Some CPO Results for Simple Probabilistic Automata

Ling Cheung

lcheung@cs.kun.nl

University of Nijmegen

The Netherlands

Overview

I. Definitions for Probabilistic Automata

Overview

I. Definitions for Probabilistic Automata

II. Our motivation

Overview

I. Definitions for Probabilistic Automata

II. Our motivation

III. CPO Definitions

Overview

I. Definitions for Probabilistic Automata

II. Our motivation

III. CPO Definitions

IV. Main Theorems

Overview

I. Definitions for Probabilistic Automata
II. Our motivation
III. CPO Definitions
IV. Main Theorems
V. Future Work

Probabilistic Automata

Simple probabilistic automaton:

$$\langle \mathsf{States}(A),\ \mathsf{root}(\mathrm{A}),\ \rightarrow \rangle,$$

where $\rightarrow \subseteq \mathsf{States}(A) \times Act \times \mathsf{Disc}(A)$

Probabilistic Automata

Simple probabilistic automaton:

$$\langle \mathsf{States}(A),\ \mathsf{root}(\mathrm{A}),\ \rightarrow \rangle,$$

where $\rightarrow \subseteq \mathsf{States}(A) \times Act \times \mathsf{Disc}(A)$

Finite execution: $s_0 a_1 \mu_1 s_1 \ldots a_n \mu_n s_n,$

Probabilistic Automata

Simple probabilistic automaton:

$$\langle \mathsf{States}(A),\ \mathsf{root}(\mathrm{A}),\ \rightarrow \rangle,$$

where $\rightarrow \subseteq \mathsf{States}(A) \times Act \times \mathsf{Disc}(A)$

Finite execution: $s_0 a_1 \mu_1 s_1 \ldots a_n \mu_n s_n$,
provided:

- $s_0 = \mathsf{root}(\mathrm{A})$,
- $s_i \stackrel{a_{i+1}}{\rightarrow} \mu_{i+1}$, and
- $s_{i+1} \in \mathsf{Supp}(\mu_{i+1})$ for all i.

Probabilistic Automata

Simple probabilistic automaton:

$$\langle \mathsf{States}(A),\ \mathsf{root}(\mathrm{A}),\ \rightarrow \rangle,$$

where $\rightarrow \subseteq \mathsf{States}(A) \times Act \times \mathsf{Disc}(A)$

Finite execution: $s_0 a_1 \mu_1 s_1 \ldots a_n \mu_n s_n$,
provided:

- $s_0 = \mathsf{root}(\mathrm{A})$,
- $s_i \stackrel{a_{i+1}}{\rightarrow} \mu_{i+1}$, and
- $s_{i+1} \in \mathsf{Supp}(\mu_{i+1})$ for all i.

Notation: $\mathsf{Exec}_{\mathsf{fin}}(A)$

Schedulers

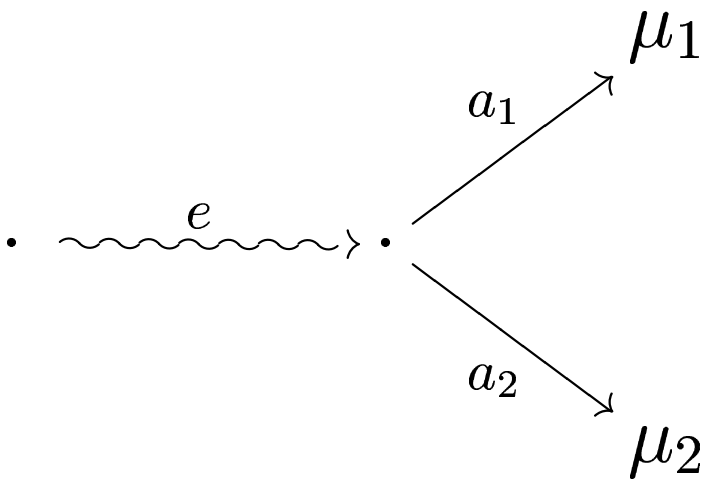
$E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow \mathsf{SubDisc}(Act \times \mathsf{Disc}(\mathsf{States}(A)))$
satisfying $\forall e \in \mathsf{Exec}_{\mathsf{fin}}(A)$,

$$\langle a, \mu \rangle \in \mathsf{Supp}(E)(e) \Rightarrow \mathsf{last}(e) \xrightarrow{a} \mu.$$

Schedulers

$E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow \mathsf{SubDisc}(Act \times \mathsf{Disc}(\mathsf{States}(A)))$
satisfying $\forall e \in \mathsf{Exec}_{\mathsf{fin}}(A)$,

