Decidability and Expressiveness results for Nondeterministic Probabilistic Automata

Alexandra Silva

joint work with

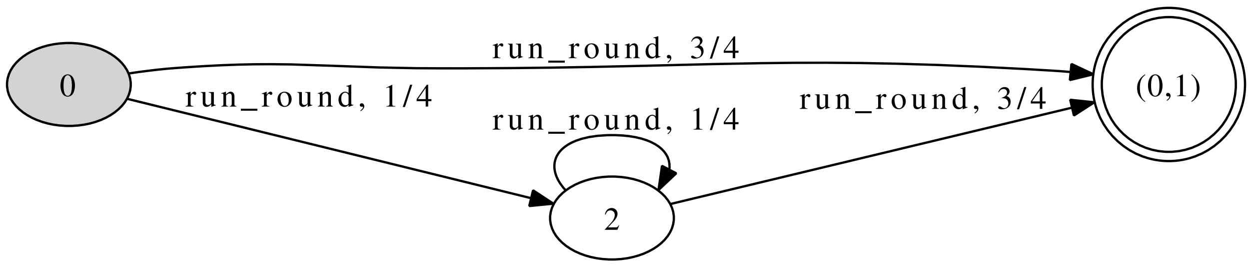
Justin Hsu

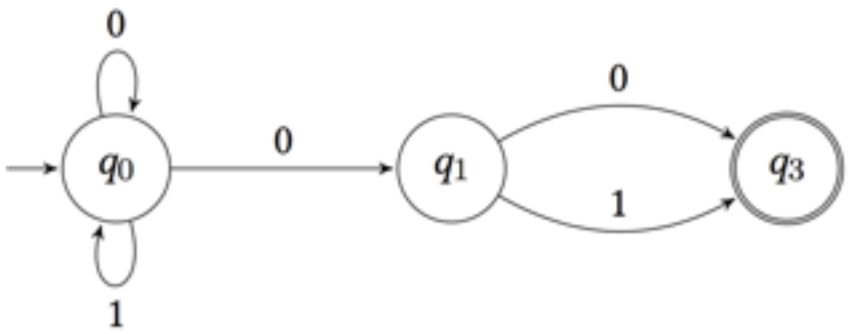
Gerco van Heerdt

Joel Ouaknine

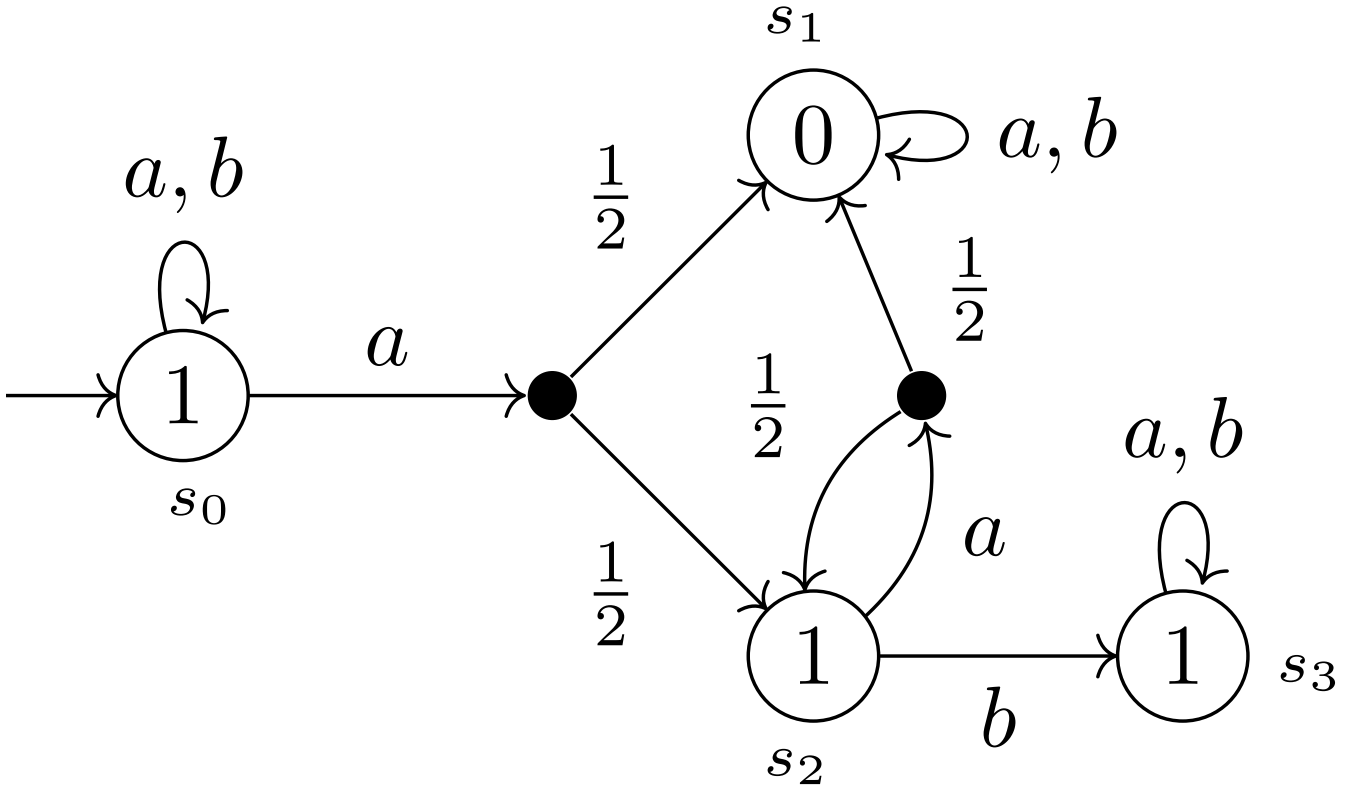
Context

- Probabilistic automata: randomized computation, semantics of programming languages, machine learning.
- Non-deterministic automata: concurrent and distributed systems
- Rabin 80’s: use of nondeterminism and probabilities





