A nondeterministic probabilistic monad with nondeterminism for coalgebraic trace semantics

Tetsuya Sato

Research Institute for Mathematical Sciences, Kyoto University

August 29, 2011

Introduction(1/2)

Goal: To express the finite trace semantics of probabilistic automata by Segala in a coalgebraic way.

- In the study of trace semantics by Hasuo, Jacobs and Sokolova.

(finite) trace semantics of nondeterministic system or probabilistic system	$\rightsquigarrow$	colagebra morphism (coinduction) in Kleisli category

- We want to consider *probabilistic systems with nondeterminism.*,

Introduction(2/2)

- To consider probabilistic systems with nondeterminism, we want to do as follows.

A nondeterministic monad A probabilistic monad	$\rightsquigarrow$ compose	A probabilistic monad with nondeterminism

However, we cannot do this in a natural way.

- We constructed coalgebraic a trace semantics which is similar to the original trace semantics.

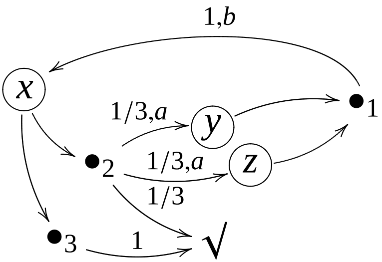
Probabilistic automata (Segala)

A probabilistic automaton is a tuple $(B, A, \mathbf{start}, \mathbf{trans})$

- B: states, A: labels/actions, $\mathbf{start} \subseteq B$: initial states
- $\mathbf{trans}\colon B \to \mathcal{P}\mathcal{D}(\{\surd\} + (A \times B))$: transitions

$$\mathcal{P}X := \left\{ Y \mid Y \subseteq X \right\}$$

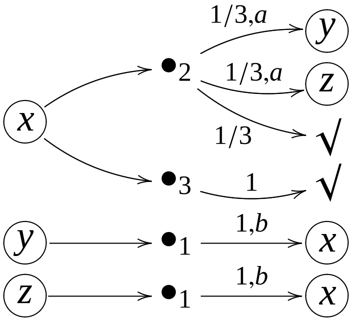
$$\mathcal{D}X := \left\{ d\colon X \to [0,1] \mid \textstyle\sum_{x \in X} d(x) \leq 1 \right\}$$



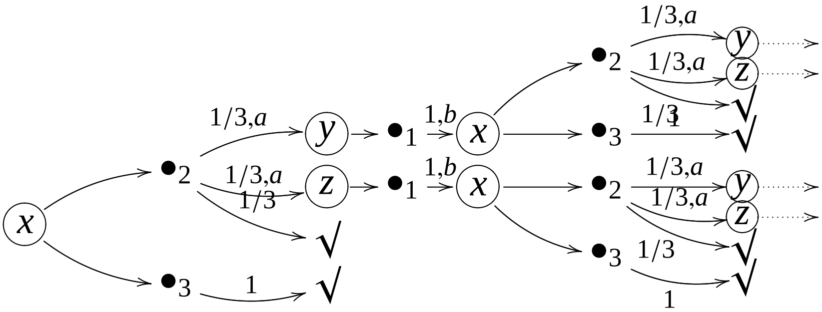
An example of probabilistic automaton

Transitions of probabilistic automata

One-step transition of “probabilistic automata” is Nondeterministic transition + Probabilistic transition.



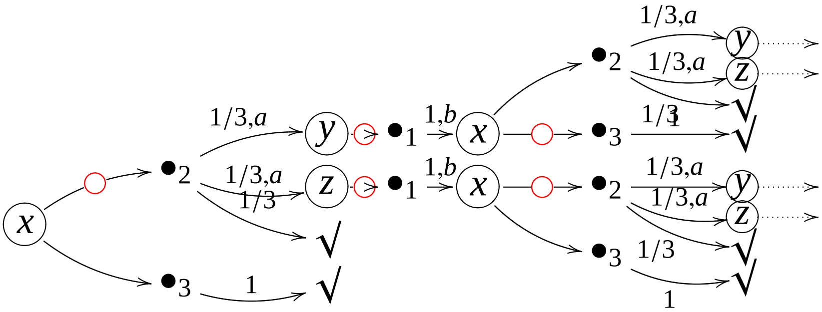
Trace semantics: notion of schedulers



A tree of transitions starting from x

Schedulers are defined as *choices of nondeterministic branches*. Each scheduler is a function.

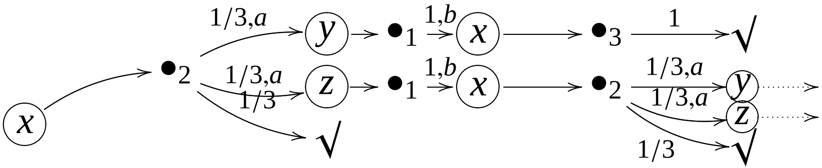
Trace semantics: notion of schedulers



Choices by a scheduler starting from x

Schedulers are defined as *choices of nondeterministic branches*. Each scheduler is a function.