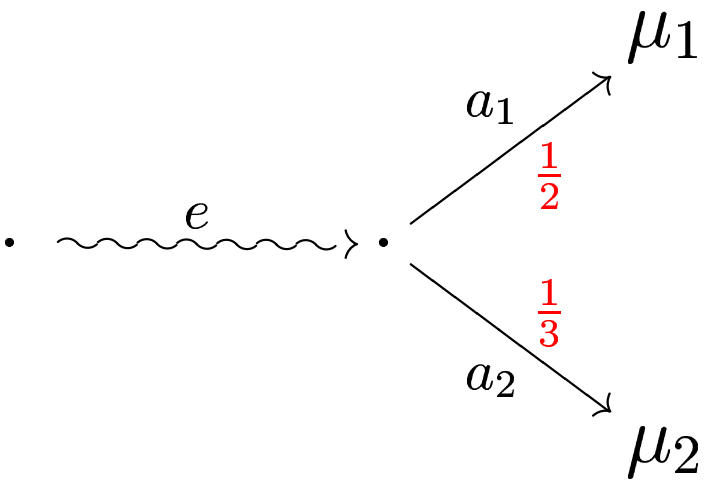
$$\langle a, \mu \rangle \in \mathsf{Supp}(E)(e) \Rightarrow \mathsf{last}(e) \xrightarrow{a} \mu.$$



Schedulers

$E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow \mathsf{SubDisc}(Act \times \mathsf{Disc}(\mathsf{States}(A)))$
satisfying $\forall e \in \mathsf{Exec}_{\mathsf{fin}}(A)$,

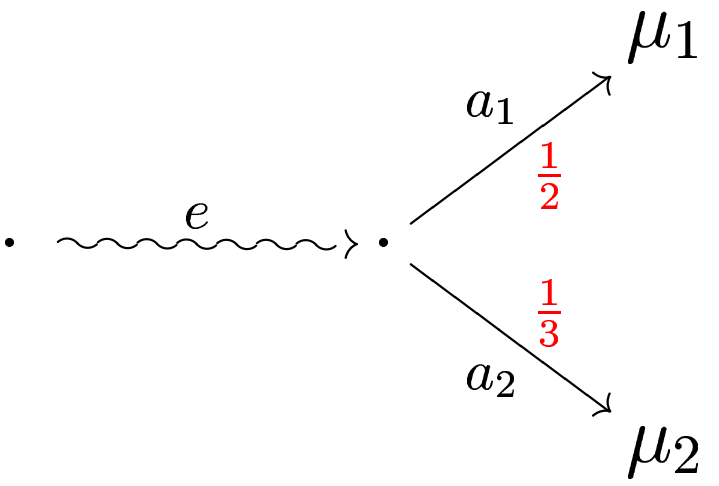
$$\langle a, \mu \rangle \in \mathsf{Supp}(E)(e) \Rightarrow \mathsf{last}(e) \xrightarrow{a} \mu.$$



Schedulers

$E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow \mathsf{SubDisc}(Act \times \mathsf{Disc}(\mathsf{States}(A)))$
satisfying $\forall e \in \mathsf{Exec}_{\mathsf{fin}}(A)$,

$$\langle a, \mu \rangle \in \mathsf{Supp}(E)(e) \Rightarrow \mathsf{last}(e) \xrightarrow{a} \mu.$$



Notation: $\mathsf{Sched}(A)$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a, \mu\rangle \cdot \mu(s)$$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a, \mu\rangle \cdot \mu(s)$$

Example: $s_0 a_1 \mu_1 s_1 a_2 \mu_2 s_2 \ldots$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a,\ \mu\rangle \cdot \mu(s)$$

Example: $s_0 a_1 \mu_1 s_1 a_2 \mu_2 s_2 \ldots$

$$1 \xrightarrow{\frac{1}{2}} \frac{1}{2} \longrightarrow \frac{1}{4} \longrightarrow \frac{1}{8} \longrightarrow \cdots$$

$$E(s_0)\langle a_1,\ \mu_1\rangle = \frac{1}{2}$$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a, \mu\rangle \cdot \mu(s)$$

Example: $s_0 a_1 \mu_1 s_1 a_2 \mu_2 s_2 \ldots$

$$1 \xrightarrow{\frac{1}{2}} \frac{1}{2} \xrightarrow{\frac{1}{2}} \frac{1}{4} \longrightarrow \frac{1}{8} \longrightarrow \cdots$$

$$\mu_1(s_1) = \frac{1}{2}$$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a,\, \mu\rangle \cdot \mu(s)$$

Example: $s_0 a_1 \mu_1 s_1 a_2 \mu_2 s_2 \ldots$

$$1 \xrightarrow{\frac{1}{2}} \frac{1}{2} \xrightarrow{\frac{1}{2}} \frac{1}{4} \xrightarrow{\frac{1}{2}} \frac{1}{8} \longrightarrow \cdots$$

$$E(s_0 a_1 \mu_1 s_1)\langle a_2,\, \mu_2\rangle = \frac{1}{2}$$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a, \mu\rangle \cdot \mu(s)$$