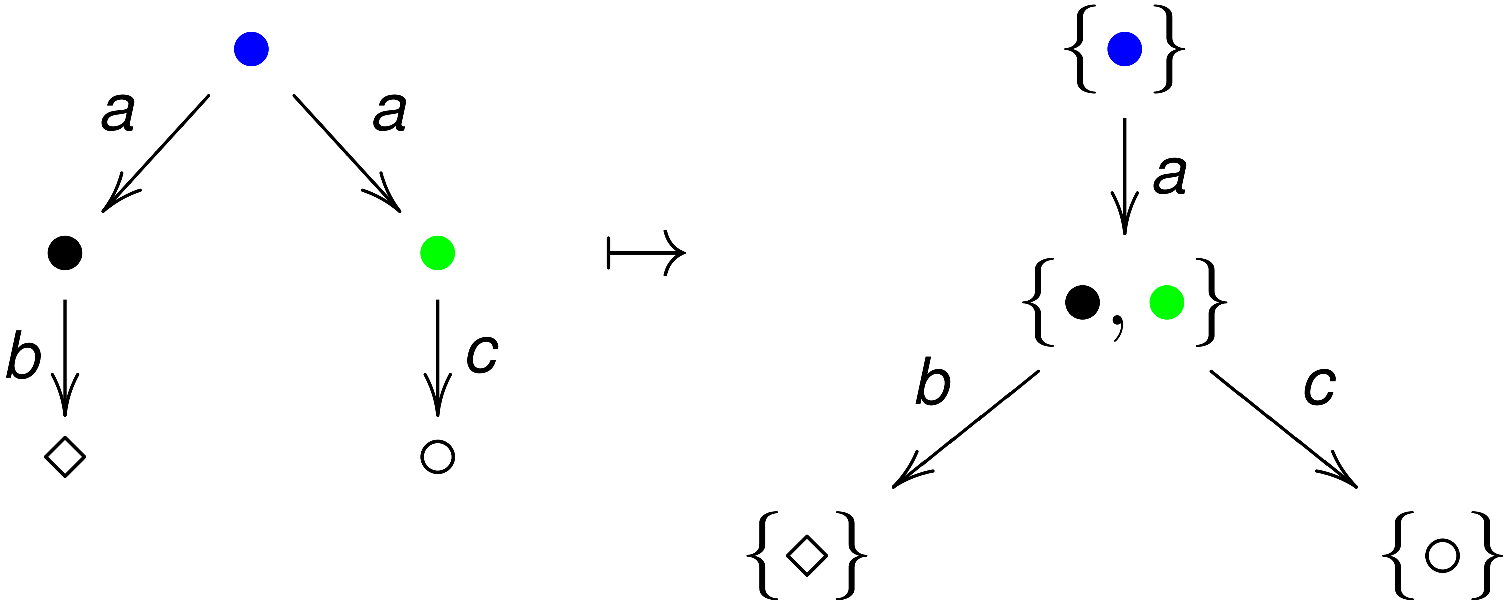
Non-deterministic probabilistic automata



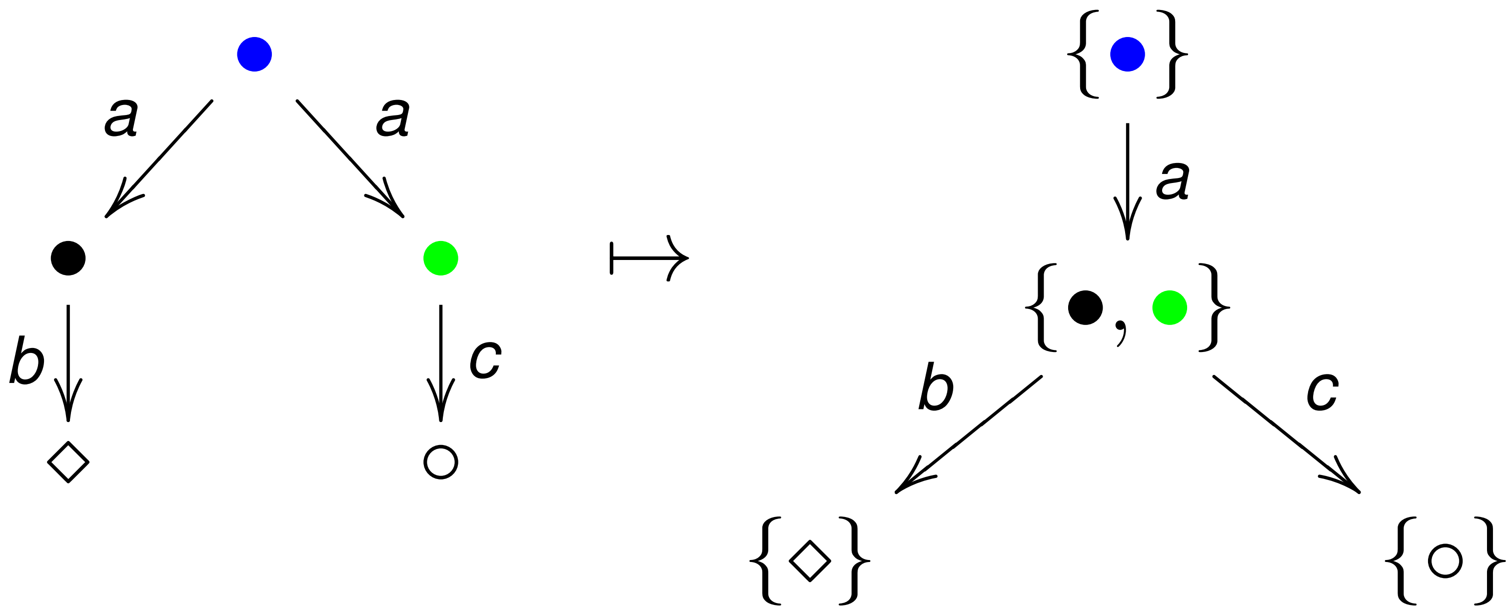
This talk

- Language semantics for NPA
- (Un)decidability
- Expressiveness of non-determinism

