Context-Free Languages via Coalgebraic Trace Semantics

Ichiro Hasuo Bart Jacobs

Radboud University Nijmegen, the Netherlands

Historical remarks

Coalgebraic treatment of **trace semantics** for non-deterministic systems...

- Power & Turi, CTCS'99
  - For $\mathcal{P}(1 + \Sigma \times -)$-coalgebras.
  - Working in Kleisli category $\mathbf{Sets}_{\mathcal{P}}$.
- BJ, CMCS'04
  - Final coalgebra in $\mathbf{Sets}$ yields weakly final coalgebra in $\mathbf{Rel} = \mathbf{Sets}_{\mathcal{P}}$.
  - First result for general functors.
- IH & BJ, CALCO'05 IH & BJ, CALCO-jnr
  - Current work...

Overview

- **Initial algebra in $\mathbf{Sets}$ yields final coalgebra in $\mathbf{Rel}$**
  - Finite trace semantics for **non-deterministic** systems
  - Application: context-free grammars/languages
  - $\mathbf{Rel} \cong \mathbf{Sets}_{\mathcal{P}}$, Kleisli category

- **Same for subdistribution monad $\mathcal{D}$, instead of $\mathcal{P}$**
  - Finite trace semantics for **probabilistic** systems
  - In Proc. CALCO-jnr.

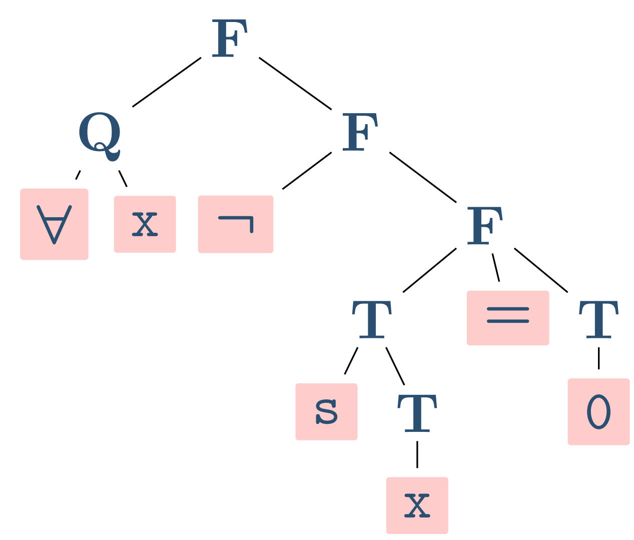
- **(Co)algebraic formulation of CFG/CFL**
  - Monad structure, "fundamental span" of languages

Context-free grammars: example

- Terminal symbols: $0, \mathrm{x}, \mathrm{s}, =, \wedge, \neg, \forall$
- Non-terminal symbols: $\mathbf{T}, \mathbf{Q}, \mathbf{F}$
- Generation rules:

$$\mathbf{T} \blacktriangleright 0 \qquad \mathbf{Q} \blacktriangleright \forall \mathrm{x} \qquad \mathbf{F} \blacktriangleright \mathbf{T}{=}\mathbf{T} \qquad \mathbf{F} \blacktriangleright \mathbf{F}\wedge\mathbf{F}$$
$$\mathbf{T} \blacktriangleright \mathrm{x} \qquad \mathbf{F} \blacktriangleright \mathbf{Q}\mathbf{F} \qquad \mathbf{F} \blacktriangleright \neg\mathbf{F}$$
$$\mathbf{T} \blacktriangleright \mathrm{s}\mathbf{T}$$

$\forall \mathrm{x} \neg(\mathrm{s}(\mathrm{x}) = 0)$ is generated as:



CFG as coalgebra

- $$\begin{array}{c} \mathcal{P}((\Sigma + X)^*) \\ \uparrow c \\ X \end{array}$$ in $\mathbf{Sets}$ $\left[\begin{array}{rl} X: & \text{non-terminals} \\ \Sigma: & \text{terminals} \end{array} \right]$

- E.g. $$\begin{array}{c} \{\mathrm{T}{=}\mathrm{T},\ \mathrm{QF},\ \mathrm{F}{\wedge}\mathrm{F},\ \neg\mathrm{F}\} \\ \uparrow \\ \mathrm{F} \end{array}$$

- **Question**

  How can we characterize a context-free language generated by a CFG?

