Coalgebras in Kleisli Categories

Generic Theory for Traces and Simulations

Ichiro Hasuo
Radboud University Nijmegen
The Netherlands

June 13, 2006

Conventional view

Theory of coalgebras

||

Mathematics for systems' behavior

coalgebraic notion	interpretation
coalgebra $\begin{array}{c} FX \\ \uparrow \\ X \end{array}$	system
morphism of coalgebras $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & & \uparrow \\ X & \longrightarrow & Y \end{array}$	behavior preserving map
coinduction $\begin{array}{ccc} FX & \dashrightarrow & FZ \\ \uparrow & & \cong \uparrow \text{final} \\ X & \underset{\mathbf{beh}}{\dashrightarrow} & Z \end{array}$	**beh** gives behavior

Theory of coalgebras in Sets

$\|$

Mathematics for systems' behavior modulo bisimulation

coalgebraic notion	interpretation in Sets
coalgebra $\begin{array}{c} FX \\ \uparrow \\ X \end{array}$	system
morphism of coalgebras $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & & \uparrow \\ X & \longrightarrow & Y \end{array}$	behavior preserving map i.e. functional bisimulation
coinduction $\begin{array}{ccc} FX & \dashrightarrow & FZ \\ \uparrow & & \cong\uparrow \text{final} \\ X & \underset{\mathbf{beh}}{\dashrightarrow} & Z \end{array}$	**beh** gives behavior modulo bisimilarity

Can the "theory of coalgebra" be more generic?

- Interpretation of coalgebraic notions, in other categories?
- Can we handle semantics other than bisimilarity?

Coalgebras in Kleisli categories

Can the "theory of coalgebra" be more generic?

- Interpretation of coalgebraic notions, in other categories?
- Can we handle semantics other than bisimilarity?

This talk : the theory of coalgebras

- In **Kleisli categories**
- For **trace semantics** and **simulations**

Coalgebras in Kleisli categories

	in $\mathbf{Sets}$	in Kleisli category
semantics to be captured	bisimilarity	
coalgebra $\begin{array}{c} FX \\ \uparrow \\ X \end{array}$	system	
morphism of coalgebras $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & & \uparrow \\ X & \longrightarrow & Y \end{array}$	functional bisimulation	
coinduction $\begin{array}{ccc} FX & \dashrightarrow & FZ \\ \uparrow & & \cong\uparrow \text{final} \\ X & \underset{\mathbf{beh}}{\dashrightarrow} & Z \end{array}$	**beh** gives bisimilarity	

Coalgebras in Kleisli categories

	in $\mathbf{Sets}$	in Kleisli category
semantics to be captured	bisimilarity	trace equivalence
coalgebra $\begin{array}{c} FX \\ \uparrow \\ X \end{array}$	system	system
morphism of coalgebras $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & & \uparrow \\ X & \longrightarrow & Y \end{array}$	functional bisimulation	lax $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & \sqsupseteq & \uparrow \\ X & \longrightarrow & Y \end{array}$: **forward simulation** oplax $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & \sqsubseteq & \uparrow \\ X & \longrightarrow & Y \end{array}$: **backward simulation**
coinduction $\begin{array}{ccc} FX & \dashrightarrow & FZ \\ \uparrow & & \cong\uparrow\text{final} \\ X & \underset{\mathbf{beh}}{\dashrightarrow} & Z \end{array}$	**beh** gives bisimilarity	**beh** gives **trace semantics**

In other words: we present

generic theory of traces and simulations

employing theory of coalgebras.

- Generalizes existing work such as [Lynch-Vaandrager'95] on simulations and traces
- Genericity: both **non-determinism** and **probabilism** are handled in a uniform manner
- Results such as: soundness of forward simulation

$$\begin{pmatrix} FX \\ c\uparrow \\ X \end{pmatrix} \sqsubseteq_{\mathrm{fwd}} \begin{pmatrix} FY \\ d\uparrow \\ Y \end{pmatrix} \quad\Longrightarrow\quad \mathbf{tr}_c \sqsubseteq \mathbf{tr}_d \text{ (trace inclusion)}$$

is via coinduction!

References

- Ichiro Hasuo, Bart Jacobs and Ana Sokolova.
  Generic Trace Theory.
  Proc. CMCS 2006.
- Ichiro Hasuo.
  Generic Forward and Backward Simulations.
  Proc. CONCUR 2006.

Coalgebras in Kleisli categories

Coalgebraic modelling of systems

For trace semantics and simulations,
we separate **branching type** and **transition type**.

