
Examples:



Probabilistic systems

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

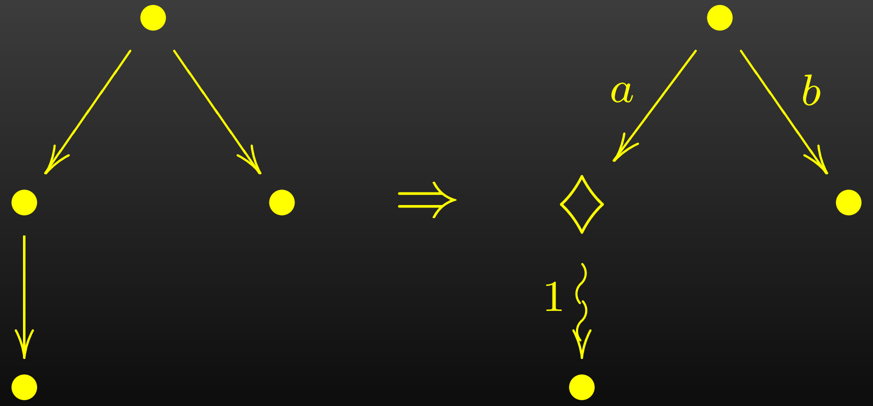
Examples:



Probabilistic systems

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

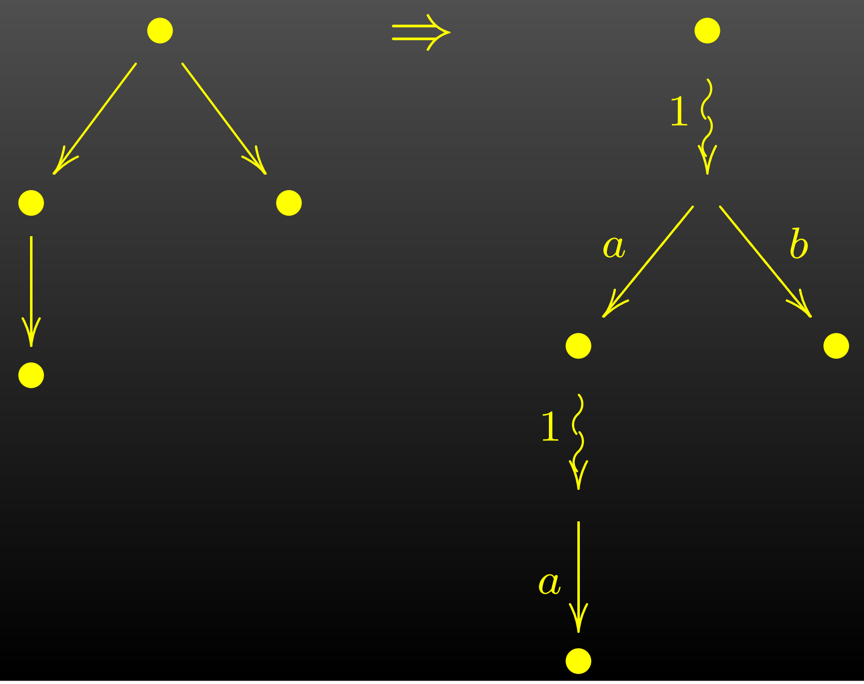
Examples:



Probabilistic systems

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

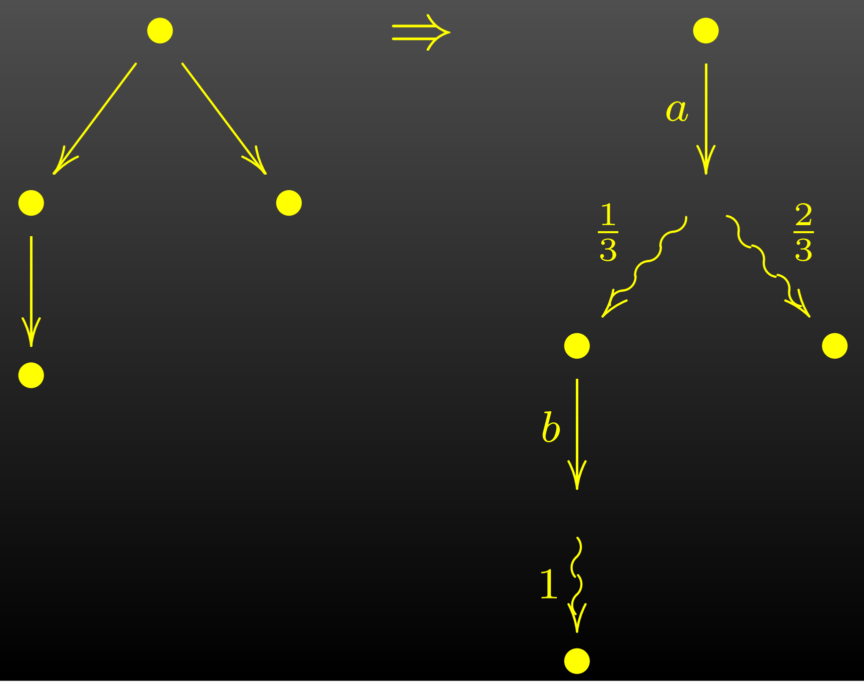
Examples:



Probabilistic systems

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

Examples:



Probabilistic systems

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

Probabilistic systems

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

$\mathcal{D}S =$ the set of all discrete probability distributions on S

Probabilistic systems

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

$$\mathcal{D}S = \begin{array}{l}\text{the set of all discrete}\\ \text{probability distributions on } S\end{array}$$

the probabilistic systems are also coalgebras

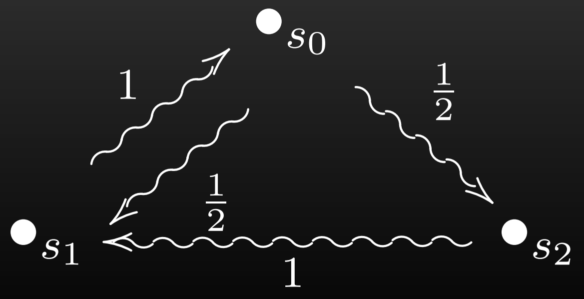
Probabilistic systems

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

$$\mathcal{D}S = \text{the set of all discrete probability distributions on } S$$

the probabilistic systems are also coalgebras

Example: $\alpha : S \to \mathcal{D}S$



Probabilistic systems

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

$$\mathcal{D}S = \begin{array}{l}\text{the set of all discrete} \\ \text{probability distributions on } S\end{array}$$

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

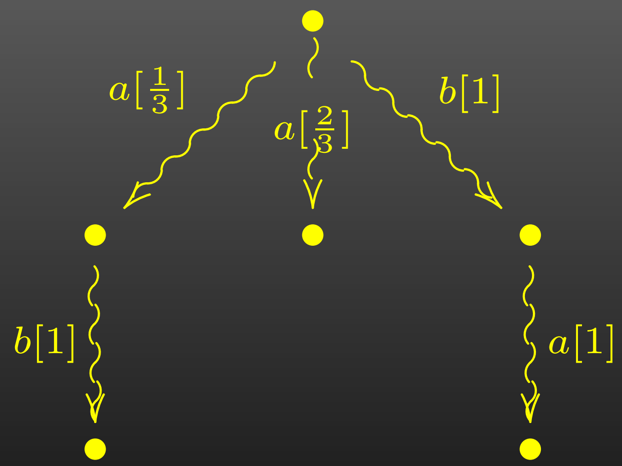
$$\mathcal{F} ::= \_ \mid A \mid \mathcal{P} \mid \mathcal{D} \mid \mathcal{G} + \mathcal{H} \mid \mathcal{G} \times \mathcal{H} \mid \mathcal{G}^A \mid \mathcal{G} \circ \mathcal{H}$$

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	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...

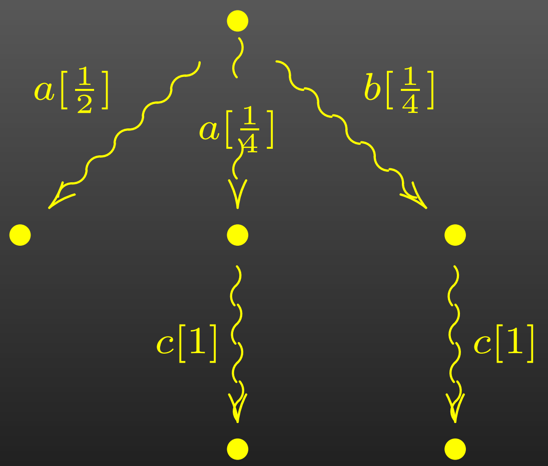
Probabilistic system types

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	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



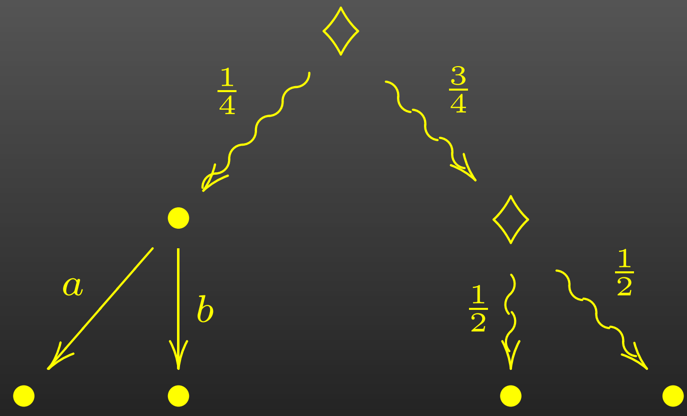
Probabilistic system types

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	$\mathcal{PD}(A \times \_)$
...	...



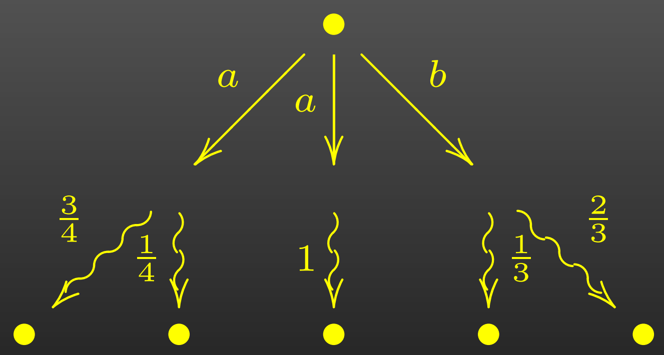
Probabilistic system types

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	$\mathcal{P}\mathcal{D}(A \times \_)$
...	...