Language semantics via powerset construction



Language semantics via powerset construction



$S \to 2 \times \mathcal{P}(S)^A$

$\mathcal{P}(S) \to 2 \times \mathcal{P}(S)^A$

$Q \to 2 \times Q^A$

bisimilarity

language equivalence

Powerset construction

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] & 2 \times \mathcal{P}(S)^A \arrow[d] \\
\mathcal{P}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & 2 \times \mathcal{P}(S)^A
\end{tikzcd}$$

Powerset construction

$$\begin{tikzcd} S \xrightarrow{<o,t>} 2 \times \mathcal{P}(S)^A \arrow[d] \\ \mathcal{P}(S) \xrightarrow{<\bar{o},\bar{t}>} 2 \times \mathcal{P}(S)^A \end{tikzcd}$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

Powerset construction

$$\begin{tikzcd}
S \arrow[r, "<o,t>"] & 2 \times \mathcal{P}(S)^A \arrow[d] \\
\mathcal{P}(S) \arrow[r, "<\bar{o},\bar{t}>"] & 2 \times \mathcal{P}(S)^A
\end{tikzcd}$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

Powerset construction

$$\begin{tikzcd}
S \xrightarrow{<o,t>} 2 \times \mathcal{P}(S)^A \arrow[d] \\
\mathcal{P}(S) \xrightarrow{<\bar{o},\bar{t}>} 2 \times \mathcal{P}(S)^A
\end{tikzcd}$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

2 is a JSL

Powerset construction

$$S \xrightarrow{<o,t>} 2 \times \mathcal{P}(S)^A$$

$$\Big\downarrow$$

$$\mathcal{P}(S) \xrightarrow{<\bar{o},\bar{t}>} 2 \times \mathcal{P}(S)^A$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

2 is a JSL

$$\bar{t}(U)(a) = \bigcup_{U} t(u)(a)$$

Powerset construction

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & 2 \times \mathcal{P}(S)^A \\
\mathcal{P}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & 2 \times \mathcal{P}(S)^A
\end{tikzcd}$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

2 is a JSL

$$\bar{t}(U)(a) = \bigcup_{U} t(u)(a)$$

P(S) is a JSL

Powerset construction

$$\begin{tikzcd} S \arrow[r, "<o{,}t>"] \arrow[d] & 2 \times \mathcal{P}(S)^A \\ \mathcal{P}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & 2 \times \mathcal{P}(S)^A \end{tikzcd}$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

2 is a JSL

$$\bar{t}(U)(a) = \bigcup_{U} t(u)(a)$$

P(S) is a JSL

JSL and P are closely related in a very general way

Powerset construction

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & 2 \times \mathcal{P}(S)^A \\
\mathcal{P}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & 2 \times \mathcal{P}(S)^A
\end{tikzcd}$$

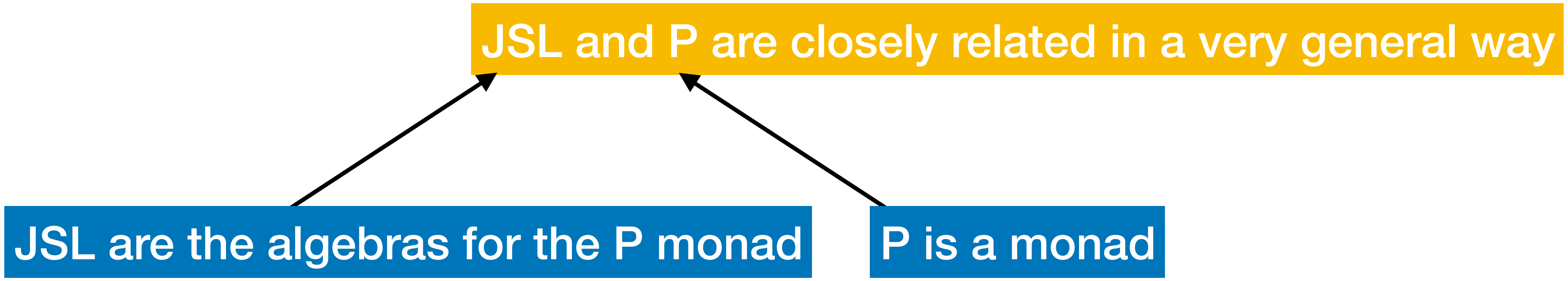
$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

2 is a JSL

$$\bar{t}(U)(a) = \bigcup_{U} t(u)(a)$$

P(S) is a JSL



JSL and P are closely related in a very general way

JSL are the algebras for the P monad

P is a monad

Powerset construction

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & 2 \times \mathcal{P}(S)^A \\
\mathcal{P}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & 2 \times \mathcal{P}(S)^A
\end{tikzcd}$$

$$\bar{o}(U) = 1 \iff \exists u \in U . o(u) = 1$$

$$\bar{o}(U) = \bigvee_{U} o(u)$$

2 is a JSL

$$\bar{t}(U)(a) = \bigcup_{U} t(u)(a)$$

P(S) is a JSL

J

We will used monads and algebras to define a *generalised powerset construction* and have a general *language semantics* for a class of automata