Example: $s_0 a_1 \mu_1 s_1 a_2 \mu_2 s_2 \ldots$

$$1 \xrightarrow{\frac{1}{2}} \frac{1}{2} \xrightarrow{\frac{1}{2}} \frac{1}{4} \xrightarrow{\frac{1}{2}} \frac{1}{8} \xrightarrow{\frac{1}{2}} \ldots$$

$$\mu_2(s_2) = \frac{1}{2}$$

Probabilistic Executions

The probabilistic execution induced by E:
$Q_E : \mathsf{Exec}_{\mathsf{fin}}(A) \rightarrow [0,1]$

$$Q_E(\mathsf{root}(\mathrm{A})) := 1;$$
$$Q_E(ea\mu s) := Q_E(e) \cdot E(e)\langle a, \mu\rangle \cdot \mu(s)$$

Example: $s_0 a_1 \mu_1 s_1 a_2 \mu_2 s_2 \ldots$

$$1 \xrightarrow{\frac{1}{2}} \frac{1}{2} \xrightarrow{\frac{1}{2}} \frac{1}{4} \xrightarrow{\frac{1}{2}} \frac{1}{8} \xrightarrow{\frac{1}{2}} \ldots$$

Notation: $\mathsf{ProbExec}(A) := \mathsf{Im}_Q(\mathsf{Sched}(A)))$

Pause...

Where is the σ-field generated by cones of executions?
And the probabilistic measure generated by Q_E?

Pause...

Where is the σ-field generated by cones of executions?
And the probabilistic measure generated by Q_E?

Answer: In the background.

Trace Distributions

The trace distribution induced by E:

$$\begin{aligned} \mathsf{td}(E) &: Act^* \to [0,1] \\ &: \alpha \mapsto \sum_{e \in \mathsf{tr}^{-1}(\alpha)} Q_E(e) \end{aligned}$$

Trace Distributions

The trace distribution induced by E:

$$\begin{aligned} \mathsf{td}(E) &: Act^* \to [0,1] \\ &: \alpha \mapsto \sum_{e \in \mathsf{tr}^{-1}(\alpha)} Q_E(e) \end{aligned}$$

Notation: $\mathsf{TrDist}(A) := \mathsf{Im}_{\mathsf{td}}(\mathsf{Sched}(A))$

Trace Distributions

The trace distribution induced by E:

$$\begin{aligned} \mathsf{td}(E) : Act^* &\to [0,1] \\ &: \alpha \mapsto \sum_{e \in \mathsf{tr}^{-1}(\alpha)} Q_E(e) \end{aligned}$$

Notation: $\mathsf{TrDist}(A) := \mathsf{Im}_{\mathsf{td}}(\mathsf{Sched}(A))$

$$\mathsf{Sched}(A) \xrightarrow{Q} \mathsf{ProbExec}(A) \xrightarrow{\mathsf{tr}} \mathsf{TrDist}(A)$$

$$\mathsf{Sched}(A) \xrightarrow{\mathsf{td}=\mathsf{tr}\circ Q} \mathsf{TrDist}(A)$$

Our Motivation

To better understand trace-style semantics for probabilistic automata.

- Closure under limits (definition principle)
- Distinction between finitary and infinitary objects
- Infinitary objects generated by finitary ones (proof principle)
- Limit-preserving operators (reasoning between semantic domains)

Our Motivation

To better understand trace-style semantics for probabilistic automata.

- Closure under limits (definition principle)
- Distinction between finitary and infinitary objects
- Infinitary objects generated by finitary ones (proof principle)
- Limit-preserving operators (reasoning between semantic domains)

Our Motivation

To better understand trace-style semantics for probabilistic automata.

- Closure under limits (definition principle)
- Distinction between finitary and infinitary objects
- Infinitary objects generated by finitary ones (proof principle)
- Limit-preserving operators (reasoning between semantic domains)

Our Motivation

To better understand trace-style semantics for probabilistic automata.

- Closure under limits (definition principle)
- Distinction between finitary and infinitary objects
- Infinitary objects generated by finitary ones (proof principle)
- Limit-preserving operators (reasoning between semantic domains)

CPO Definitions

A partially ordered set P is a CPO (complete partial order) if P has

- a bottom element, and
- least upper bound for every ordinal indexed chain.

CPO Definitions

A partially ordered set P is a CPO (complete partial order) if P has

- a bottom element, and
- least upper bound for every ordinal indexed chain.

CPO Definitions

A partially ordered set P is a CPO (complete partial order) if P has

- a bottom element, and
- least upper bound for every ordinal indexed chain.