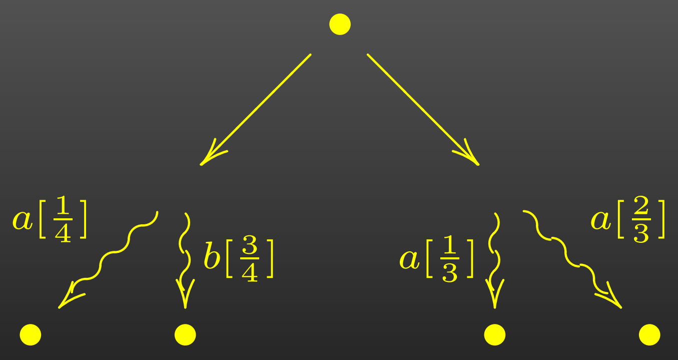
Probabilistic system types

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

R - equivalence on states, is a bisimulation if

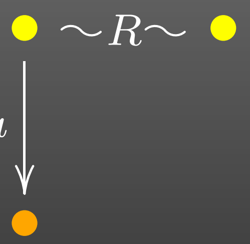
Bisimulation - LTS

R - equivalence on states, is a bisimulation if

- $\sim R \sim$

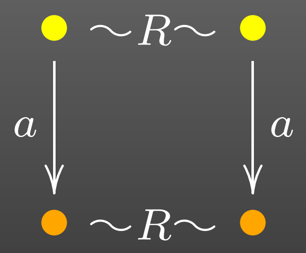
Bisimulation - LTS

R - equivalence on states, is a bisimulation if



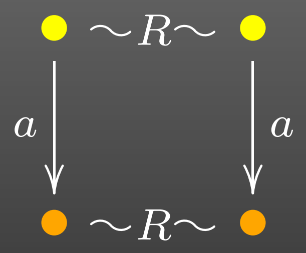
Bisimulation - LTS

R - equivalence on states, is a bisimulation if



Bisimulation - LTS

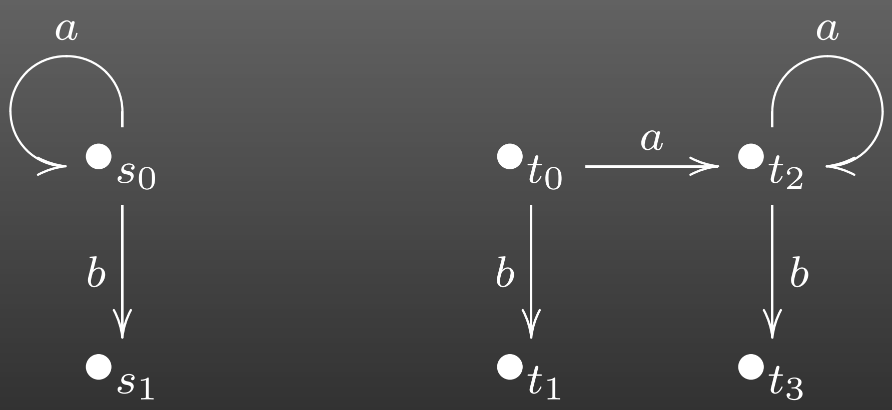
R - equivalence on states, is a bisimulation if



... two states are bisimilar if they are related by some bisimulation

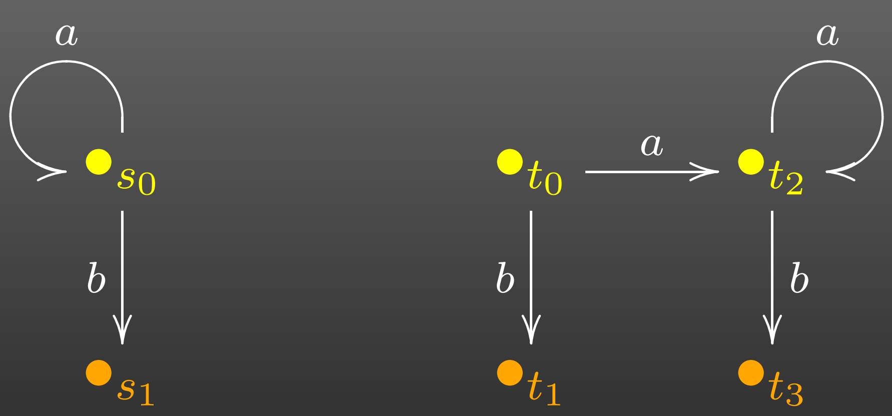
Bisimulation - LTS

Example: Consider the LTS



Bisimulation - LTS

Example: Consider the LTS



the coloring is a bisimulation, so s_0 and t_0 are bisimilar

Bisimulation - generative

R - equivalence on states, is a bisimulation if

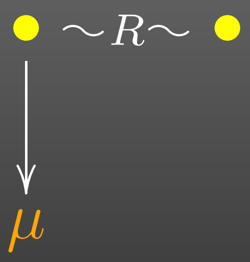
Bisimulation - generative

R - equivalence on states, is a bisimulation if

$$\bullet \sim R \sim \bullet$$

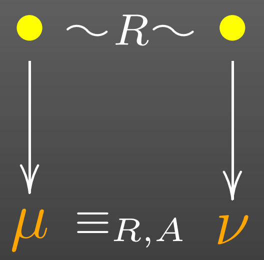
Bisimulation - generative

R - equivalence on states, is a bisimulation if



Bisimulation - generative

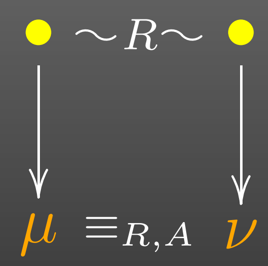
R - equivalence on states, is a bisimulation if



$\equiv_{R,A}$ relates distributions that assign the same probability to each label and each R-class

Bisimulation - generative

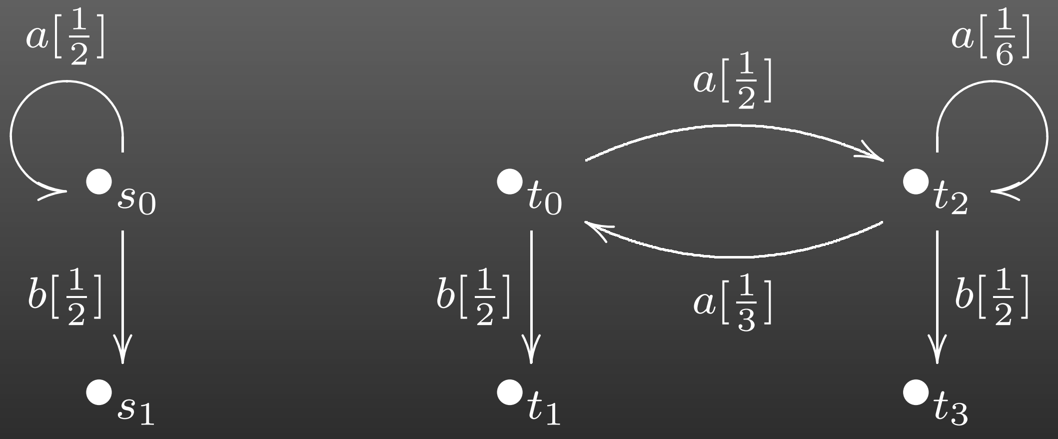
R - equivalence on states, is a bisimulation if