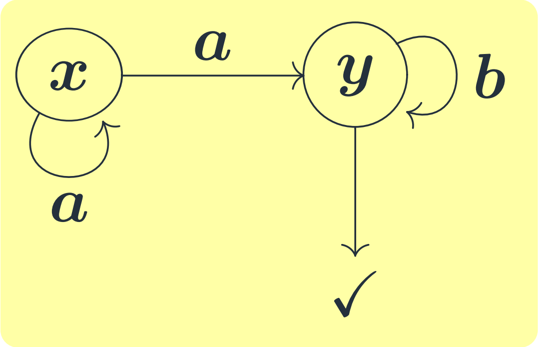
In Sets

$$\begin{array}{c} TFX \\ \uparrow c \\ X \end{array}$$

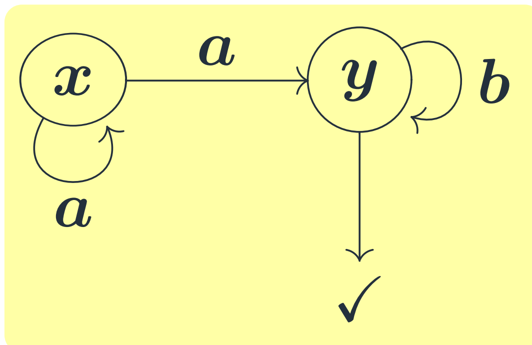
- T: a **monad** with suitable order, specifying branching type.
- F: a functor, specifying transition type.
- $FT \Rightarrow TF$, distributive law.

Let's look at examples...

(Non-deterministic) LTS with explicit termination



(Non-deterministic) LTS with explicit termination



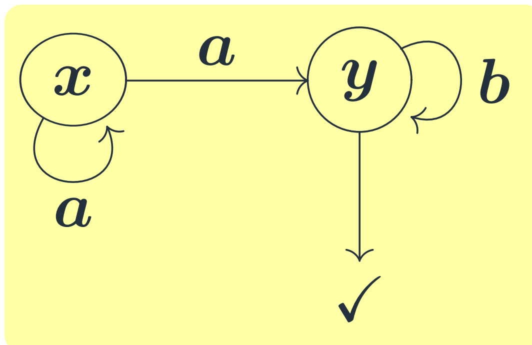
as a coalgebra

$$\mathcal{P}(1 + \Sigma \times X) \qquad \{(a,x),\ (a,y)\} \qquad \{(b,y),\ \checkmark\}$$

$$\uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow$$

$$X \qquad\qquad x \qquad\qquad y$$

(Non-deterministic) LTS with explicit termination



as a coalgebra

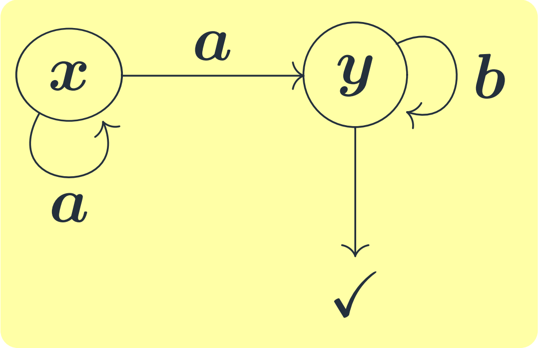
$$\mathcal{P}(1 + \Sigma \times X) \qquad \{\, (a, x),\ (a, y)\} \qquad \{\, (b, y),\ \checkmark\}$$

$$\uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow$$

$$X \qquad\qquad x \qquad\qquad y$$

- Branching: **non-deterministic**, modelled by the **powerset monad** $\mathcal{P}$.

(Non-deterministic) LTS with explicit termination



as a coalgebra

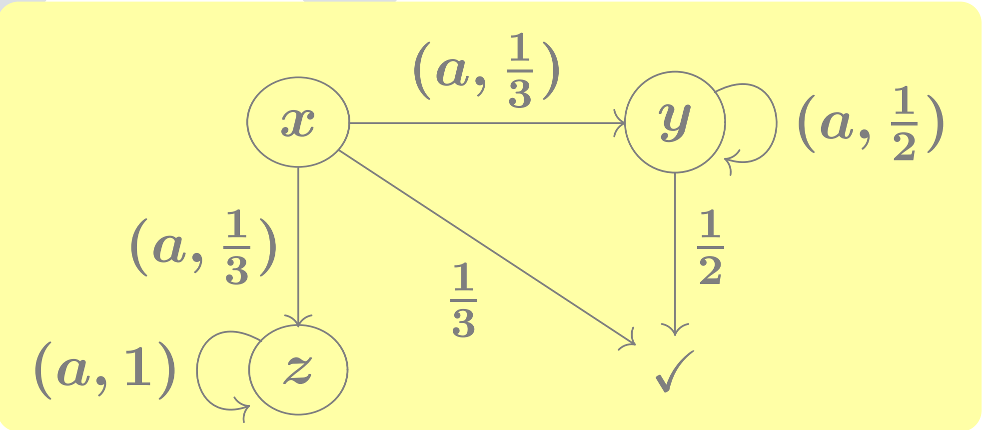
$$\mathcal{P}(1 + \Sigma \times X) \qquad \{(a, x), (a, y)\} \qquad \{(b, y), \checkmark\}$$
$$\uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow$$
$$X \qquad\qquad x \qquad\qquad y$$

- Branching: **non-deterministic**, modelled by the **powerset monad** $\mathcal{P}$.
- Transition: either
  - output symbol + next state, or
  - termination $\checkmark$

  modelled by the functor $1 + \Sigma \times \_$.