Monads and their algebras

- Monads as *effectful* computation (Moggi 80’s)
- Effects : non-determinism, probabilities, input-output, …

Monads and their algebras

- Monads as *effectful* computation (Moggi 80's)
- Effects : non-determinism, probabilities, input-output, …

$$\mathcal{T} : Set \to Set$$

$$\eta \colon S \to \mathcal{T}S$$

$$\mu \colon \mathcal{T}\mathcal{T}S \to \mathcal{T}S$$

Monads and their algebras

- Monads as *effectful* computation (Moggi 80's)
- Effects : non-determinism, probabilities, input-output, …

$$\mathcal{T} : Set \to Set \qquad \mathcal{P}(S) = \{U \mid U \subseteq S\}$$

$$\eta \colon S \to \mathcal{T}S \qquad \eta(s) = \{s\}$$

$$\mu \colon \mathcal{T}\mathcal{T}S \to \mathcal{T}S \qquad \mu(\Phi) = \bigcup_{U \in \Phi} U$$

Monads and their algebras

- Monads as *effectful* computation (Moggi 80’s)
- Effects : non-determinism, probabilities, input-output, …

$$\mathcal{T} : Set \to Set \qquad \mathcal{P}(S) = \{U \mid U \subseteq S\}$$

$$\eta \colon S \to \mathcal{T}S \qquad \eta(s) = \{s\}$$

$$\mu \colon \mathcal{T}\mathcal{T}S \to \mathcal{T}S \qquad \mu(\Phi) = \bigcup_{U \in \Phi} U$$

μ, η **satisfy some reasonable laws e.g.** $\mu \circ \eta = id$

Monads and their algebras

- Associated with each monad there is a category of (free) algebras

Monads and their algebras

- Associated with each monad there is a category of (free) algebras

Definition *An* algebra for a monad (T, η, μ) *is a pair* (X, h) *consisting of a carrier set* X *and a function* $h\colon TX \to X$ *making the following diagrams commute.*

$$\begin{tikzcd}
X \arrow[r, "\eta"] \arrow[dr, equal] & TX \arrow[d, "h"] \\
 & X
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
TTX \arrow[r, "Th"] \arrow[d, "\mu"'] & TX \arrow[d, "h"] \\
TX \arrow[r, "h"] & X
\end{tikzcd}$$

Monads and their algebras

- Associated with each monad there is a category of (free) algebras

Definition *An* algebra for a monad (T, η, μ) *is a pair* (X, h) *consisting of a carrier set* X *and a function* $h\colon TX \to X$ *making the following diagrams commute.*

$$\begin{tikzcd}
X \arrow[r, "\eta"] \arrow[dr, equal] & TX \arrow[d, "h"] \\
 & X
\end{tikzcd}
\qquad
\begin{tikzcd}
TTX \arrow[r, "Th"] \arrow[d, "\mu"'] & TX \arrow[d, "h"] \\
TX \arrow[r, "h"] & X
\end{tikzcd}$$

$\mathcal{P}$ **join-semilattices (with a bottom element)**

Language semantics for more general automata

$$\begin{tikzcd}
S \arrow[r, "{<}o{,}t{>}"] \arrow[d] & O \times \mathcal{T}(S)^A \\
\mathcal{T}(S) \arrow[r, "{<}o{,}t{>}"] & O \times \mathcal{T}(S)^A
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T} o)(s)$$

Language semantics for more general automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] & O \times \mathcal{T}(S)^A \arrow[d] \\
\mathcal{T}(S) \arrow[r, "<o{,}t>"] & O \times \mathcal{T}(S)^A
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T}o)(s)$$

$$\alpha \colon \mathcal{T}(O) \to O$$

O is a T-algebra

Language semantics for more general automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & O \times \mathcal{T}(S)^A \\
\mathcal{T}(S) \arrow[r, "<o{,}t>"] & O \times \mathcal{T}(S)^A
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T}o)(s)$$

$$\bar{t}(s)(a) = \mu \circ (\mathcal{T}t)(s)(a)$$

$$\alpha \colon \mathcal{T}(O) \to O$$

O is a T-algebra

Language semantics for more general automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & O \times \mathcal{T}(S)^A \\
\mathcal{T}(S) \arrow[r, "<o{,}t>"] & O \times \mathcal{T}(S)^A
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T}o)(s)$$

$$\bar{t}(s)(a) = \mu \circ (\mathcal{T}t)(s)(a)$$

$$\alpha\colon \mathcal{T}(O) \to O$$

O is a T-algebra

$$\mu\colon \mathcal{T}\mathcal{T}(S) \to \mathcal{T}(S)$$

T(S) is a T-algebra (free)

Deterministic Probabilistic automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\
\mathcal{D}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A}
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T} o)(s)$$

$$\alpha = \mathbb{E} \colon \mathcal{D}[0,1] \to [0,1]$$

Deterministic Probabilistic automata

$$\begin{tikzcd} S \arrow[r, "<o,t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\ \mathcal{D}(S) \arrow[r, "<\bar{o},\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A} \end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T} o)(s)$$

$$\alpha = \mathbb{E} \colon \mathcal{D}[0,1] \to [0,1]$$

[0,1] is a convex algebra

Deterministic Probabilistic automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\
\mathcal{D}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A}
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T} o)(s)$$

$$\bar{t}(s)(a) = \mu \circ (\mathcal{T} t)(s)(a)$$

$$\alpha = \mathbb{E} \colon \mathcal{D}[0,1] \to [0,1]$$

[0,1] is a convex algebra

Deterministic Probabilistic automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\
\mathcal{D}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A}
\end{tikzcd}$$

$$\bar{o}(s) = \alpha \circ (\mathcal{T} o)(s)$$

$$\bar{t}(s)(a) = \mu \circ (\mathcal{T} t)(s)(a)$$

$$\alpha = \mathbb{E} \colon \mathcal{D}[0,1] \to [0,1]$$

$$\mu = m \colon \mathcal{D}\mathcal{D}X \to \mathcal{D}X$$

[0,1] is a convex algebra

m - monad multiplication

Deterministic Probabilistic automata

$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\
\mathcal{D}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A}
\end{tikzcd}$$