... two states are bisimilar if they are related by some bisimulation

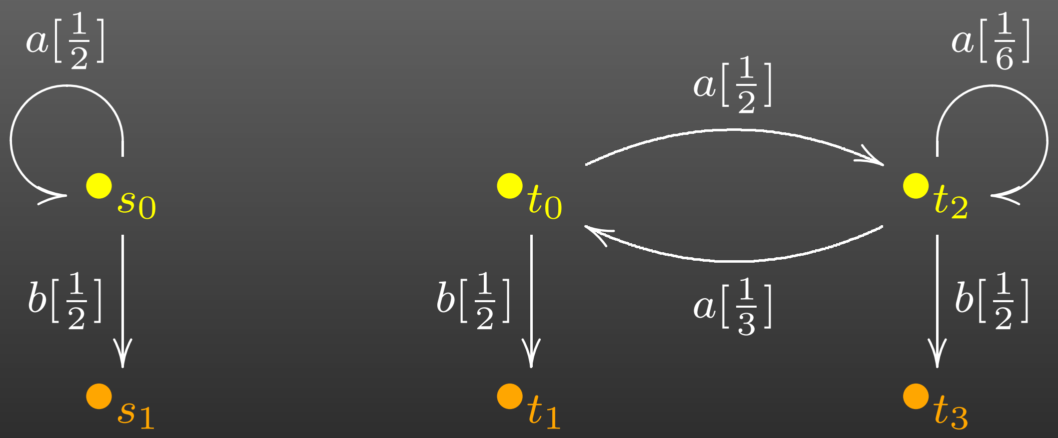
Bisimulation - generative

Consider the generative systems



Bisimulation - generative

Example: Consider the generative systems



the coloring is a bisimulation, so s_0 and t_0 are bisimilar

Bisimulation - simple Segala

R - equivalence on states, is a bisimulation if

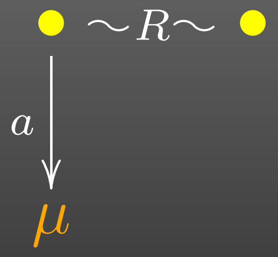
Bisimulation - simple Segala

R - equivalence on states, is a bisimulation if

$$\bullet \sim R \sim \bullet$$

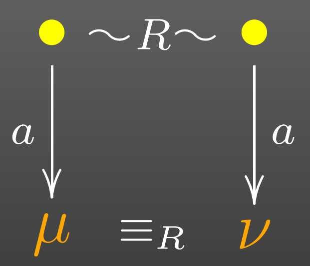
Bisimulation - simple Segala

R - equivalence on states, is a bisimulation if



Bisimulation - simple Segala

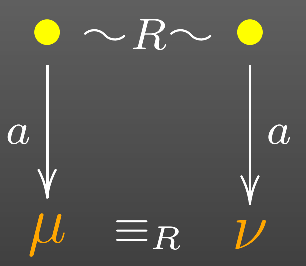
R - equivalence on states, is a bisimulation if



$\equiv_R$ relates distributions that assign the same probability to each R-class

Bisimulation - simple Segala

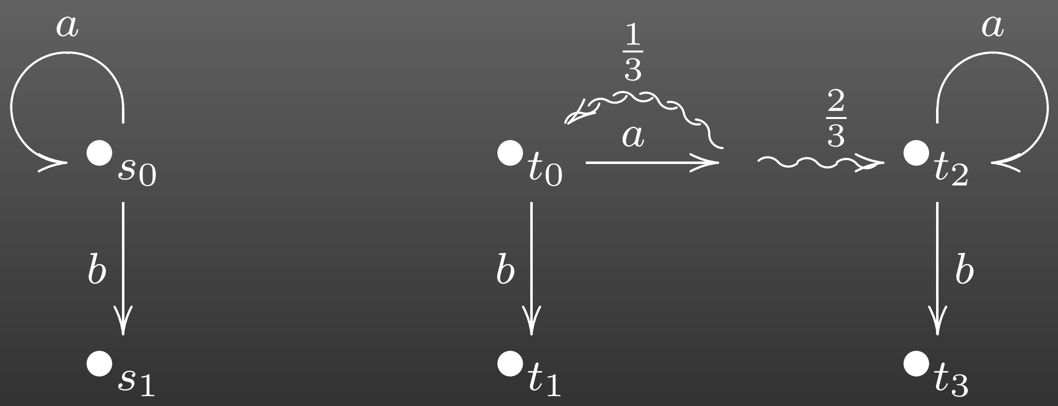
R - equivalence on states, is a bisimulation if