CFG/CFL, coalgebraically

Introduction

Motivating example: CFG
- Context-free grammars: example
- CFG as coalgebra
- CFG/CFL, coalgebraically

Main technical result

Monad structures in languages

Application of main result

Generalization: probabilistic systems

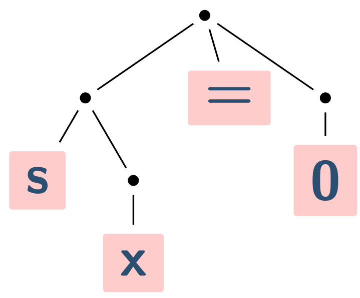
Future work and conclusions

$$\begin{array}{c} \mathcal{P}FX \\ c\uparrow \\ X \end{array}$$

where $F = (\Sigma + -)^*$

Observations

- Initial F-algebra
  = set of finite parse trees like



- Final F-coalgebra =
  initial F-algebra + (**infinite** parse trees, undesired)
- c is a coalgebra in **Kleisli category** $\mathbf{Sets}_{\mathcal{P}}$.
- Equivalently, c is a coalgebra in $\mathbf{Rel}$, because

$$\frac{c : X \longrightarrow \mathcal{P}FX}{R \rightarrowtail X \times FX}$$

Main technical result

Let F be a shapely functor in $\mathbf{Sets}$ (details later).

Let $\begin{array}{c} FA \\ \cong \big\downarrow \alpha \\ A \end{array}$ be the initial F-algebra in $\mathbf{Sets}$.

Theorem

Coalgebra $\begin{array}{c} \mathcal{P}FA \\ \big\uparrow \{-\} \\ FA \\ \big\uparrow \alpha^{-1} \\ A \end{array}$ in $\mathbf{Sets}$ i.e. $\begin{array}{c} FA \\ \big\uparrow \\ A \end{array}$ in $\mathbf{Sets}_{\mathcal{P}}$ is final in $\mathbf{Sets}_{\mathcal{P}}$.

Finite trace of non-deterministic systems

Hence, a non-deterministic coalgebra in $\mathbf{Sets}$

$$\begin{array}{c} \mathcal{P}FX \\ c \uparrow \\ X \end{array}$$

has a unique **finite trace** $X \xrightarrow{\mathsf{ft}(c)} \mathcal{P}A$

such that:

In $\mathbf{Sets}_{\mathcal{P}}$

$$\begin{tikzcd} FX \arrow[r, dashed, "F(\mathsf{ft}(c))"] & FA \\ X \arrow[u, "c"] \arrow[r, dashed, "\mathsf{ft}(c)"'] & A \arrow[u, "\cong", "\{-\} \circ \alpha^{-1}"'] \end{tikzcd}$$

History of proofs

Three versions of proofs for the main result...

- Nov. 2004, by IH

  In submitted paper/technical report.
  3 pages, lengthy, hard to get intuition.

History of proofs

Three versions of proofs for the main result...

- Nov. 2004, by IH

  In submitted paper/technical report.
  3 pages, lengthy, hard to get intuition.

- March 2005, by IH/BJ

  In the proceedings, what you have at hand.
  2 pages, concrete, intuitive.
  Easy to generalize.

History of proofs

Three versions of proofs for the main result...

- Nov. 2004, by IH

  In submitted paper/technical report.
  3 pages, lengthy, hard to get intuition.

- March 2005, by IH/BJ

  In the proceedings, what you have at hand.
  2 pages, concrete, intuitive.
  Easy to generalize.

- June 2005, by BJ

  Latest version.
  2 lines. Made IH depressed.

  Presented now.

Proof of main result, shortest version

$$
\begin{tikzcd}
\mathbf{Sets} \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathbf{Sets}_{\mathcal{P}} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\cong"] & (\mathbf{Sets}_{\mathcal{P}})^{\mathrm{op}} \arrow[l, bend left]
\end{tikzcd}
$$

Kleisli construction and self-duality.

Note that: $\mathbf{Sets}_{\mathcal{P}} \cong \mathbf{Rel}$.