An operator $F : P \to Q$ between CPO's is

- strict: bottom-preserving;
- continuous: limit-preserving.

CPO Definitions

A partially ordered set P is a CPO (complete partial order) if P has

- a bottom element, and
- least upper bound for every ordinal indexed chain.

An operator $F : P \to Q$ between CPO's is

- strict: bottom-preserving;
- continuous: limit-preserving.

CPO Definitions

A partially ordered set P is a CPO (complete partial order) if P has

- a bottom element, and
- least upper bound for every ordinal indexed chain.

An operator $F : P \to Q$ between CPO's is

- strict: bottom-preserving;
- continuous: limit-preserving.

Algebraic CPO: infinitary elements generated by finitary ones.

Proof Methods

(Algebraic) CPO properties provide proof methods for many automata-based models.

Proof Methods

(Algebraic) CPO properties provide proof methods for many automata-based models.

- ND automata: executions with prefix ordering
- Hybrid automata: (Lynch and Vaandrager, 2002)
  - Hybrid sequences form an algebraic CPO;
  - Compositionality and soundness of simulations.
- Probabilistic automata: (Stoelinga and Vaandrager, 2001)
  - Approximation Induction Principle;
  - Testing equivalence coincides with trace distribution equivalence.

Proof Methods

(Algebraic) CPO properties provide proof methods for many automata-based models.

- ND automata: executions with prefix ordering
- Hybrid automata: (Lynch and Vaandrager, 2002)
  - Hybrid sequences form an algebraic CPO;
  - Compositionality and soundness of simulations.
- Probabilistic automata: (Stoelinga and Vaandrager, 2001)
  - Approximation Induction Principle;
  - Testing equivalence coincides with trace distribution equivalence.

Proof Methods

(Algebraic) CPO properties provide proof methods for many automata-based models.

- ND automata: executions with prefix ordering
- Hybrid automata: (Lynch and Vaandrager, 2002)
  - Hybrid sequences form an algebraic CPO;
  - Compositionality and soundness of simulations.
- Probabilistic automata: (Stoelinga and Vaandrager, 2001)
  - Approximation Induction Principle;
  - Testing equivalence coincides with trace distribution equivalence.

Theorems for Pointwise Ordering

Assumptions	CPO's	Continuous Map
ω-branching	$\mathsf{Sched}(A)$, $\mathsf{ProbExec}(A)$	Q
ω-branching,image finite	$\mathsf{TrDist}(A)$	tr, td

$$\mathsf{Sched}(A) \xrightarrow{Q} \mathsf{ProbExec}(A)$$

Theorems for Pointwise Ordering

Assumptions	CPO's	Continuous Map
ω-branching	$\mathsf{Sched}(A)$, $\mathsf{ProbExec}(A)$	Q
ω-branching,image finite	$\mathsf{TrDist}(A)$	tr, td

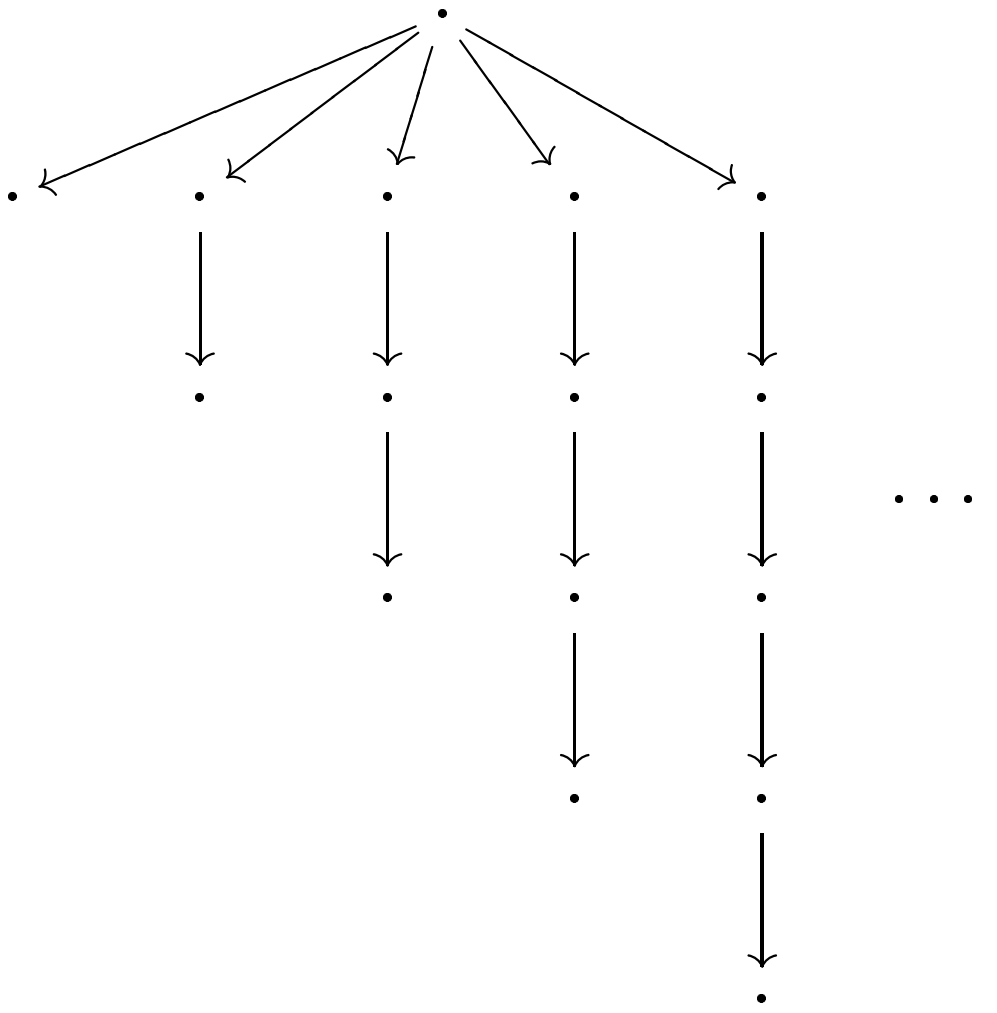
image finite: $\{\langle a, \mu\rangle \mid s \xrightarrow{a} \mu\}$ finite for all s.