... two states are bisimilar if they are related by some bisimulation

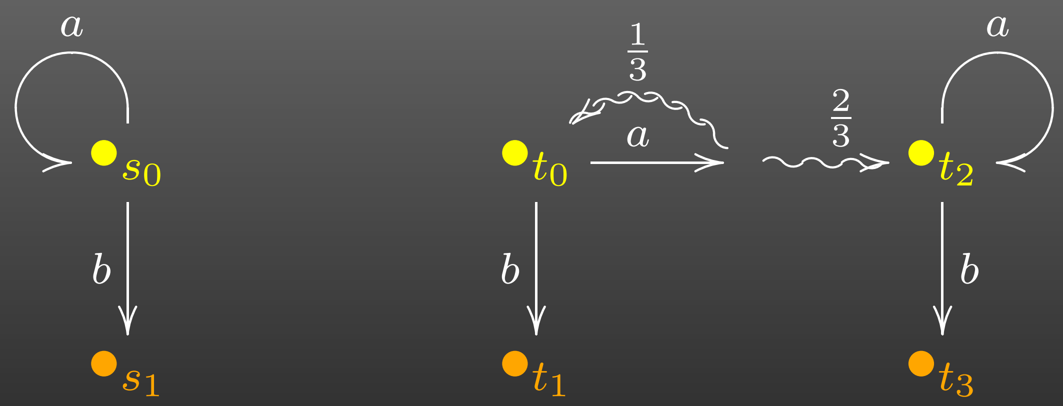
Bisimulation - simple Segala

Example: Consider the simple Segala systems



Bisimulation - simple Segala

Example: Consider the simple Segala systems



the coloring is a bisimulation, so s_0 and t_0 are bisimilar

Coalgebraic bisimulation

A bisimulation on

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

is $R \subseteq S \times S$ such that

Coalgebraic bisimulation

A bisimulation on

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

is $R \subseteq S \times S$ such that γ exists:

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

Coalgebraic bisimulation

A bisimulation on

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

is $R \subseteq S \times S$ such that

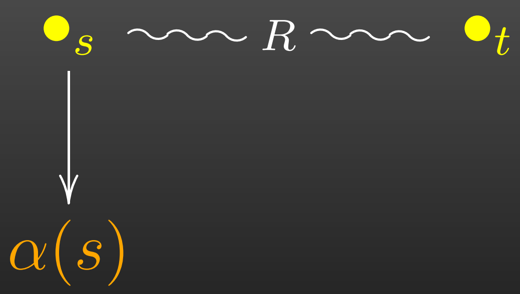
$$\bullet_s \rightsquigarrow R \rightsquigarrow \bullet_t$$

Coalgebraic bisimulation

A bisimulation on

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

is $R \subseteq S \times S$ such that

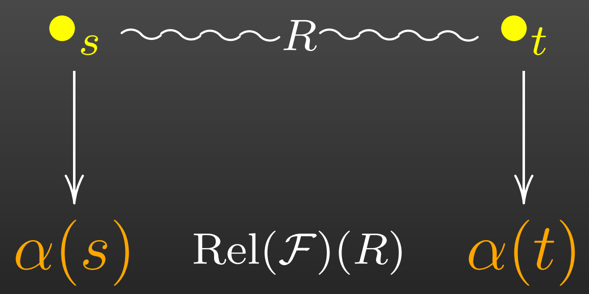


Coalgebraic bisimulation

A bisimulation on

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

is $R \subseteq S \times S$ such that

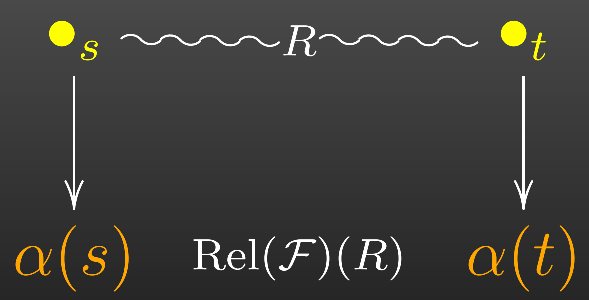


Coalgebraic bisimulation

A bisimulation on

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

is $R \subseteq S \times S$ such that



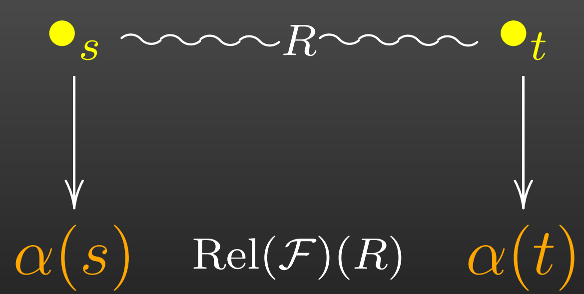
... two states are bisimilar if they are related by some bisimulation

Coalgebraic bisimulation

A bisimulation on

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

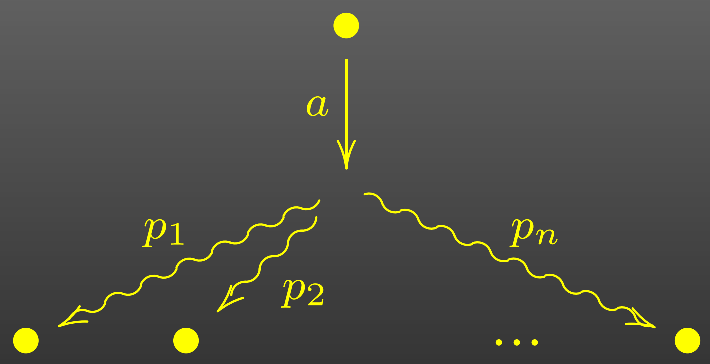
is $R \subseteq S \times S$ such that



Theorem: Coalgebraic and concrete bisimilarity coincide !

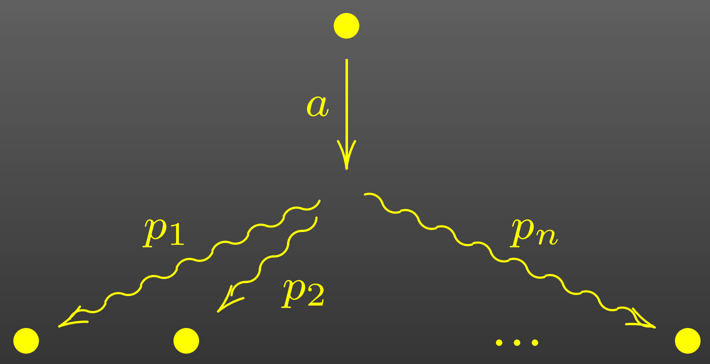
Expressiveness

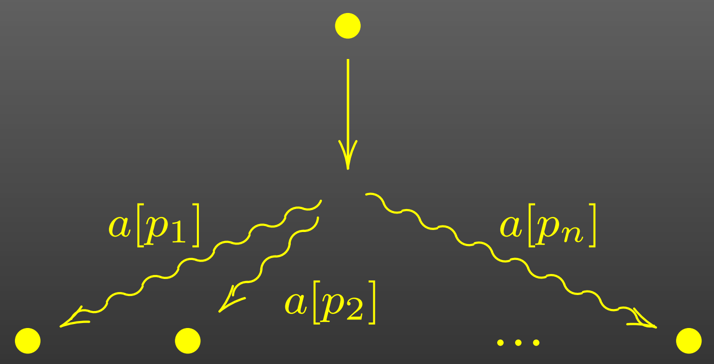
simple Segala system $\longrightarrow$ Segala system