Proof of main result, shortest version

$$\begin{tikzcd}[column sep=huge]
\mathbf{Sets} \arrow[loop left, "F"] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathbf{Sets}_{\mathcal{P}} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\cong"] & (\mathbf{Sets}_{\mathcal{P}})^{\mathrm{op}} \arrow[l, bend left] \arrow[loop right, "(F_{\mathcal{P}})^{\mathrm{op}}"]
\end{tikzcd}$$

[Power & Turi, CTCS'99]
Due to the distributive law $F\mathcal{P} \Rightarrow \mathcal{P}F$ (**power law**),
F lifts to $F_{\mathcal{P}}$ on $\mathbf{Sets}_{\mathcal{P}}$.

Proof of main result, shortest version

$$
\begin{tikzcd}
\mathbf{Alg}(F) \arrow[d] & & \mathbf{Alg}\big((F_{\mathcal{P}})^{\mathrm{op}}\big) \arrow[d] \\
\mathbf{Sets} \arrow[loop left, "F"] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathbf{Sets}_{\mathcal{P}} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\cong"] & (\mathbf{Sets}_{\mathcal{P}})^{\mathrm{op}} \arrow[l, bend left] \arrow[loop right, "(F_{\mathcal{P}})^{\mathrm{op}}"]
\end{tikzcd}
$$

Proof of main result, shortest version

$$
\begin{tikzcd}
\mathbf{Alg}(F) \arrow[rr, bend left] \arrow[rr, phantom, "\perp"] \arrow[d] & & \mathbf{Alg}\big((F_{\mathcal{P}})^{\mathbf{op}}\big) \arrow[ll, bend left] \arrow[d] \\
\mathbf{Sets} \arrow[r, bend left] \arrow[r, phantom, "\perp"] \arrow[loop below, "F"] & \mathbf{Sets}_{\mathcal{P}} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\cong"] & (\mathbf{Sets}_{\mathcal{P}})^{\mathbf{op}} \arrow[l, bend left] \arrow[loop below, "(F_{\mathcal{P}})^{\mathbf{op}}"]
\end{tikzcd}
$$

By a standard result the adjunction lifts.
See e.g. [Hermida & BJ, Inf. & Comp., 1998].

Proof of main result, shortest version

$$
\begin{tikzcd}
 & & (\mathbf{CoAlg}(F_{\mathcal{P}}))^{\mathrm{op}} \\
\mathbf{Alg}(F) \arrow[d] \arrow[rr, bend left] \arrow[rr, phantom, "\perp"] & & \mathbf{Alg}((F_{\mathcal{P}})^{\mathrm{op}}) \arrow[u, "\cong"'] \arrow[d] \arrow[ll, bend left] \\
\mathbf{Sets} \arrow[loop left, "F"] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathbf{Sets}_{\mathcal{P}} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\cong"] & (\mathbf{Sets}_{\mathcal{P}})^{\mathrm{op}} \arrow[l, bend left] \arrow[loop right, "(F_{\mathcal{P}})^{\mathrm{op}}"]
\end{tikzcd}
$$

Initial object is preserved by left adjoints.
Q.E.D.

Possibly infinite traces

- To each $\begin{array}{c} \mathcal{P}FX \\ c \uparrow \\ X \end{array}$, our finality result assigns its

  finite trace $X \xrightarrow{\mathsf{ft}(c)} \mathcal{P}A$,

  where A is the initial F-algebra.

- **Possibly infinite trace** $X \xrightarrow{\mathsf{trace}(c)} \mathcal{P}Z$,

  where Z is final F-coalgebra, is given by [BJ, CMCS'04].
  Not via finality, but via (weak finality + "biggest" choice).

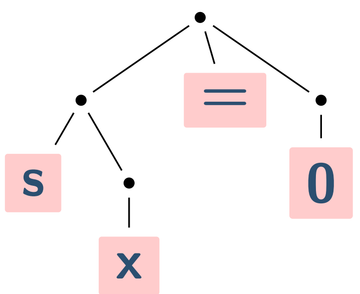
- Their relationship is investigated in our paper.

Intermezzo:

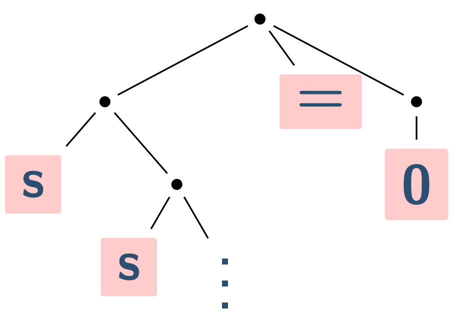
Monad structures in languages

Fundamental span of languages

- Σ^* set of **flat strings** over Σ
- $\Sigma^\triangle$ Initial $(\Sigma + -)^*$-algebra,
  set of **finite** parse trees over Σ e.g.