$$\mathsf{Sched}(A) \xrightarrow{Q} \mathsf{ProbExec}(A) \xrightarrow{\mathsf{tr}} \mathsf{TrDist}(A)$$

$$\mathsf{td} = \mathsf{tr} \circ Q$$

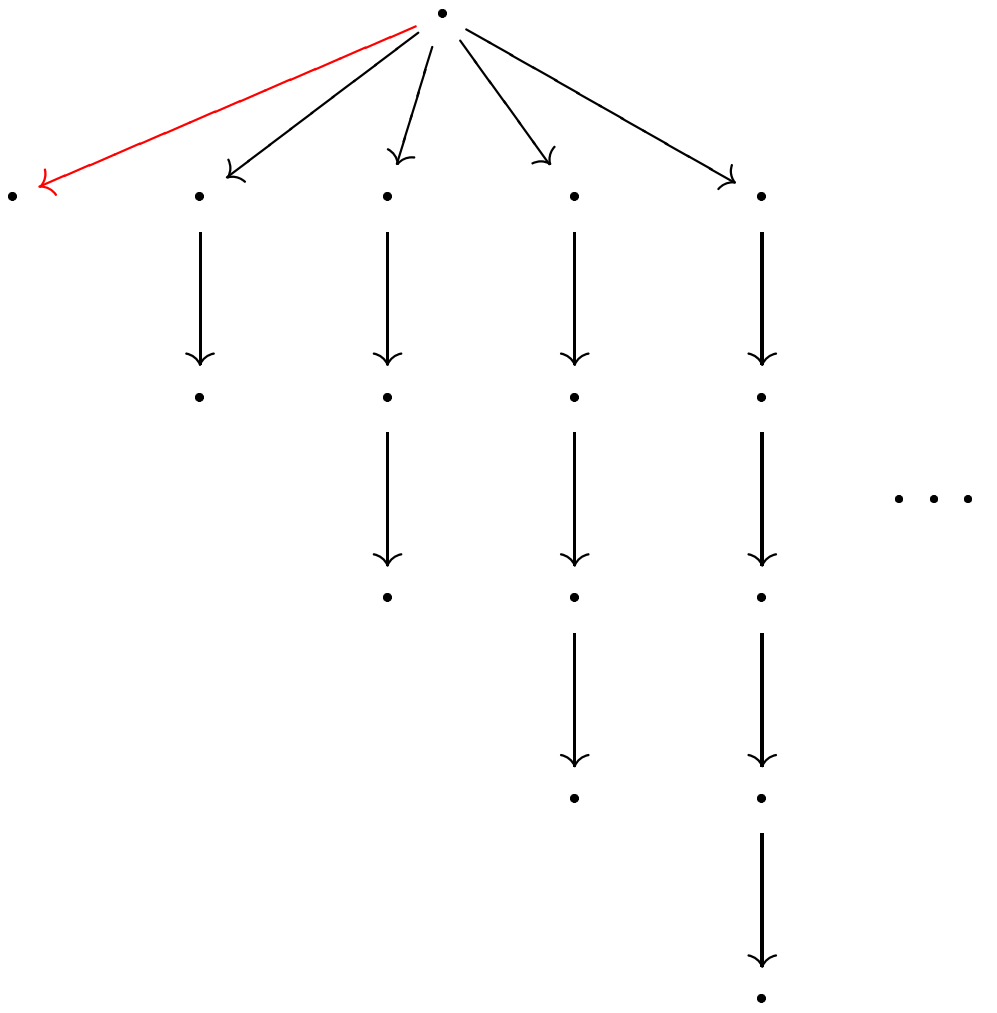
Counterexample

Consider the following non-image finite automaton.