Example 2 (skip!)

Probabilistic LTS with explicit termination



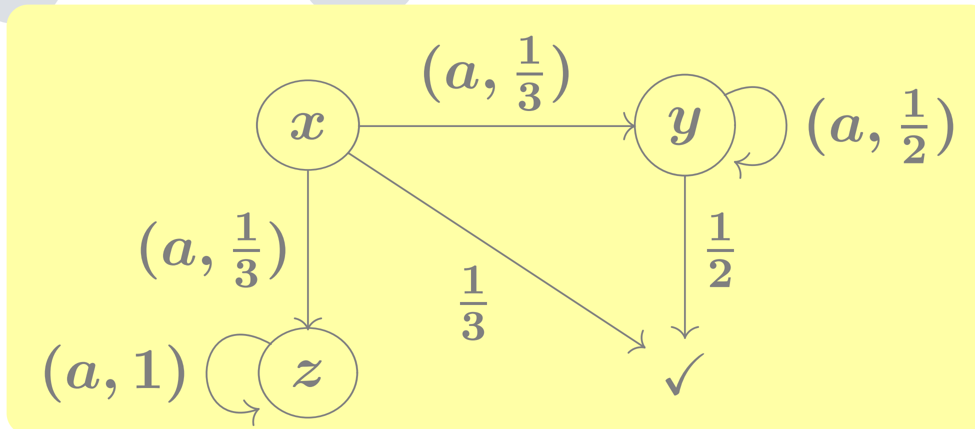
Compared to the previous one,

- The transition type is the same, $1 + \Sigma \times \_$.
- Branching: **probabilistic**,
  modelled by the **subdistribution monad** $\mathcal{D}$

$$\mathcal{D}X = \{\text{probability (sub)distributions over } X\}$$
$$= \{d : X \to [0,1] \mid \sum_{x \in X} d(x) \leq 1\}$$

Example 2 (skip!)

Probabilistic LTS with explicit termination



$$\mathcal{D}(1 + \Sigma \times X) \qquad \begin{bmatrix} (a,z) & \mapsto & 1/3 \\ (a,y) & \mapsto & 1/3 \\ \checkmark & \mapsto & 1/3 \end{bmatrix} \qquad \begin{bmatrix} (a,y) & \mapsto & 1/2 \\ \checkmark & \mapsto & 1/2 \end{bmatrix}$$

$$c \uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow$$

$$X \qquad\qquad x \qquad\qquad y$$

Kleisli category $\mathcal{K}\ell(T)$ for monad T

- Objects are sets (same as $\mathbf{Sets}$)
- Arrows: $$\frac{X \longrightarrow Y \text{ in } \mathcal{K}\ell(T)}{X \longrightarrow TY \text{ in } \mathbf{Sets}}$$
  - Arrows in $\mathcal{K}\ell(T)$ are functions with **structured output**. [Moggi]
  - T's effect is hidden in $\mathcal{K}\ell(T)$.

Kleisli category $\mathcal{K}\ell(T)$ for monad T

- Objects are sets (same as $\mathbf{Sets}$)
- Arrows: $$\frac{X \longrightarrow Y \text{ in } \mathcal{K}\ell(T)}{X \longrightarrow TY \text{ in } \mathbf{Sets}}$$
  - Arrows in $\mathcal{K}\ell(T)$ are functions with **structured output**. [Moggi]
  - T's effect is hidden in $\mathcal{K}\ell(T)$.

Examples

- $X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$

  $\iff$ function $X \to \mathcal{P}Y$ "non-deterministic function"

  $\iff$ relation between X and Y
- $X \to Y$ in $\mathcal{K}\ell(\mathcal{D})$

  $\iff$ function $X \to \mathcal{D}Y$ "probabilistic function"

  mapping $x \mapsto$ (prob. distribution over Y)

Coalgebras in Kleisli category

$$\frac{X \longrightarrow Y \text{ in } \mathcal{K}\ell(T)}{X \longrightarrow TY \text{ in Sets}} \quad \text{hence}$$

$$\left(\begin{array}{c} TFX \\ c \uparrow \\ X \end{array} \text{ in Sets} \right) = \left(\begin{array}{c} FX \\ c \uparrow \\ X \end{array} \text{ in } \mathcal{K}\ell(T) \right)$$

- Branching is “absorbed in the base category $\mathcal{K}\ell(T)$”
- [Power-Turi, CTCS’99]
- Now a system is an F-coalgebra!
- We can apply theory of coalgebras in $\mathcal{K}\ell(T)$: what is
  - coinduction?
  - morphism of coalgebras?