- $\Sigma^\wedge$ Final $(\Sigma + -)^*$-coalgebra,
  $\Sigma^\triangle$ plus **infinite** parse trees e.g.

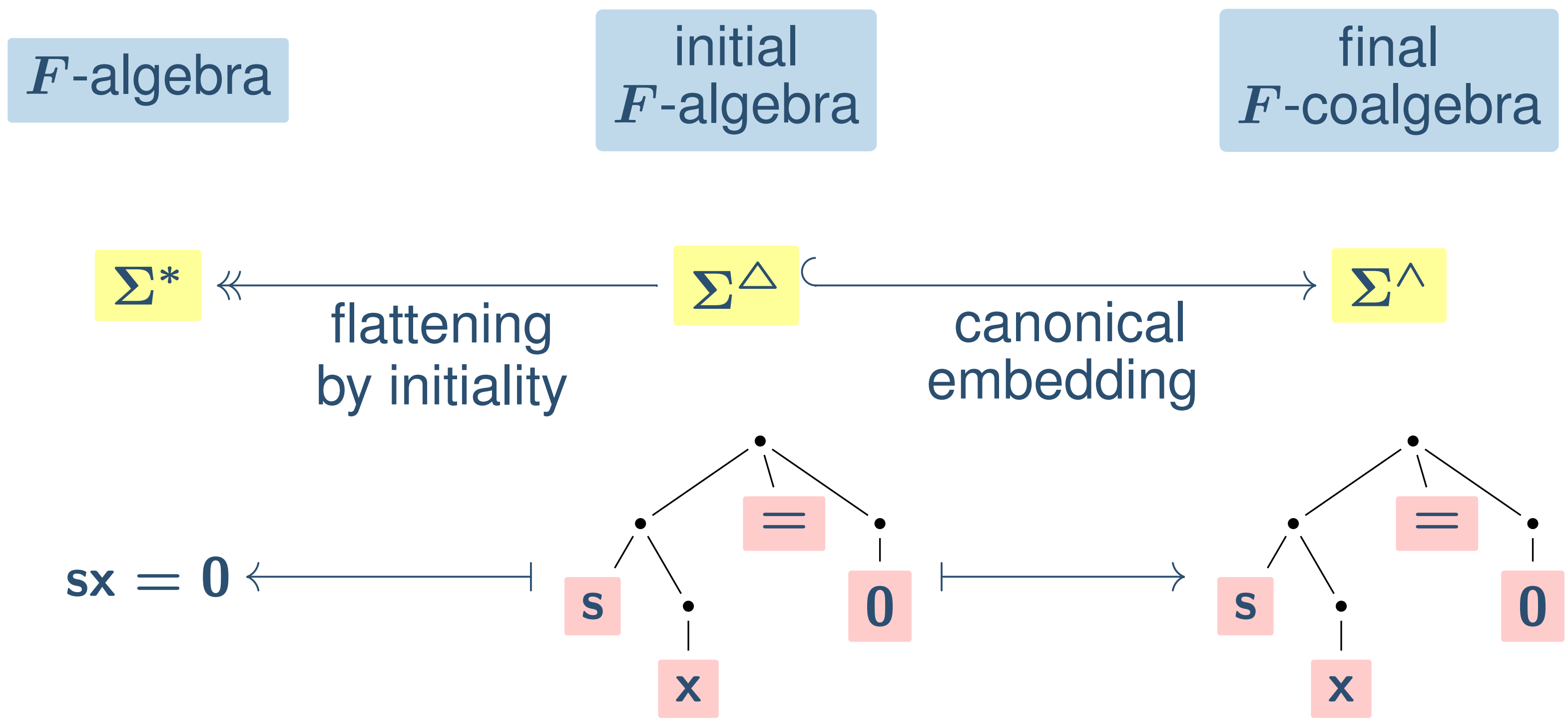


All of these constructions are **monads**:

$$\Sigma \mapsto \Sigma^* \qquad \Sigma \mapsto \Sigma^\triangle \qquad \Sigma \mapsto \Sigma^\wedge$$

Fundamental span of languages

For functor $F = (\Sigma + -)^*$,



Both arrows are **maps of monads**.