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

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

Comparison criterion

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

if there is a way to map each $\mathcal{F}$-coalgebra to a $\mathcal{G}$-coalgebra with the same states such that

Comparison criterion

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

if there is a way to map each $\mathcal{F}$-coalgebra to a $\mathcal{G}$-coalgebra with the same states such that

bisimilarity is preserved and reflected

Comparison criterion

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

if there is a way to map each $\mathcal{F}$-coalgebra to a $\mathcal{G}$-coalgebra with the same states such that

bisimilarity is preserved and reflected

states are bisimilar in the original system iff they are bisimilar in the translation

Comparison criterion

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

if there is a way to map each $\mathcal{F}$-coalgebra to a $\mathcal{G}$-coalgebra with the same states such that

bisimilarity is preserved and reflected

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

Comparison criterion

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

if there is a way to map each $\mathcal{F}$-coalgebra to a $\mathcal{G}$-coalgebra with the same states such that

bisimilarity is preserved and reflected

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

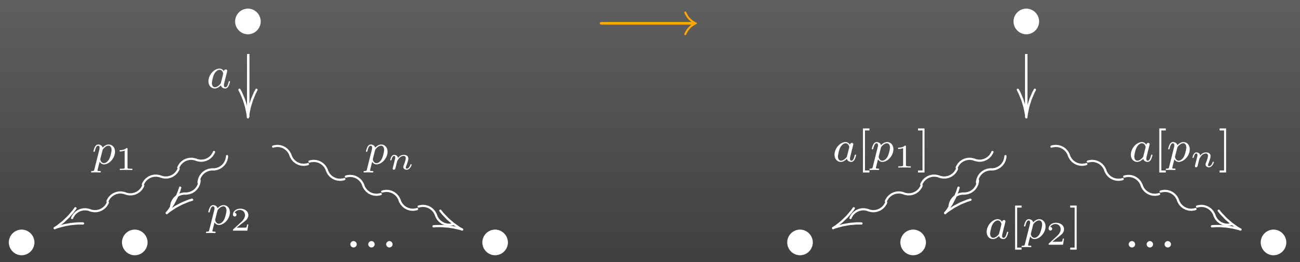
proof via cocongruences - behavioral equivalence