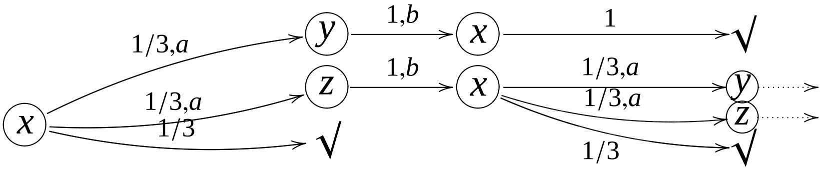
Execution fragment for scheduler



An execution fragment for *a scheduler*

Each scheduler gives a probabilistic automata *without* nondeterminism (an execution fragment).

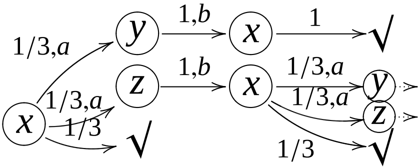
Execution fragment for scheduler



An execution fragment for *a scheduler*

Each scheduler gives a probabilistic automata *without* nondeterminism (an execution fragment).

Trace semantics: Traces for schedulers



$$\Rightarrow \begin{cases} \langle\rangle & \mapsto \frac{1}{3} \\ \cdots & \\ ab & \mapsto \frac{1}{3} + \frac{1}{9} \\ \cdots & \\ abab & \mapsto \cdots \\ \cdots & \end{cases}$$

A finite trace for an execution fragment

For each execution fragment, we can define a finite trace.

Each execution fragment for scheduler gives finite

Trace semantics: Definition

On an probabilistic automaton M,

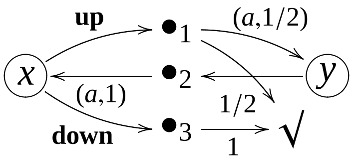
$$\underset{\text{scheduler}}{f} \longrightarrow \underset{\text{execution fragment}}{M_f} \longrightarrow \underset{\text{finite trace}}{\mathbf{tdistr}(M_f)}$$

The finite trace semantics of a probabilistic automaton is defined as follows:

$$\mathbf{Tr}(x) = \left\{ \mathbf{tdistr}(M_f) \;\middle|\; f \text{ is a scheduler starting from } x \right\}$$

Another example of trace semantics

The schedulers (from x) can be written as sequences in $\{\mathbf{up}, \mathbf{down}\}^\omega$.



schedulers	trace for scheduler (omitting $\cdots \mapsto 0$)
$\mathbf{down}, \sigma$	$\langle\rangle \mapsto 1$
$\mathbf{up}, \mathbf{down}, \sigma$	$\langle\rangle \mapsto 1/2, aa \mapsto 1/2$
$\vdots$	$\vdots$
$\mathbf{up}^n, \mathbf{down}, \sigma$	$\langle\rangle \mapsto 1/2, aa \mapsto 1/4, \dots a^{2n} \mapsto 1/2^{n+1}, a^{2^{(n+1)}} \mapsto 1/2^{n+1}$
$\vdots$	$\vdots$
Forever $\mathbf{up}$	$\langle\rangle \mapsto 1/2, aa \mapsto 1/4, \dots a^{2n} \mapsto 1/2^{(n+1)}, \dots$

$(\sigma \in \{\mathbf{up}, \mathbf{down}\}^\omega)$

The trace semantics on x

How is coalgebraic trace?

Probabilistic automata are $\mathcal{PD}F$-coalgebras with $F = 1 + (A \times id)$.
To construct coalgebraic trace semantics, we need $\mathcal{PD}$ to be a monad whose unit is

$$\eta^{\mathcal{PD}}(x) = \{\chi_x\} = \eta^{\mathcal{P}} \circ \eta^{\mathcal{D}}(x)$$

$$\text{where, } \chi_x(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

However, there is no distributive law $\mathcal{DP} \Rightarrow \mathcal{PD}$. Thus $\mathcal{PD}$ can not be a monad with the above unit [Beck].
Therefore, we have to construct a probabilistic monad with nondeterminism in other way.

Indexed distribution monad

Based on the indexed valuation monad by Varacca and Winskel, we define the monad captures that probabilistic transition with nondeterminism.

An *indexed distribution* on a set X is a pair $(\mathbf{Ind}, v)$ of two functions

- $\mathbf{Ind}\colon I \to X$
- $v\colon I \to [0,\infty]$ such that $\sum_{i \in I} v(i) \leq 1$.