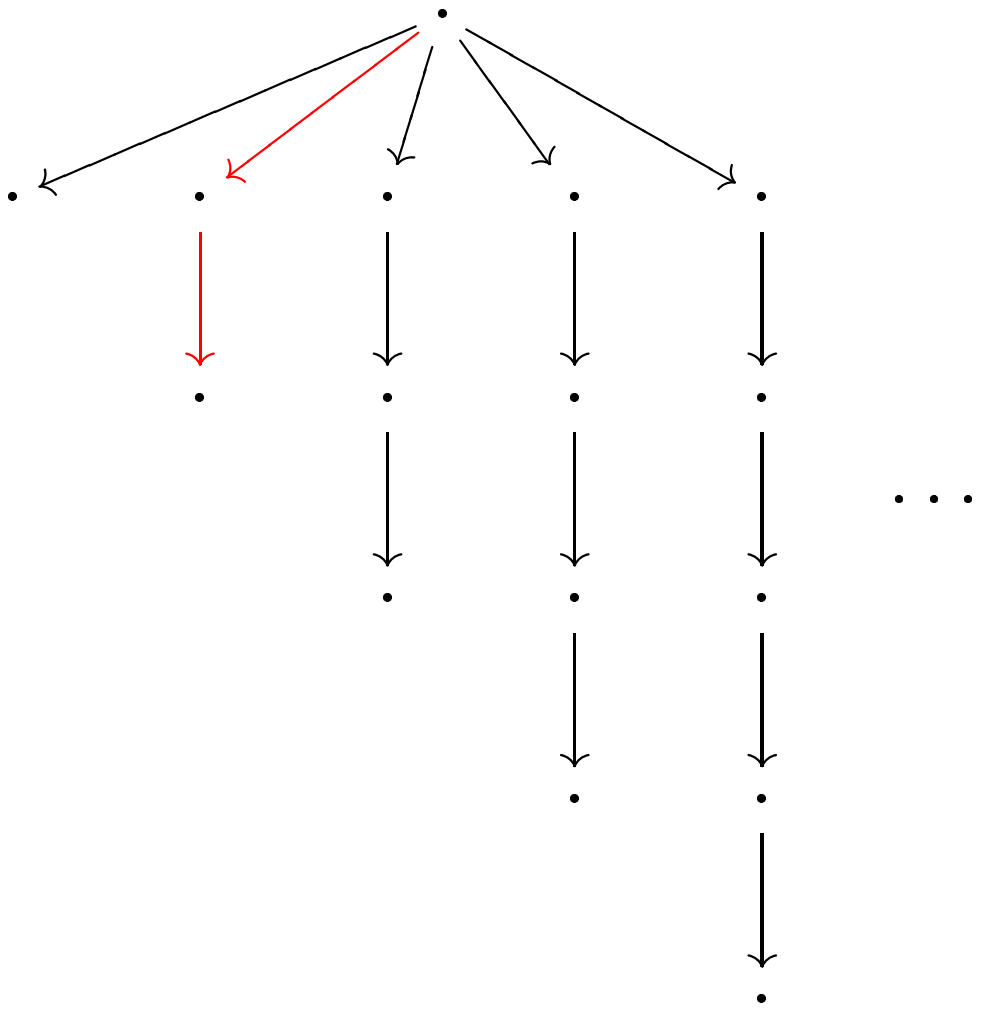
Counterexample

Consider the following non-image finite automaton.