Trace semantics via coinduction [CMCS'06]

Coinduction in Kleisli category

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

Questions

- What is the final coalgebra in $\mathcal{K}\ell(T)$?
- What is the unique map via coinduction?

Coinduction in Kleisli category

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

$$\begin{array}{ccc} FX & \dashrightarrow & FZ \\ c \uparrow & & \uparrow \cong \\ X & \dashrightarrow & Z \end{array}$$

system — final coalg.

Questions

- What is the final coalgebra in $\mathcal{K}\ell(T)$?
  Answer [CMCS'06] Initial F-algebra in $\mathbf{Sets}$!
- What is the unique map via coinduction?
  Answer [CMCS'06] Trace semantics!

This holds for

- a wide variety of functors F and
- monads T with a suitable order structure, such as $\mathcal{P}$ and $\mathcal{D}$.

Example

Coinduction in $\mathcal{K}\ell(\mathcal{P})$:

The commutation amounts to
the standard (co)inductive definition of traces:

- $\langle\rangle \in \mathsf{tr}_c(x)$ iff $\checkmark \in c(x)$
- $a \cdot s \in \mathsf{tr}_c(x)$ iff
  $\exists x' \in X.\ \ (a, x') \in c(x) \ \wedge\ s \in \mathsf{tr}_c(x')$

Simulations as lax/oplax coalgebra morphisms [CONCUR'06]

Explicit start states

It is convenient to have start states explicit.

Now a **system** is:

$$\text{in } \mathcal{K}\ell(T) \quad \begin{array}{c} FX \\ \uparrow c \\ X \\ \uparrow s \\ 1 \end{array} \quad , \quad \text{that is,} \quad \text{in Sets} \left(\begin{array}{c} TX \\ \uparrow s \\ 1 \end{array} , \begin{array}{c} TFX \\ \uparrow c \\ X \end{array} \right)$$

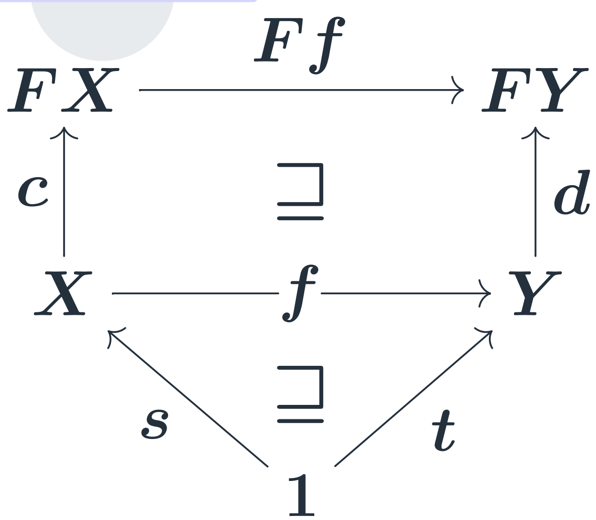
Examples

- $\begin{array}{c} \mathcal{P}X \\ \uparrow s \\ 1 \end{array}$: set of possible start states
- $\begin{array}{c} \mathcal{D}X \\ \uparrow s \\ 1 \end{array}$: probability distribution over possible start states

Forward simulations

are identified as **lax morphisms of coalgebras**!

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



- $d \circ f \sqsubseteq Ff \circ c$, as functions $X \rightrightarrows TFY$
- (s, c) simulates (t, d)
- Yields trace inclusion: $\mathbf{tr}_{(t,d)} \sqsubseteq \mathbf{tr}_{(s,c)}$

Let's take a closer look...

Forward simulations

In $\mathcal{K}\ell(\mathcal{P})$: Note an arrow is a relation

$$\begin{array}{ccc} 1 + \Sigma \times X & \longrightarrow & 1 + \Sigma \times Y \\ {\scriptstyle c}\uparrow & \sqsupseteq & \uparrow{\scriptstyle d} \\ X & \longrightarrow & Y \end{array}$$

x

Forward simulations

In $\mathcal{K}\ell(\mathcal{P})$:

Note an arrow is a relation

$$x \xrightarrow{a} x'$$

$$\begin{array}{ccc} x & \xrightarrow{a} & x' \\ & & \vdots \\ & & y' \end{array}$$

$$\begin{array}{ccc} 1 + \Sigma \times X & \longrightarrow & 1 + \Sigma \times Y \\ c \uparrow & \sqsupseteq & \uparrow d \\ X & \longrightarrow & Y \end{array}$$

$$x$$

In $\mathcal{K}\ell(\mathcal{P})$:

Note an arrow is a relation

$$\begin{array}{ccc} 1 + \Sigma \times X & \longrightarrow & 1 + \Sigma \times Y \\ c \uparrow & \sqsupseteq & \uparrow d \\ X & \longrightarrow & Y \end{array}$$

x
$\vdots$
$y \xrightarrow{a} y'$

x

x
$\vdots$
y

In $\mathcal{K}\ell(\mathcal{P})$:

Note: an arrow is a relation

$x \xrightarrow{a} x'$

$x \xrightarrow{a} x'$, $\vdots$, y' $\Longleftarrow$ x, $\vdots$, $y \xrightarrow{a} y'$

$$\begin{array}{ccc} 1+\Sigma\times X & \longrightarrow & 1+\Sigma\times Y \\ c\uparrow & \sqsupseteq & \uparrow d \\ X & \longrightarrow & Y \end{array}$$

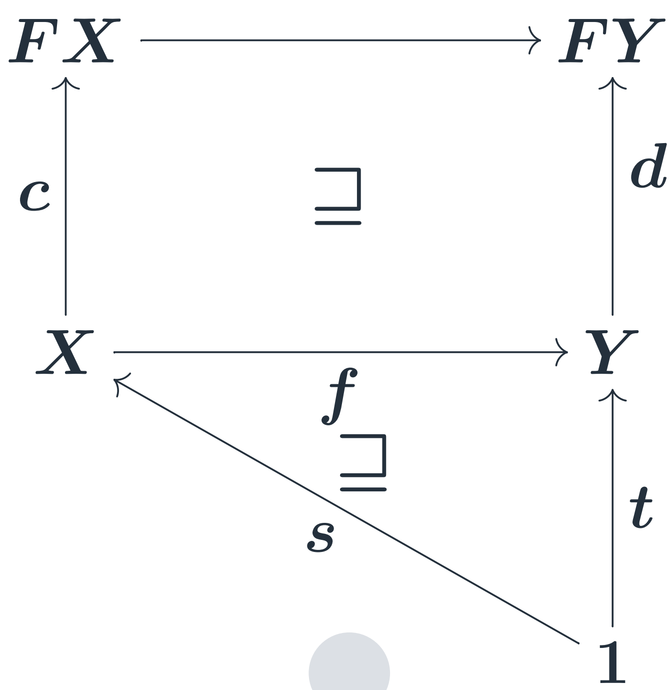
x

x, $\vdots$, y

Hence $\begin{array}{ccc} x & & \\ \vdots & & \\ y & \xrightarrow{a} & y' \end{array}$ implies $\begin{array}{ccc} x & \xrightarrow{a} & \exists x' \\ \vdots & & \vdots \\ y & \xrightarrow{a} & y' \end{array}$.

Simulations yield trace inclusion (soundness)

Theorem $(t, d) \sqsubseteq_{\mathrm{fwd}} (s, c) \implies \mathbf{tr}_{(t,d)} \sqsubseteq \mathbf{tr}_{(s,c)}$.



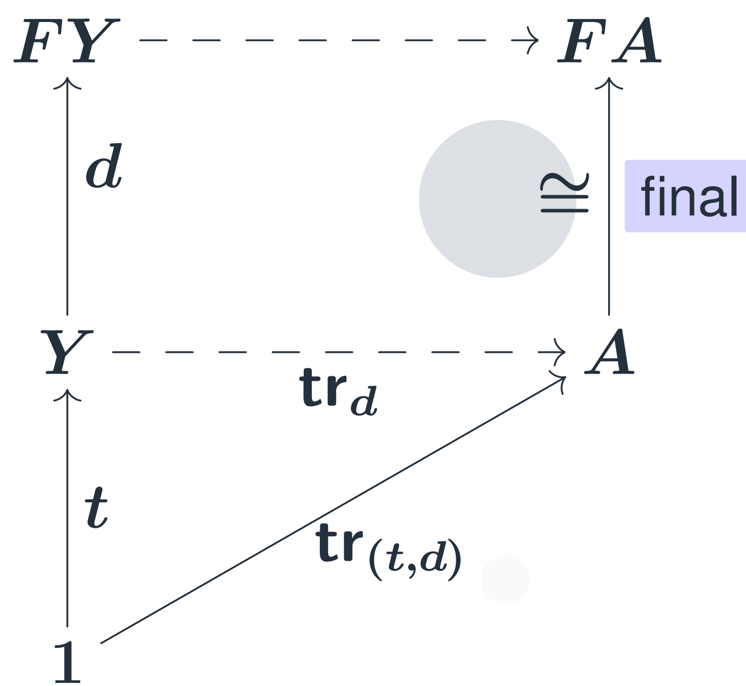
Simulations yield trace inclusion (soundness)