And we define,

- support: $\mathrm{Supp}(v) = \{i \in I | v(i) \neq 0\}$
- zero $\underline{0}$: (Ind, v) such that $v = 0$

Indexed distributions are pre-ordered as follows:

$$(\mathrm{Ind}, v) \sqsubseteq (\mathrm{Ind}', v')$$

if there is an *injection* $h\colon \mathrm{Supp}(v) \to \mathrm{Supp}(v')$ such that

$$\begin{aligned} \mathrm{Ind}(i) &= \mathrm{Ind}'(h(i)) \\ v(i) &= v'(h(i)) \text{ for all } i \in \mathrm{Supp}(v) \end{aligned}$$

We then define $\mathbb{D}(X)$ by

$$\mathbb{D}(X) = \{(\mathrm{Ind}, v)\colon \text{indexed distributions on } X\}/\sim$$

where $\sim$ is the eqivalence relation generated from $\sqsubseteq$.
We then define an partial order $\sqsubseteq_{\mathbb{D}(X)}$ on $\mathbb{D}(X)$.

Proposition

For any set X, ordered set $(\mathbb{D}(X), \sqsubseteq_{\mathbb{D}(X)})$ is ω-complete with the smallest element $\underline{0}$.

We obtain a monad $\mathbb{D}$.

Proposition

$\mathbb{D}$ forms a set functor.

$$\mathbb{D}(f)((\mathbf{Ind}, v)) := (f \circ \mathbf{Ind}, v)$$

Functor $\mathbb{D}$ is extended to a monad $(\mathbb{D}, \eta^{\mathbb{D}}, \mu^{\mathbb{D}})$.

$$\begin{aligned} \eta^{\mathbb{D}}_X(x) &:= (\mathrm{Ind} \colon * \mapsto x, v \colon * \mapsto 1) \\ \mu^{\mathbb{D}}_X((\mathrm{Ind}, v)) &:= ([\mathrm{Ind}_i]_{i \in I}, [v(i) \cdot w_i]_{i \in I}) \\ & \quad \textit{where, } \mathrm{Ind}(i) = (\mathrm{Ind}_i, w_i) \end{aligned}$$

For any $f_i \colon I_i \to Y (i \in I)$, $[f_i]_{i \in I} \colon \coprod_{i \in I} \to Y$ is defined as $[f_i]_{i \in I}(j) = f_i(j)$.

A probabilistic monad with nondeterminism

The nonempty powerset monad is denoted by $\mathcal{P}_+$. Composing $\mathcal{P}_+$ and $\mathbb{D}$, we construct a probabilistic monad with nondeterminism.

Proposition

There is a distributive law $d\colon \mathbb{D}\mathcal{P}_+ \Rightarrow \mathcal{P}_+\mathbb{D}$ *of* $\mathbb{D}$ *over* $\mathcal{P}_+$*.*
For any $(\mathrm{Ind}, d) \in \mathbb{D}(\mathcal{P}_+(X))$*,*

$$d_X((\mathrm{Ind}, v)) := \left\{ (h, v) \,\middle|\, h\colon I \to X,\ h(i) \in \mathrm{Ind}(i) \right\}$$

In fact, the composite monad $\mathcal{P}_+\mathbb{D}$ is commutative. Thus, any polynomial functor F has a lifting $\overline{F}$ in $\mathbf{Set}_{\mathcal{P}_+\mathbb{D}}$.

Probabilistic automata as coalgebras

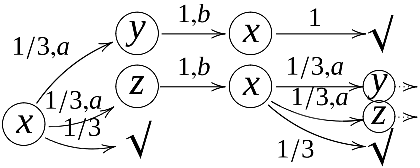
Now, let F be a polynomial functor $1 + (A \times id)$ on **Set** (A is the set of labels/actions).
For a probabilistic automaton $c\colon X \to \mathcal{PD}FX$, $\overline{F}$-coalgebra c' is defined as follows:

$$c'(x) = \begin{cases} \{\underline{0}\} & \text{if } c(x) = \emptyset \\ c(x) & \text{if } c(x) \neq \emptyset \end{cases}$$

When $c(x)$ is not empty, any $f \in c(x)\colon \mathcal{D}(X)$ is regarded as a probabilistic indexed distribution $(id_X\colon X \to X, f\colon X \to [0,\infty]) \in I\!D(X)$.