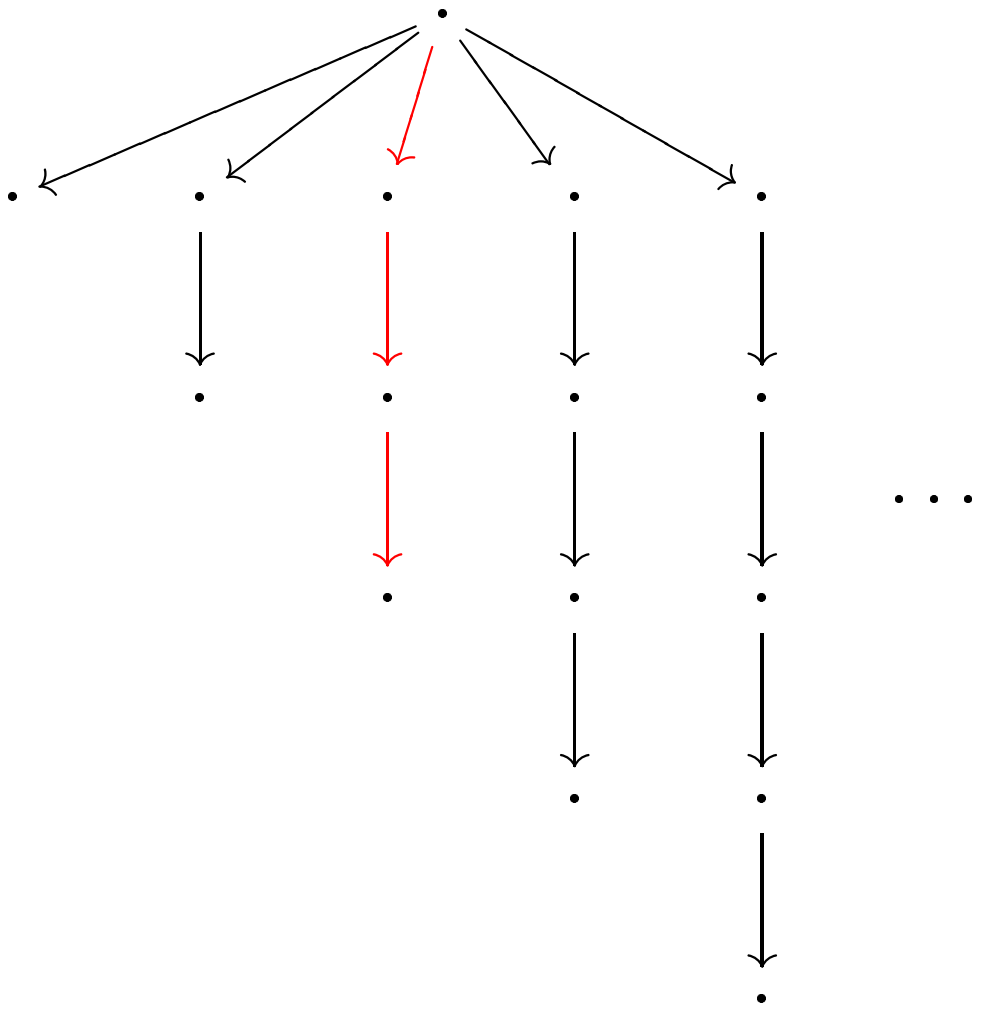
Counterexample

Consider the following non-image finite automaton.