Belief automaton

$$\bar{o}(s) = \alpha \circ (\mathcal{T} o)(s)$$

$$\bar{t}(s)(a) = \mu \circ (\mathcal{T} t)(s)(a)$$

$$\alpha = \mathbb{E} \colon \mathcal{D}[0,1] \to [0,1]$$

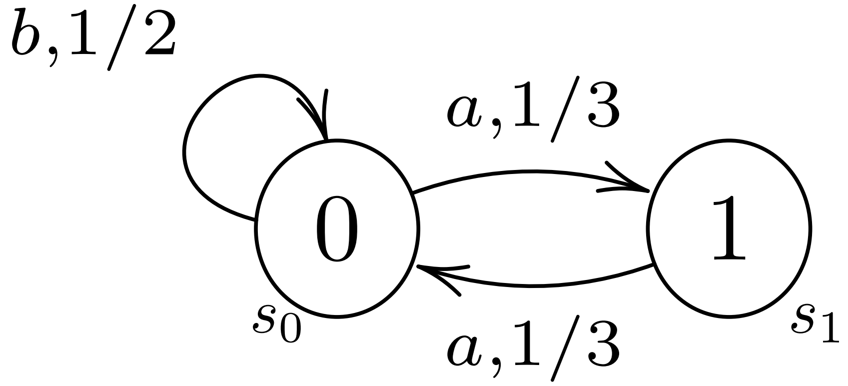
[0,1] is a convex algebra

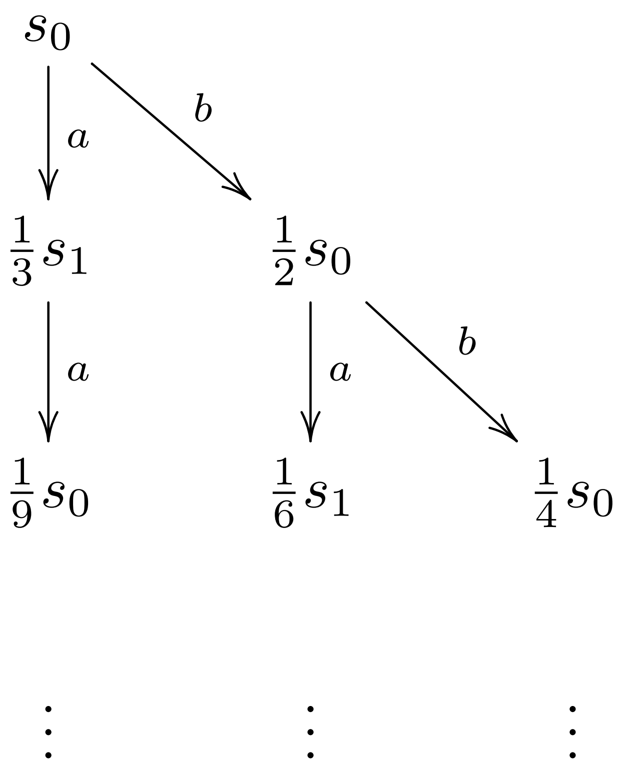
$$\mu = \ m \colon \mathcal{D}\mathcal{D}X \to \mathcal{D}X$$

m - monad multiplication

DPA, example

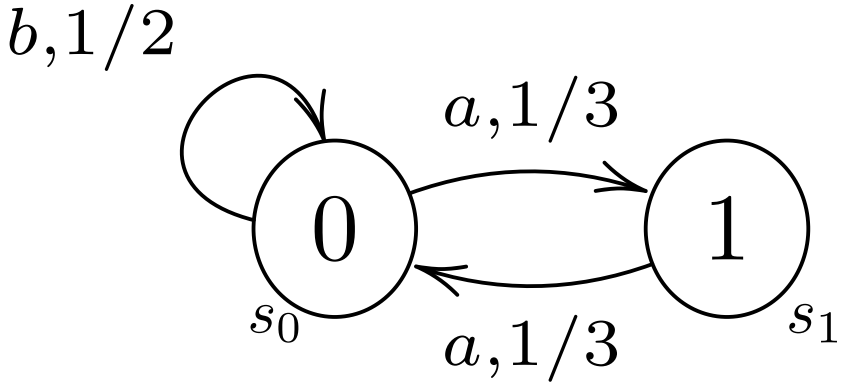
$$\begin{tikzcd}
S \arrow[r, "<o{,}t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\
\mathcal{D}(S) \arrow[r, "<\bar{o}{,}\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A}
\end{tikzcd}$$