Back to coalgebraic trace semantics...

CFL via trace semantics

For $F = (\Sigma + -)^*$ and a CFG $\begin{array}{c}\mathcal{P}FX \\ c \uparrow \\ X\end{array}$,

- By finality result in $\mathbf{Sets}_{\mathcal{P}}$, we obtain

In Sets

$$X \xrightarrow{\mathsf{ft}(c)} \mathcal{P}(\Sigma^{\triangle})$$

$$x \longmapsto \{\text{generated parse trees from } x\}$$

- Generated language is obtained by flattening:

$$X \xrightarrow[\mathsf{ft}(c)]{} \mathcal{P}(\Sigma^{\triangle}) \xrightarrow[\mathcal{P}(\text{flattening})]{} \mathcal{P}(\Sigma^*)$$

Non-deterministic automata

Here is another application...

- Non-deterministic automaton:

$$\begin{array}{c} \mathcal{P}FX \\ \uparrow \\ X \end{array}$$

where $F = 1 + \Sigma \times -$,
list functor

- For $F = 1 + \Sigma \times -$, initial F-algebra is Σ^* (**lists**).

- By our finality result we obtain **accepted language**:

$$X \xrightarrow{\mathsf{ft}(c)} \mathcal{P}(\Sigma^*)$$

$$x \longmapsto \text{(language accepted by state } x\text{)}$$

Alternative construction

- Non-deterministic automaton:

$$\begin{array}{c} \mathcal{P}(1 + \Sigma \times X) = \mathbf{2} \times (\mathcal{P}X)^{\Sigma} \\ \uparrow \\ X \end{array}$$

- Bartels' **generalized coinduction** scheme [CMCS'01] applies to

$$\begin{array}{c} G\mathcal{P}X \\ \uparrow \\ X \end{array}$$

where $G = \mathbf{2} \times (-)^{\Sigma}$, **deterministic automaton** functor

- It **also** gives the map

$$\begin{array}{rcl} X & \longrightarrow & \mathcal{P}(\Sigma^*) \\ x & \longmapsto & \text{(language accepted by state } x\text{)} \end{array}$$

- Connection is to be investigated.

Trace semantics

for probabilistic systems

(in Proc. CALCO-jnr.)

Generalization (in Proc. of CALCO-jnr.)

The finality result above holds for

- powerset monad $\mathcal{P}$, and
- endofunctor F which is **shapely** [C. Barry Jay], i.e.

$$F ::= \mathrm{id} \mid \Sigma \mid F \times F \mid \coprod_i F_i$$

How about other monads/functors?

Currently we have obtained a result for

subdistribution monad $\mathcal{D}$ instead of $\mathcal{P}$

which gives trace distribution of probabilistic systems.
In Proc. of CALCO-jnr.

Subdistribution monad

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

Coalgebra $\begin{array}{c} \mathcal{P}FX \\ \uparrow \\ X \end{array}$: **Non-deterministic** system

Coalgebra $\begin{array}{c} \mathcal{D}FX \\ \uparrow \\ X \end{array}$: **Probabilistic** system

Trace semantics for probabilistic systems

Again we move to Kleisli category $\mathbf{Sets}_{\mathcal{D}}$:

$$\mathbf{Sets}_{\mathcal{D}} \left\{ \begin{array}{ll} \text{object} & \text{that of } \mathbf{Sets} \\ \text{arrow} & \dfrac{X \to Y \quad \text{in } \mathbf{Sets}_{\mathcal{D}}}{X \to \mathcal{D}Y \quad \text{in } \mathbf{Sets}} \end{array} \right.$$

Trace semantics for probabilistic systems

Let F be a shapely functor in $\mathbf{Sets}$.

Let $\begin{array}{c} FA \\ \cong \big\downarrow \alpha \\ A \end{array}$ be the initial F-algebra in $\mathbf{Sets}$.