Segala's construction of trace semantics

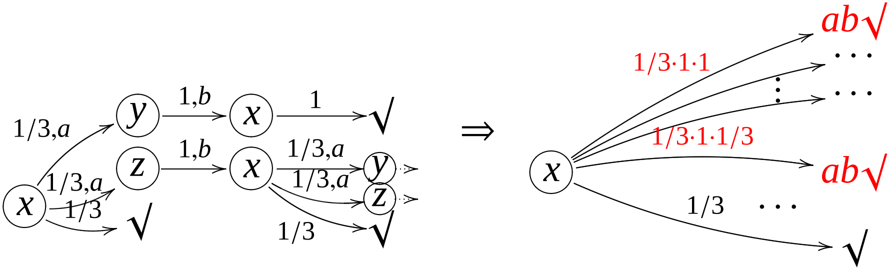
On Segala's trace semantics, two or more path to $\surd$ are marged when each of them have the same sequence of actions and termination.



$$\Rightarrow \begin{cases} \langle\rangle & \mapsto \frac{1}{3} \\ \cdots & \\ ab & \mapsto \frac{1}{3} + \frac{1}{9} \\ \cdots & \\ abab & \mapsto \cdots \\ \cdots & \end{cases}$$

Our construction of trace semantics

In our trace semantics *all paths are distinguished*:



This semantics is expressed as an arrow $X \to A^*$ in Kleisli category $\mathbf{Set}_{\mathcal{P}_+\mathbb{D}}$.

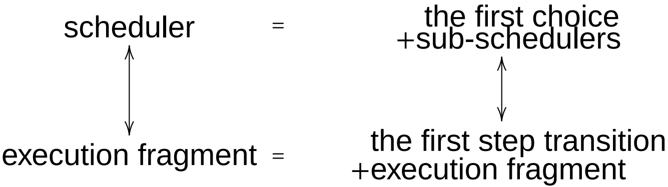
The main result 1

Trace semantics of probabilistic automata are captured as a coalgebra morphisms in Kleisli category $\mathbf{Set}_{\mathcal{P}_+\mathbb{D}}$.

Theorem

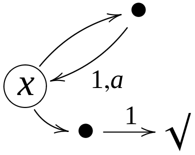
For any $\overline{F}$-coalgebra $c\colon X \to \overline{F}X$, the trace semantics $\mathbf{Tr}^c$ is an $\overline{F}$-coalgebra morphism $\mathbf{Tr}^c\colon c \to \eta^T_{FA^*} \circ [Nil, Cons]^{-1}$.

A key point is the following diagram.



Failure of finality

$\overline{F}$-coalgebra $\eta^T_{FA^*} \circ [Nil, Cons]^{-1}$ is weakly final, but *not final*. For example, consider an automaton as follows:



There are *two* coalgebra morphisms $f_1, f_2 \colon c \to \eta^T_{FA^*} \circ [Nil, Cons]^{-1}$

$$f_1(x) = \left\{\underline{0}, (\langle\rangle, 1), (a, 1), (aa, 1), \ldots, (a^k, 1), \ldots\right\}$$
$$f_2(x) = \left\{(\langle\rangle, 1), (a, 1), (aa, 1), \ldots, (a^k, 1), \ldots\right\}$$

The main result 2

However, the coalgebra $\eta^T_{FA^*} \circ [Nil, Cons]^{-1}$ has a nice property.

Theorem

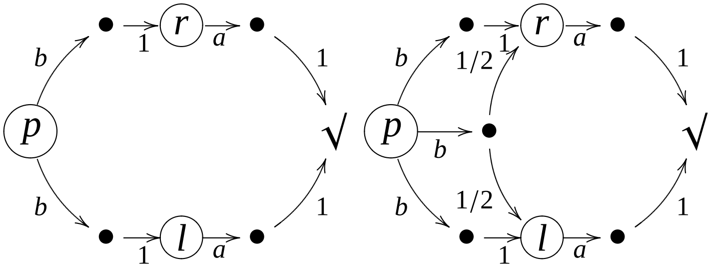
Whenever $f : c \to \eta^T_{FA^*} \circ [Nil, Cons]^{-1}$ is an $\overline{F}$-coalgebra morphism, $f(x) \subseteq \mathbf{Tr}^c(x)$ holds for any $x \in X$.

Key points are the following:

- Since f is a coalgebra morphism, each $(\mathbf{Ind}, v) \in f(x)$ can be decomposed into one-step transition and a collecton of $(\mathbf{Ind}_i, v_i) \in f(x_i)$.
- Inductively, each $(\mathbf{Ind}, v) \in f(x)$ gives a scheduler and an execution fragment whose trace is $(\mathbf{Ind}, v)$.

Future Work

- To generalize this work to any polynomial functor F.
- To compare this work and Jacobs' work which is based on convex subsets of distributions. I guess ours is detailed.



Monad $\mathcal{P}_+\mathbb{D}$ forces to distinguish the above automata.

Conclusion

- We constructed a probabilistic monad with nondeterminism based on indexed valuation by Varacca and Winskel.
- We constructed trace semantics of probabilistic automata with simple translation. This trace semantics is coalgebraic.
- And our construction of trace semantics is similar to the original construction by Segala.