DPA, example

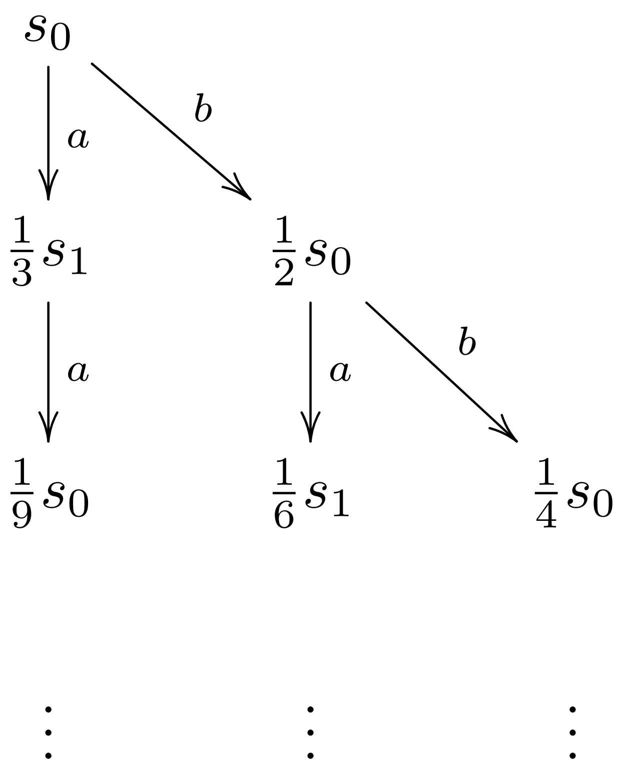
$$\begin{tikzcd}
S \arrow[r, "<o,t>"] \arrow[d] & {[0,1] \times \mathcal{D}(S)^A} \\
\mathcal{D}(S) \arrow[r, "<\bar{o},\bar{t}>"] & {[0,1] \times \mathcal{D}(S)^A}
\end{tikzcd}$$



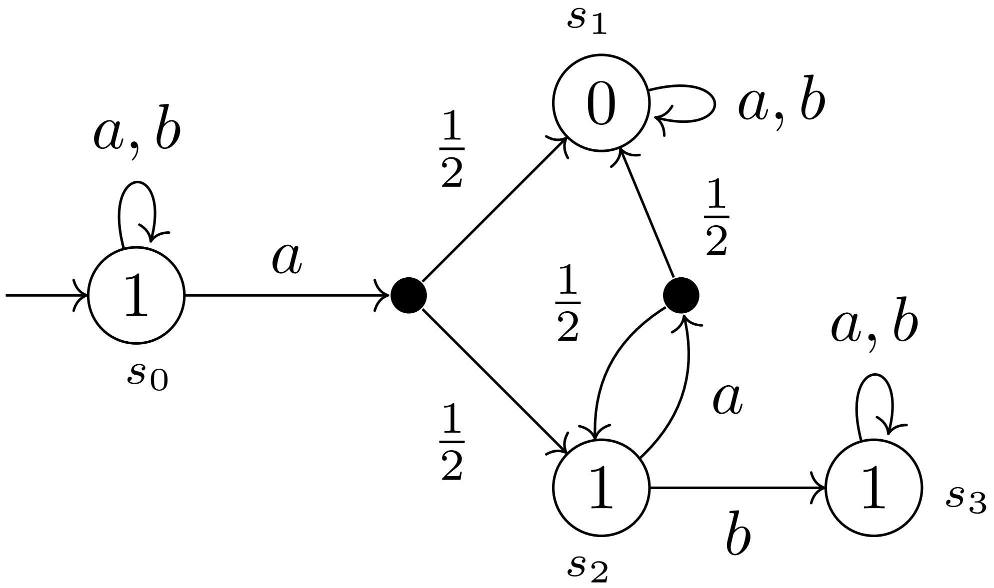
$$\mathcal{L}_s \colon A^* \to [0,1]$$

$$aa \mapsto \frac{1}{9} \times o(s_0) = \frac{1}{9} \times 0 = 0$$

$$ba \mapsto \frac{1}{6} \times o(s_1) = \frac{1}{6} \times 1 = \frac{1}{6}$$

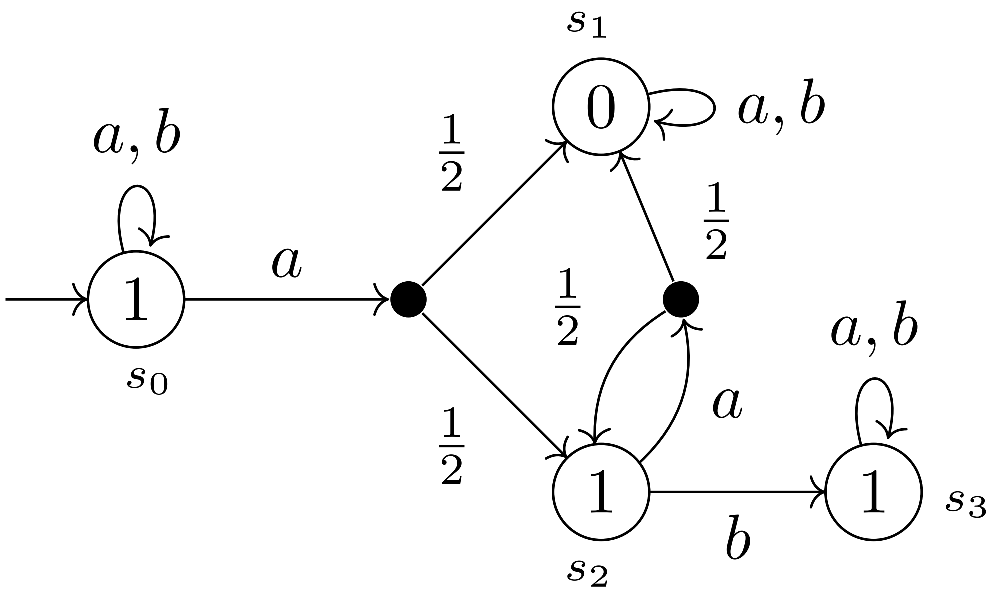


NPA



Definition *A* nondeterministic probabilistic automaton *(NPA) over a (finite) alphabet A is defined by a tuple $(S, s_0, \gamma, \{\tau_a\}_{a \in A})$, where S is a finite set of* states*, $s_0 \in S$ is the* initial state*, $\gamma \colon S \to [0, 1]$ is the* output function*, and $\tau_a \colon S \to \mathcal{P}_c\mathcal{D}S$ are the* transition functions *indexed by inputs $a \in A$.*