Example translation

Indeed $\mathbf{SSeg} \rightarrow \mathbf{Seg}$ since

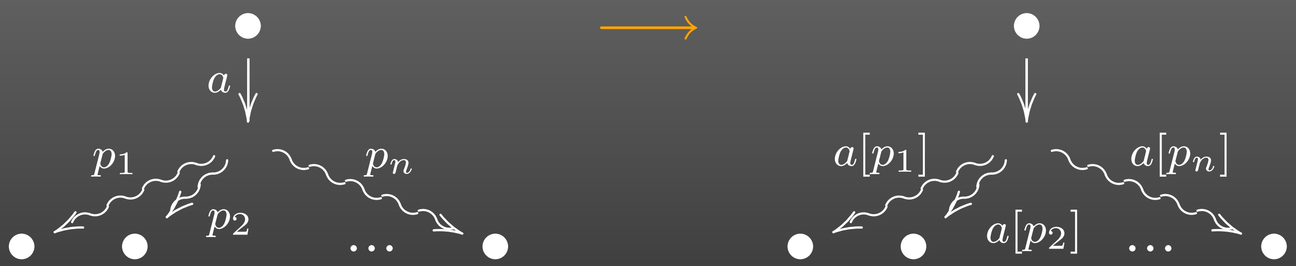
Example translation

Indeed $\mathbf{SSeg} \rightarrow \mathbf{Seg}$ since



Example translation

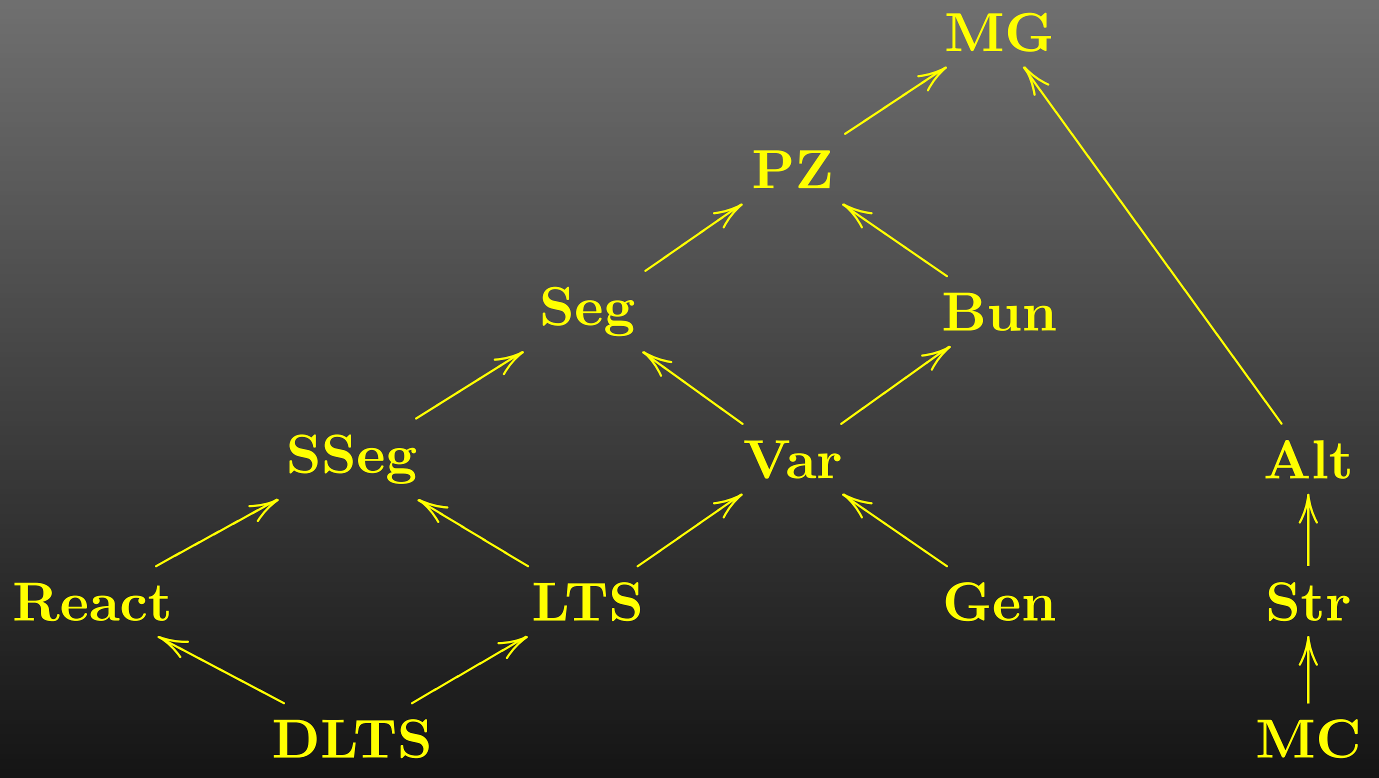
Indeed $\mathbf{SSeg} \to \mathbf{Seg}$ since



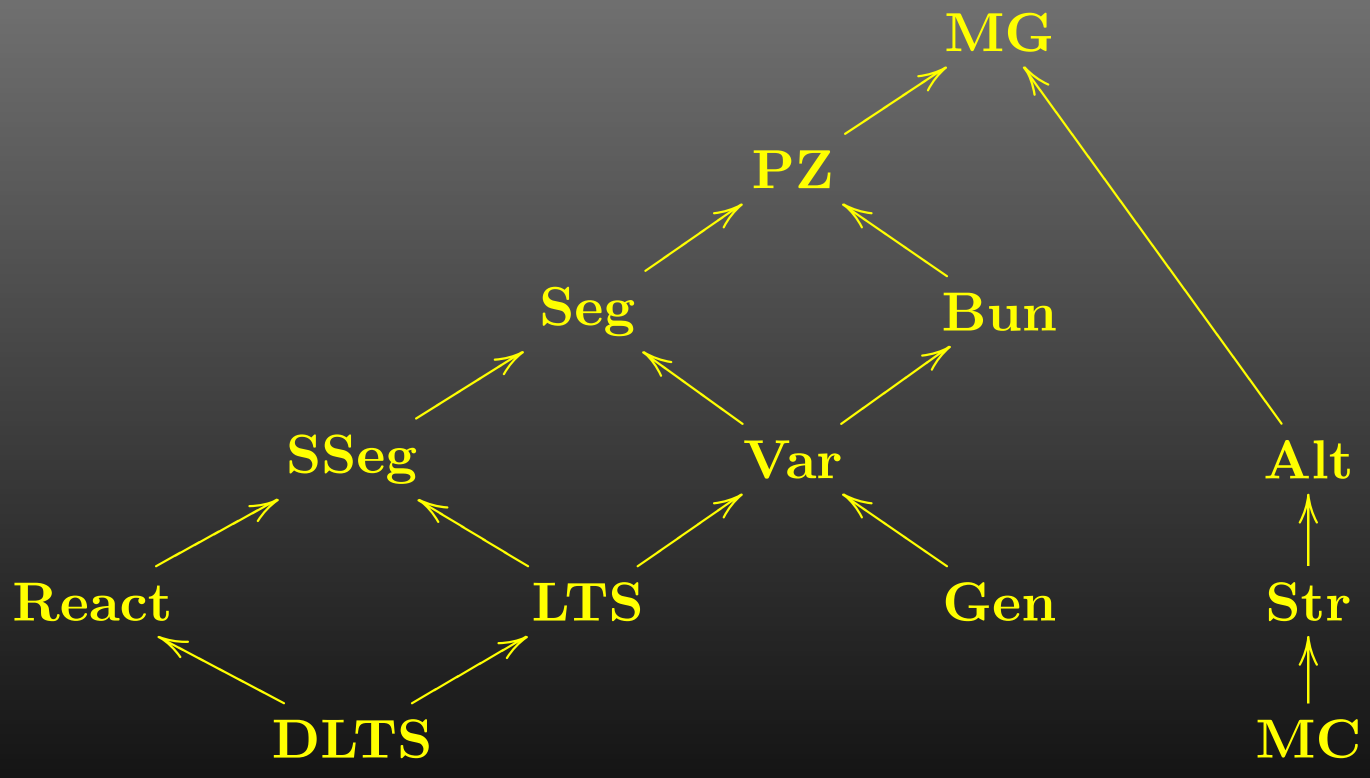
gives us an injective natural transformation

$$\mathcal{P}(A \times \mathcal{D}) \Longrightarrow \mathcal{P}\mathcal{D}(A \times \_\,)$$

The hierarchy...



The hierarchy...