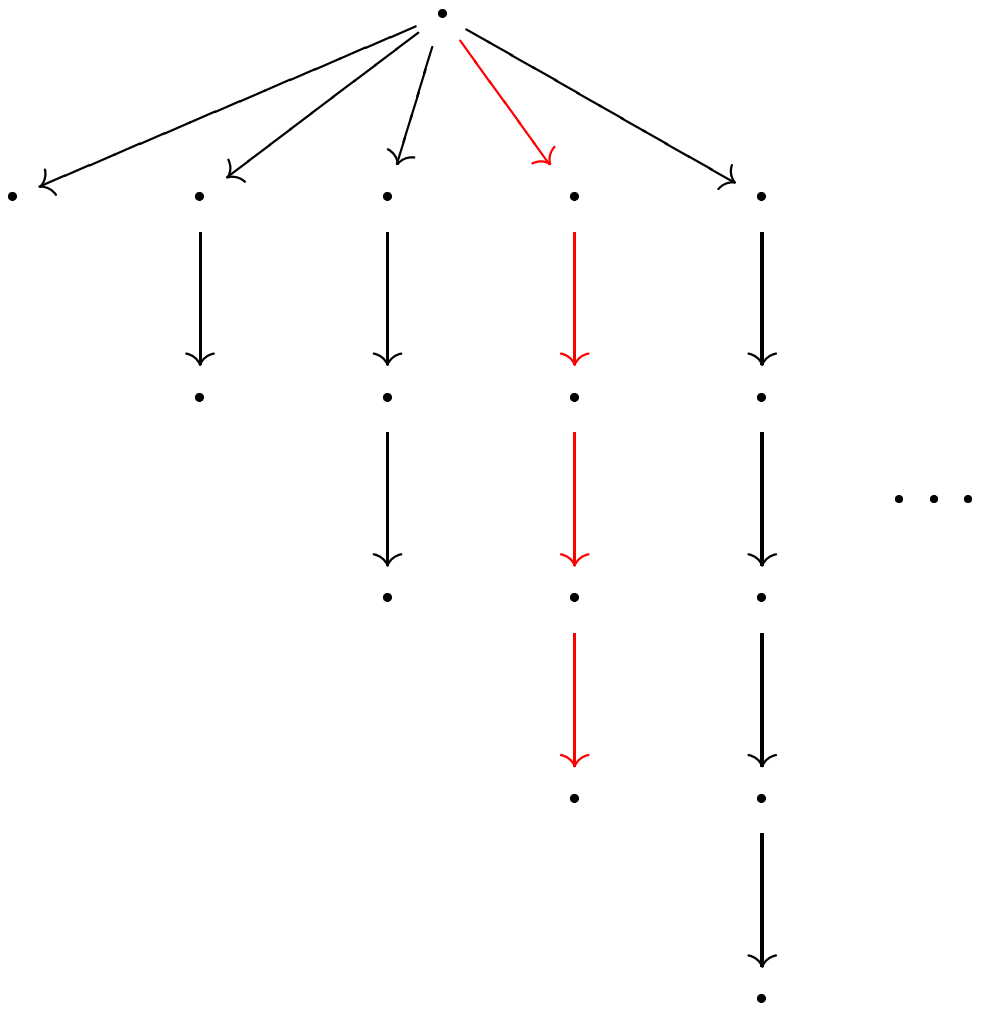
Counterexample

Consider the following non-image finite automaton.



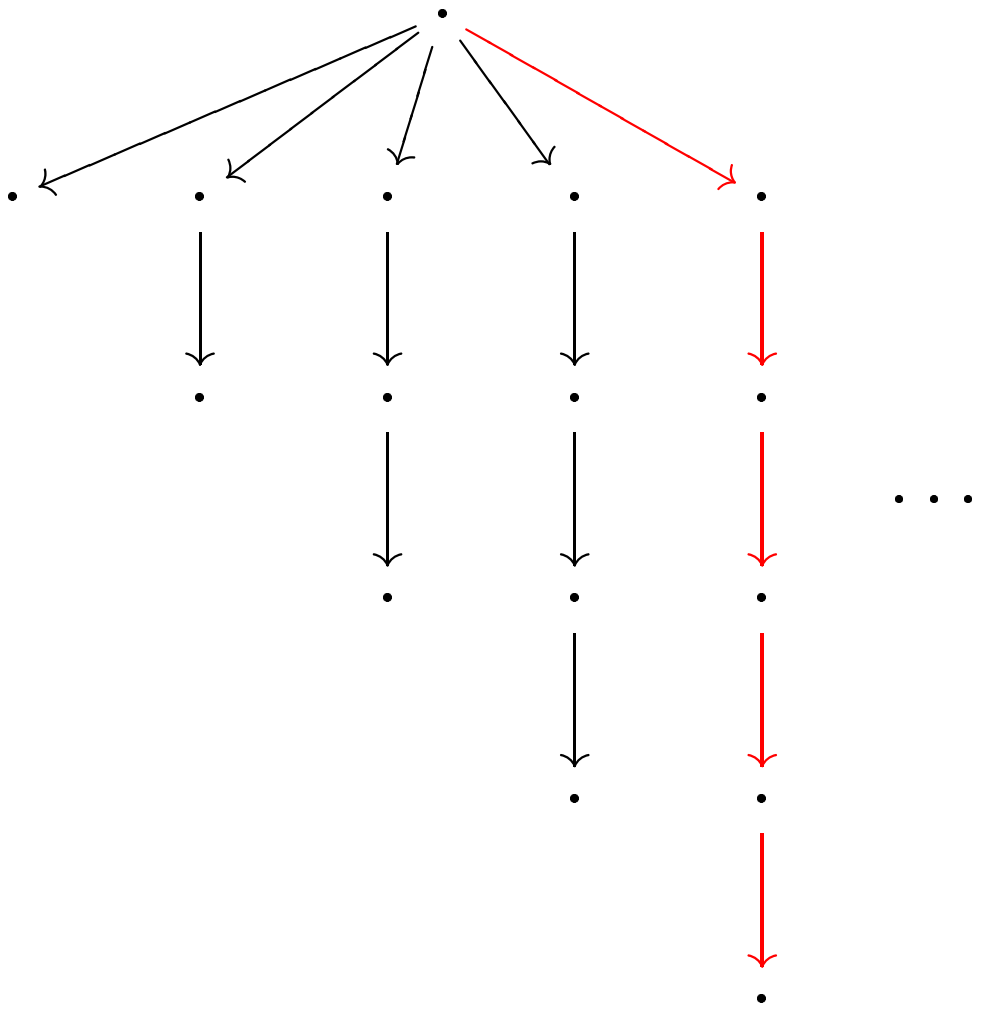
Counterexample

Consider the following non-image finite automaton.



Counterexample

Consider the following non-image finite automaton.



Future Work

- Algebraic CPO with new ordering
- Hidden actions
- More Applications
- Uncountably branching automata
- Characterization of $\mathsf{TrDist}(A)$
- Comparison with other probabilistic models