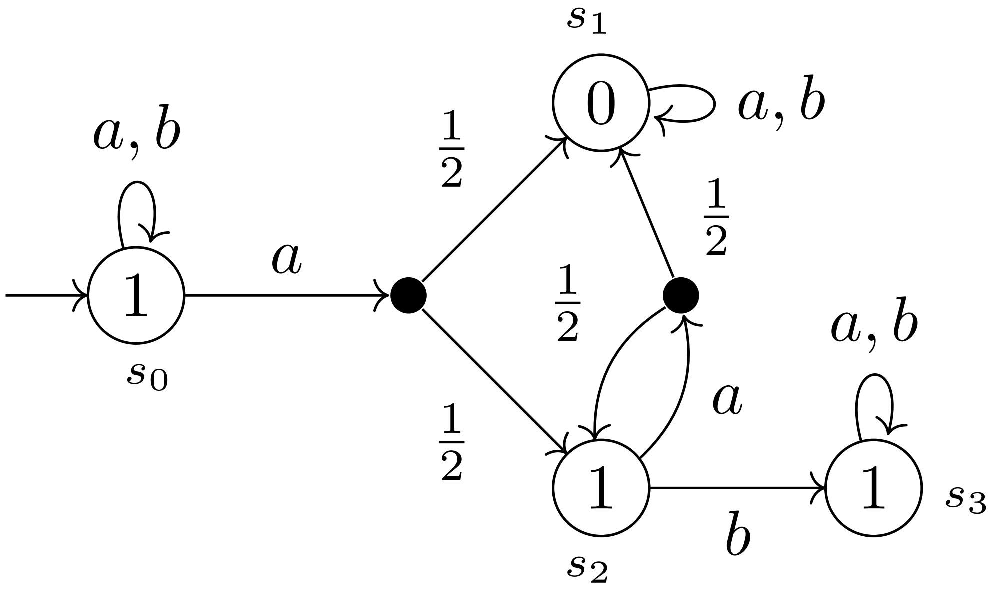
NPA



Definition *A* nondeterministic probabilistic automaton *(NPA) over a (finite) alphabet A is defined by a tuple* $(S, s_0, \gamma, \{\tau_a\}_{a \in A})$, *where S is a finite set of* states, *$s_0 \in S$ is the* initial state, *$\gamma \colon S \to [0,1]$ is the* output function, *and $\tau_a \colon S \to \mathcal{P}_c\mathcal{D}S$ are the* transition functions *indexed by inputs $a \in A$.*

$$S \to 2 \times \mathcal{P}(S)^A$$

NPA

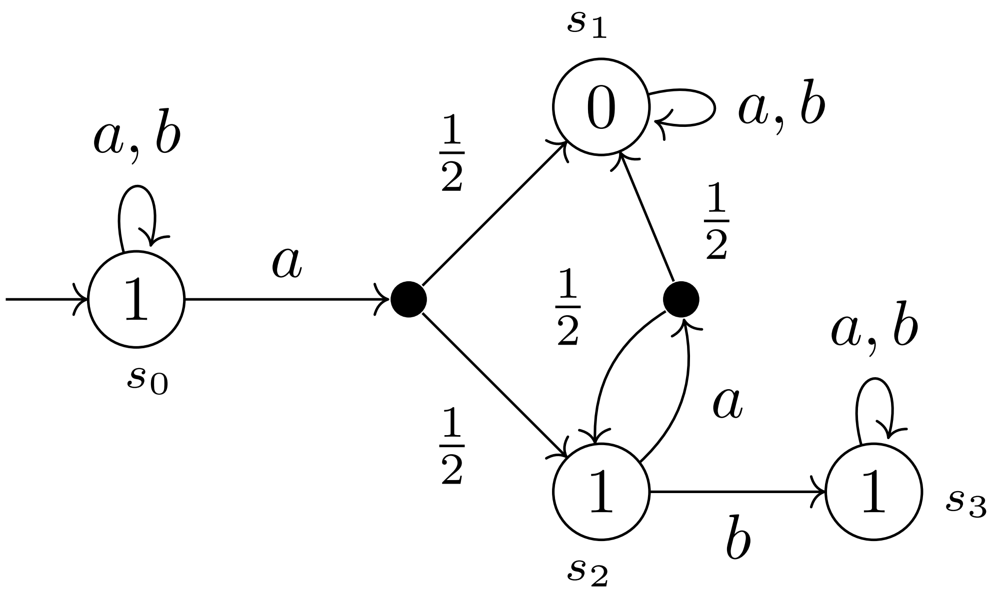


Definition *A* nondeterministic probabilistic automaton *(NPA) over a (finite) alphabet A is defined by a tuple* $(S, s_0, \gamma, \{\tau_a\}_{a \in A})$, *where S is a finite set of* states, *$s_0 \in S$ is the* initial state, *$\gamma \colon S \to [0, 1]$ is the* output function, *and $\tau_a \colon S \to \mathcal{P}_c\mathcal{D}S$ are the* transition functions *indexed by inputs $a \in A$.*