Theorem $$(t, d) \sqsubseteq_{\mathrm{fwd}} (s, c) \implies \mathsf{tr}_{(t,d)} \sqsubseteq \mathsf{tr}_{(s,c)} \ .$$

$$1 \xrightarrow{t} Y \xrightarrow{d} FY$$

Simulations yield trace inclusion (soundness)

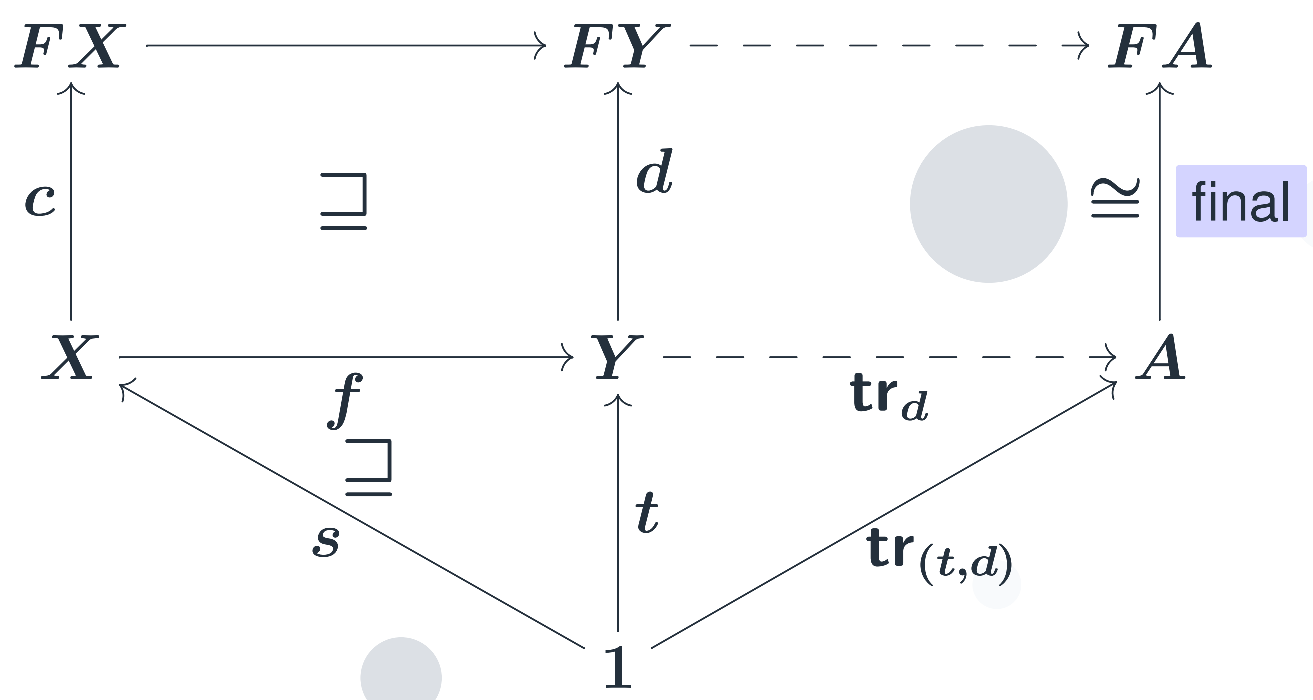
Theorem $\quad (t, d) \sqsubseteq_{\mathrm{fwd}} (s, c) \implies \mathsf{tr}_{(t,d)} \sqsubseteq \mathsf{tr}_{(s,c)}$.



$$\mathsf{tr}_{(t,d)} = \mathsf{tr}_d \circ t$$

Simulations yield trace inclusion (soundness)