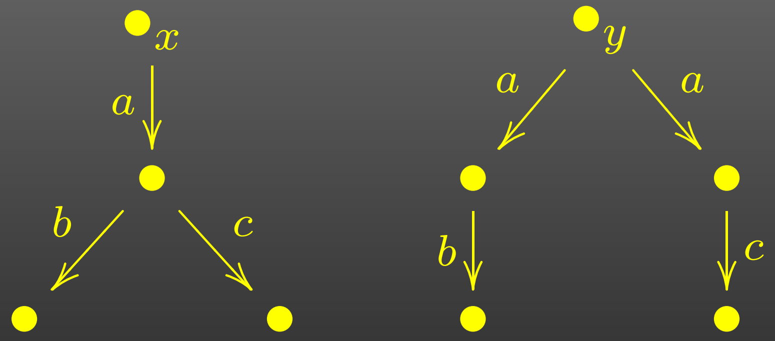
* Falk Bartels, AS, Erik de Vink

LT/BT spectrum

Bisimilarity is not the only semantics...

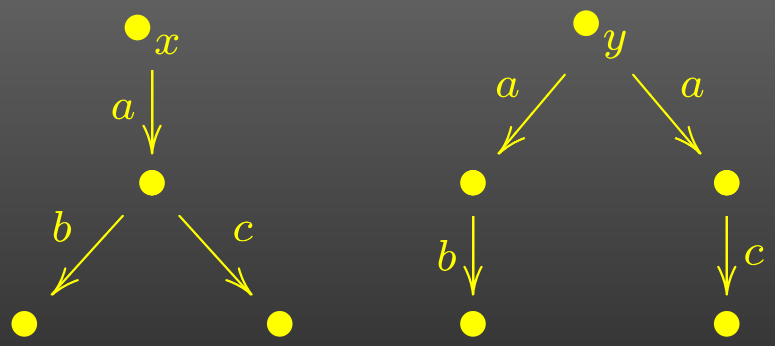
LT/BT spectrum

Are these non-deterministic systems equal ?



LT/BT spectrum

Are these non-deterministic systems equal ?

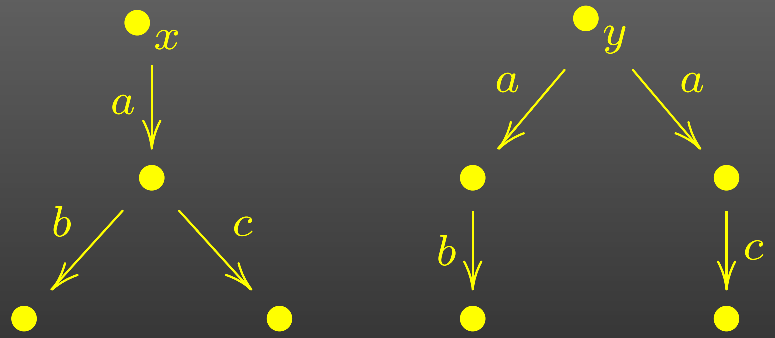


x and y are:

- 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)

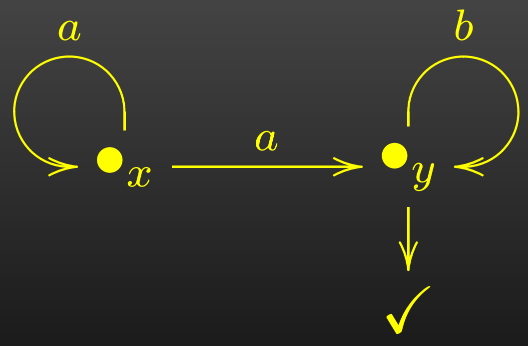
trace = the set of all possible linear behaviors

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

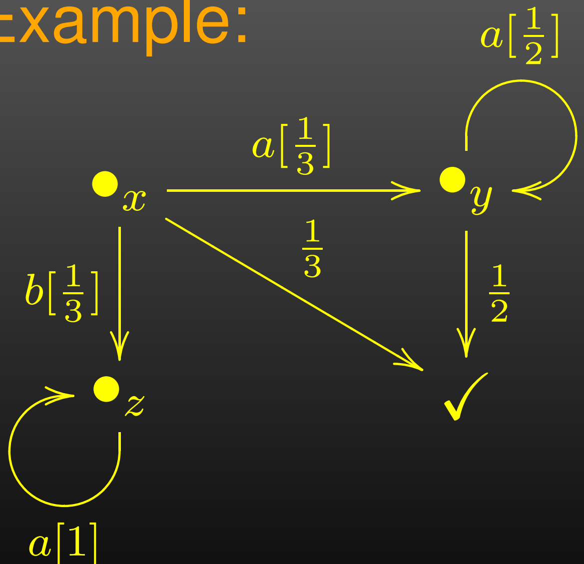
trace = sub-probability distribution over possible linear behaviors

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 ?

- ... this problem has a longer history (of partial solutions)
- Ichiro Hasuo, Bart Jacobs, AS: Generic Trace Theory

Trace of a coalgebra ?

- ... this problem has a longer history (of partial solutions)
- Ichiro Hasuo, Bart Jacobs, AS: Generic Trace Theory

main idea: coinduction in a Kleisli category

Conclusions

- probabilistic models are enriched LTS with quantitative information

Conclusions

- probabilistic models are enriched LTS with quantitative information
- coalgebras allow for a unified treatment of transition systems and bisimulation

Conclusions

- probabilistic models are enriched LTS with quantitative information
- coalgebras allow for a unified treatment of transition systems and bisimulation
- comparison of systems is then easy

Conclusions

- probabilistic models are enriched LTS with quantitative information
- coalgebras allow for a unified treatment of transition systems and bisimulation
- comparison of systems is then easy
- we have built an expressiveness hierarchy w.r.t bisimulation semantics

Conclusions

- probabilistic models are enriched LTS with quantitative information
- coalgebras allow for a unified treatment of transition systems and bisimulation
- comparison of systems is then easy
- we have built an expressiveness hierarchy w.r.t bisimulation semantics
- trace semantics can also be captured coalgebraically