Theorem

Coalgebra $\begin{array}{c} \mathcal{D}FA \\ \big\uparrow \eta^{\mathcal{D}} \\ FA \\ \big\uparrow \alpha^{-1} \\ A \end{array}$ in $\mathbf{Sets}$ i.e. $\begin{array}{c} FA \\ \Big\uparrow \\ A \end{array}$ in $\mathbf{Sets}_{\mathcal{D}}$ is final in $\mathbf{Sets}_{\mathcal{D}}$.

Trace semantics for probabilistic systems

Hence, a probabilistic coalgebra in $\mathbf{Sets}$

$$\begin{array}{c} \mathcal{D}FX \\ {\scriptstyle c}\uparrow \\ X \end{array}$$

has a unique **finite trace** $X \xrightarrow{\mathsf{ft}(c)} \mathcal{D}A$

such that:

In $\mathbf{Sets}_{\mathcal{D}}$

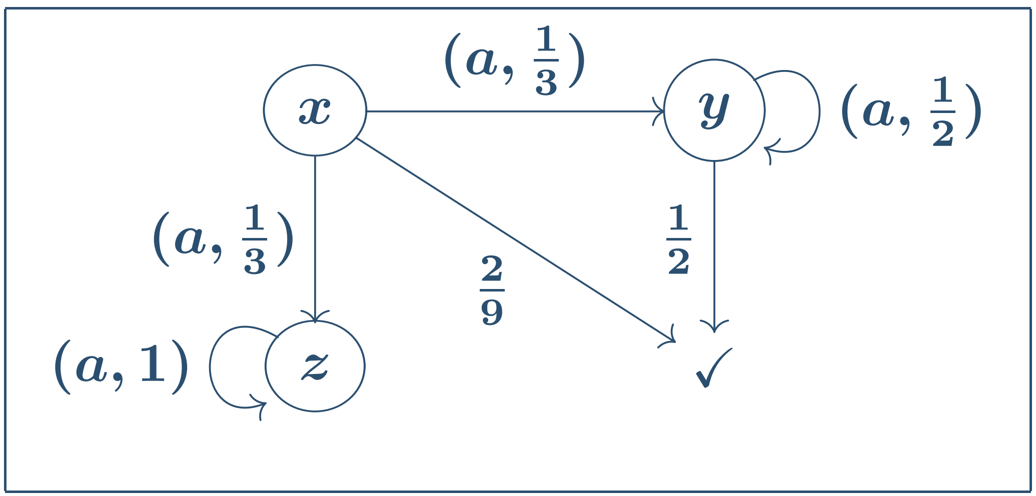
$$\begin{tikzcd} FX \arrow[r, dashed, "F(\mathsf{ft}(c))"] & FA \\ X \arrow[u, "c"] \arrow[r, dashed, "\mathsf{ft}(c)"'] & A \arrow[u, "\cong"', "\eta^{\mathcal{D}} \circ \alpha^{-1}"] \end{tikzcd}$$

Example: probabilistic systems

$$\begin{array}{c} \mathcal{D}FX \\ c\uparrow \\ X \end{array}$$

where $F = 1 + \Sigma \times -$,
list functor

Example



$$c(x) = \begin{bmatrix} \checkmark & \mapsto & 2/9 \\ (a, y) & \mapsto & 1/3 \\ (a, z) & \mapsto & 1/3 \end{bmatrix}$$

$$c(y) = \cdots$$

Remaining $1/9$ of $c(x)$: probability of **deadlock**

Example: probabilistic systems

- For $F = 1 + \Sigma \times -$, initial F-algebra is Σ^* (**lists**).