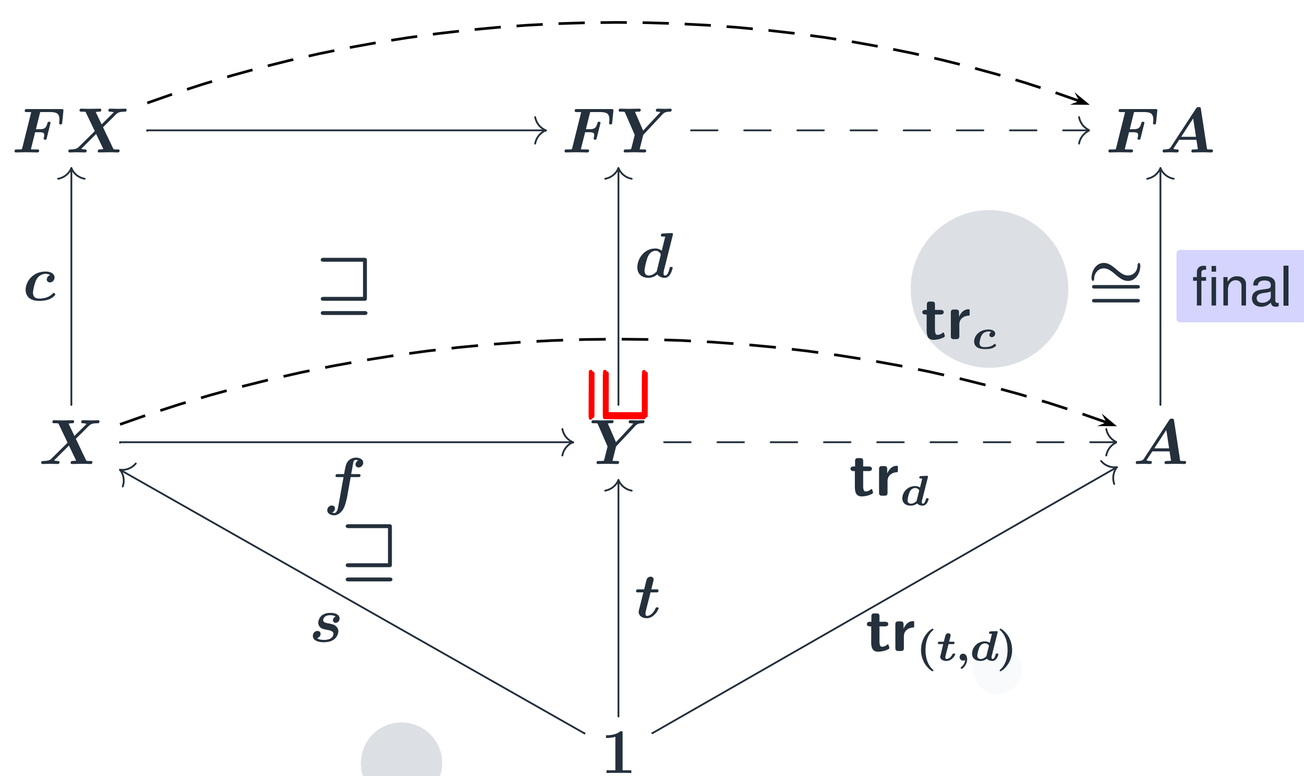
Theorem $(t, d) \sqsubseteq_{\mathrm{fwd}} (s, c) \implies \mathbf{tr}_{(t,d)} \sqsubseteq \mathbf{tr}_{(s,c)}$.



$$\mathbf{tr}_{(t,d)} = \mathbf{tr}_d \circ t \sqsubseteq \mathbf{tr}_d \circ f \circ s$$

Simulations yield trace inclusion (soundness)

Theorem $(t, d) \sqsubseteq_{\mathrm{fwd}} (s, c) \implies \mathbf{tr}_{(t,d)} \sqsubseteq \mathbf{tr}_{(s,c)}$.



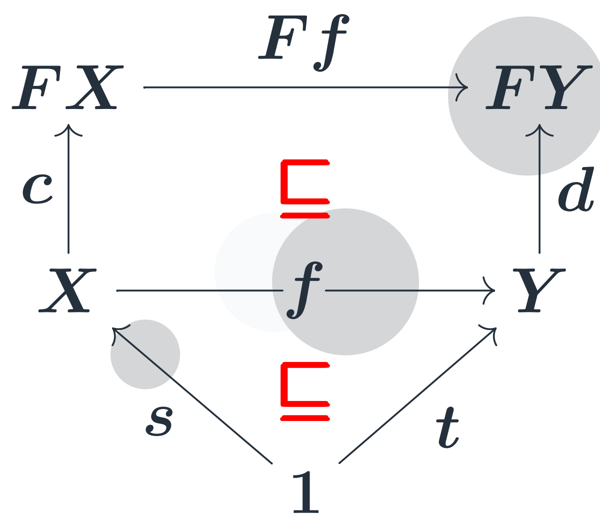
$$\mathbf{tr}_{(t,d)} = \mathbf{tr}_d \circ t \sqsubseteq \mathbf{tr}_d \circ f \circ s \sqsubseteq \mathbf{tr}_c \circ s = \mathbf{tr}_{(s,c)} .$$

- **Proposition** [Fiore'96/Plotkin]
  Trace map is the biggest lax coalgebra morphism.