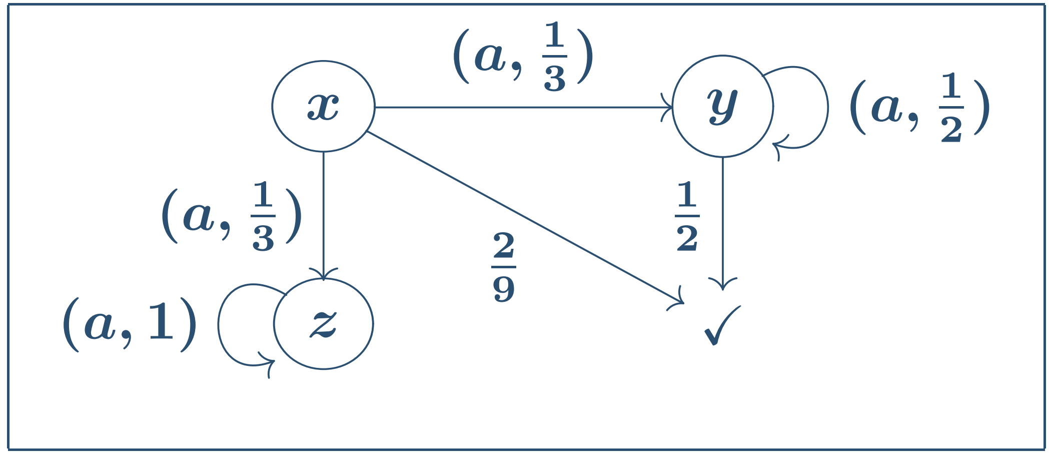
- Given $\begin{array}{c} \mathcal{D}FX \\ c\uparrow \\ X \end{array}$, by our finality result we obtain

$$\mathsf{ft}(c) : X \longrightarrow \mathcal{D}(\Sigma^*)$$

probability distribution on traces

Example: probabilistic systems

Example



$$[\![\mathbf{ft}(c)]\!](x) : \quad \langle\rangle \mapsto \frac{2}{9} \qquad a \mapsto \frac{1}{3}\cdot\frac{1}{2} \qquad a^2 \mapsto \frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{2} \qquad \cdots$$

Remaining of $[\![\mathbf{ft}(c)]\!](x)$ is understood:

$1/3$ for **livelock**, $1/9$ for deadlock

How does the proof generalize?

- For
  - **commutative** monad T and
  - shapely F,

  we have distributive law $FT \Rightarrow TF$.

- Distributive law $FT \Rightarrow TF$ lifts F to Kleisli category of T. [Power & Turi, CTCS'99]

- Subdistribution monad $\mathcal{D}$ is commutative.

How does the proof generalize?

- For **valuation monad** $\mathcal{V}X = \mathbb{R}^X$, the proof works.
  - $\mathbf{Sets}_{\mathcal{V}}$ is again self-dual.
  - Cf. $\mathcal{P}X = 2^X$

- $\mathcal{D} \hookrightarrow \mathcal{V}$, **"nice"** submonad.
  For example, distributive law $F\mathcal{V} \Rightarrow \mathcal{V}F$ restricts to $\mathcal{D}$:

$$
\begin{tikzcd}
F\mathcal{V} \arrow[r, Rightarrow] & \mathcal{V}F \\
F\mathcal{D} \arrow[u, hook] \arrow[r, Rightarrow] & \mathcal{D}F \arrow[u, hook]
\end{tikzcd}
$$

  How "nice"? Yet to be investigated.

Future work

- Other monads.

  Especially $\mathcal{PD}$ (with modification, [Varacca, LICS'02]), which is of interest in concurrency theory.

- Other functors.

- Other base categories. On arbitrary toposes?

- CFG/CFL and **pushdown automata**.

- **Parsing**, which is (partial) inverse of flattening $\Sigma^{\triangle} \twoheadrightarrow \Sigma^{*}$.

Conclusions

- Initial algebra in $\mathbf{Sets}$ yields final coalgebra in $\mathbf{Rel} = \mathbf{Sets}_{\mathcal{P}}$.
  - Trace semantics for **non-deterministic** systems
  - CFG/CFL, non-deterministic automata
- Initial algebra in $\mathbf{Sets}$ yields final coalgebra in $\mathbf{Sets}_{\mathcal{D}}$.
  - Trace semantics for **probabilistic** systems
- (Co)algebraic structures in CFG/CFL.

Conclusions

- Initial algebra in $\mathbf{Sets}$ yields final coalgebra in $\mathbf{Rel} = \mathbf{Sets}_{\mathcal{P}}$.
  - Trace semantics for **non-deterministic** systems
  - CFG/CFL, non-deterministic automata
- Initial algebra in $\mathbf{Sets}$ yields final coalgebra in $\mathbf{Sets}_{\mathcal{D}}$.
  - Trace semantics for **probabilistic** systems
- (Co)algebraic structures in CFG/CFL.
- The authors make each other depressed from time to time.

Thank you for your attention!

Contact: **Ichiro Hasuo** **www.cs.ru.nl/∼ichiro** **ichiro@cs.ru.nl**