Backward simulations

Similarly, backward simulations
as **oplax morphisms of coalgebras**:

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



- $\sqsubseteq$ instead of $\sqsupseteq$
- (t, d) simulates (s, c)
- For example:

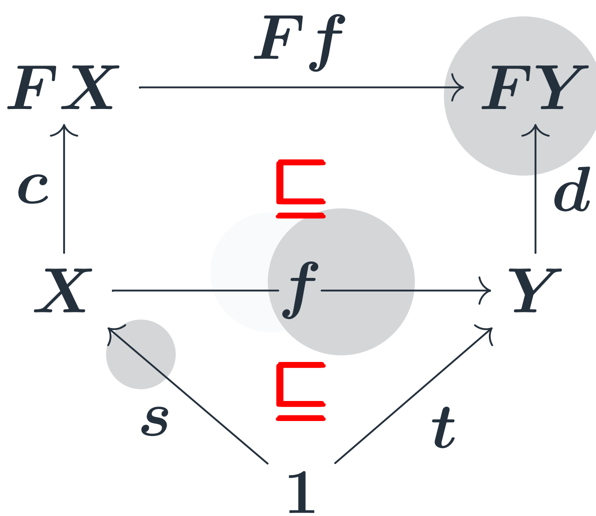
  $x \xrightarrow{a} x'$, $x' \,\vdots\, y'$ implies $\exists y$ with $x \,\vdots\, y$ and $y \xrightarrow{a} y'$

- Again yields trace inclusion:
  $\mathbf{tr}_{(s,c)} \sqsubseteq \mathbf{tr}_{(t,d)}$

Backward simulations

Similarly, backward simulations
as **oplax morphisms of coalgebras**:

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



- $\sqsubseteq$ instead of $\sqsupseteq$
- (t, d) simulates (s, c)
- For example:

$$\begin{array}{ccc} x & \xrightarrow{a} & x' \\ & & \vdots \\ & & y' \end{array} \quad \text{implies} \quad \begin{array}{ccc} x & \xrightarrow{a} & x' \\ \vdots & & \vdots \\ \exists y & \xrightarrow{a} & y' \end{array}$$

- Again yields trace inclusion:
$\mathbf{tr}_{(s,c)} \sqsubseteq \mathbf{tr}_{(t,d)}$

Also: **completeness result** for hybrid "backward-forward" simulations.

Toward a generic theory

- **In summary**: in Kleisli categories,
  - trace semantics via coinduction
  - forward/backward simulations as lax/oplax coalgebra morphisms
  - soundness/completeness of simulations
- **Genericity**: valid for
  - monad T, type of branching:
    $\mathcal{P}$ (non-determinism) or $\mathcal{D}$ (probabilism)
  - functor F, type of transition:
    shapely (i.e. almost polynomial)

- **Practical implication** : for a given type of systems,
  - definition of forward/backward simulations by instantiating coalgebraic definition,

- **Practical implication** : for a given type of systems,
  - definition of forward/backward simulations by instantiating coalgebraic definition,
  - for which soundness/completeness comes **for free**.
  - Cf. In formal verification, finding a simulation is a common technique to establish trace inclusion.

Toward a generic theory

- **Practical implication** : for a given type of systems,
  - definition of forward/backward simulations by instantiating coalgebraic definition,
  - for which soundness/completeness comes **for free**.
  - Cf. In formal verification, finding a simulation is a common technique to establish trace inclusion.
  - Especially, a gap from non-deterministic systems to its probabilistic version is trivial.
  - E.g. Probabilistic version of "anonymity simulation" [Kawabe et al '06] (ongoing work)

Conclusions and future work

Summary

	in $\mathbf{Sets}$	in Kleisli category
semantics to be captured	bisimilarity	trace equivalence
coalgebra $\begin{array}{c} FX \\ \uparrow \\ X \end{array}$	system	system
morphism of coalgebras $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & & \uparrow \\ X & \longrightarrow & Y \end{array}$	functional bisimulation	lax $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & \sqsupseteq & \uparrow \\ X & \longrightarrow & Y \end{array}$: **forward simulation** oplax $\begin{array}{ccc} FX & \longrightarrow & FY \\ \uparrow & \sqsubseteq & \uparrow \\ X & \longrightarrow & Y \end{array}$: **backward simulation**
coinduction $\begin{array}{ccc} FX & \dashrightarrow & FZ \\ \uparrow & & \cong\uparrow \text{final} \\ X & \underset{\mathbf{beh}}{\dashrightarrow} & Z \end{array}$	**beh** gives bisimilarity	**beh** gives **trace semantics**

Conclusions and future work

- Theory of coalgebras, employed in Kleisli categories
- Generic theory of traces and simulations
  - Soundness/completeness for free

Conclusions and future work

- Theory of coalgebras, employed in Kleisli categories
- Generic theory of traces and simulations
  - Soundness/completeness for free

Future work

- Infinite traces [Jacobs, CMCS'04]
- Internal actions
- Linear-time logic using $\circlearrowright \mathcal{X} \rightleftarrows \mathcal{A}^{\mathrm{op}} \circlearrowleft$
  (Ongoing work with A. Kurz)
- Process calculi and compositionality (bialgebraic view?)
- Semantics between trace sem. and bisimilarity, in the van Glabbeek spectrum (Cf. B. Klin)
- As an instance of the bigger "systems and tests" view, via $\circlearrowright \mathcal{X} \rightleftarrows \mathcal{A}^{\mathrm{op}} \circlearrowleft$