$$S \to 2 \times \mathcal{P}(S)^A$$

$$S \to [0, 1] \times \mathcal{P}_c\mathcal{D}(S)^A$$

NPA



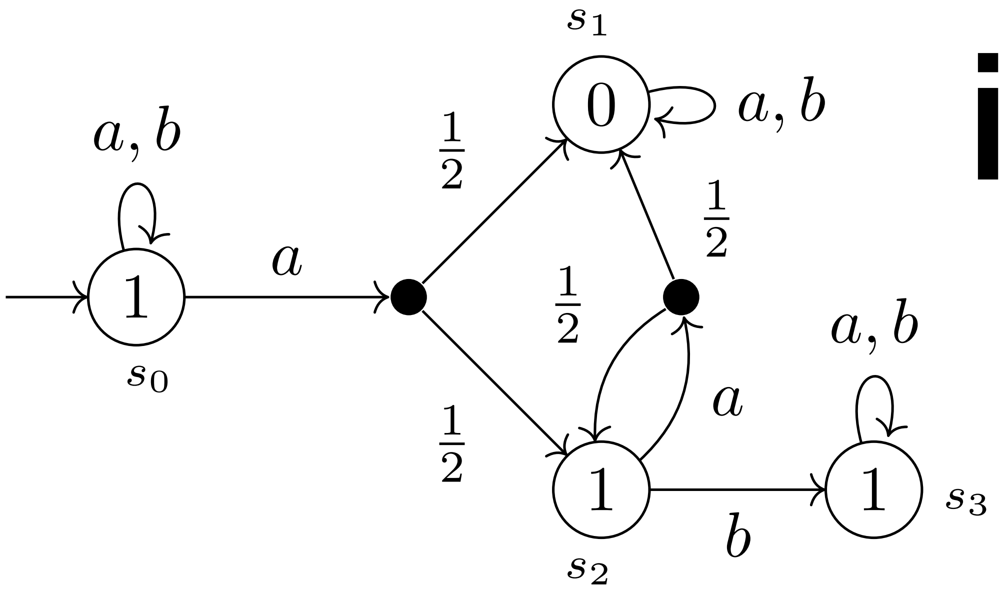
Definition *A* nondeterministic probabilistic automaton *(NPA) over a (finite) alphabet A is defined by a tuple* $(S, s_0, \gamma, \{\tau_a\}_{a \in A})$*, where S is a finite set of* states, $s_0 \in S$ *is the* initial state, $\gamma \colon S \to [0,1]$ *is the* output function, *and* $\tau_a \colon S \to \mathcal{P}_c\mathcal{D}S$ *are the* transition functions *indexed by inputs $a \in A$.*

$$S \to 2 \times \mathcal{P}(S)^A$$

$$S \to [0,1] \times \mathcal{P}_c\mathcal{D}(S)^A$$

algebra **monad**

Language semantics intuitively

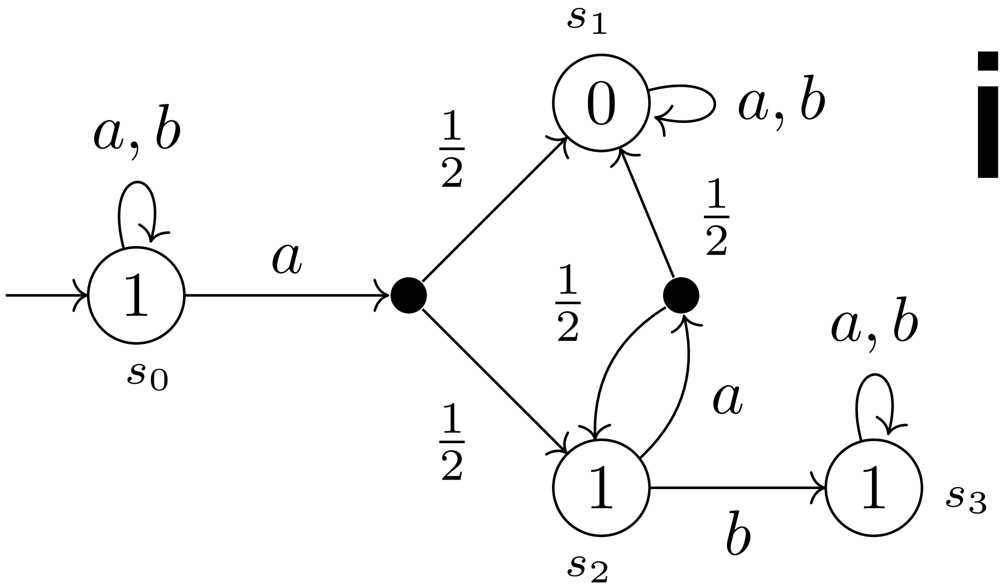


$$\begin{tikzcd}
s_0 \arrow[d, "a"] \\
s_0 \arrow[d, "a"] \\
s_0 \arrow[d, squiggly, no head] \\
1
\end{tikzcd}$$

$$\begin{tikzcd}
s_0 \arrow[d, "a"] \\
s_0 \arrow[d, "a"] \\
\frac{1}{2}s_1 + \frac{1}{2}s_2 \arrow[d, squiggly, no head] \\
\frac{1}{2}
\end{tikzcd}$$

$$\begin{tikzcd}
s_0 \arrow[d, "a"] \\
\frac{1}{2}s_1 + \frac{1}{2}s_2 \arrow[d, "a"] \\
\frac{1}{2}\left(\frac{1}{2}s_1 + \frac{1}{2}s_2\right) + \frac{1}{2}s_1 \\
= \frac{3}{4}s_1 + \frac{1}{4}s_2 \arrow[d, squiggly, no head] \\
\frac{1}{4}
\end{tikzcd}$$

Language semantics intuitively



$$\begin{tikzcd}
s_0 \arrow[d, "a"] \\
s_0 \arrow[d, "a"] \\
s_0 \arrow[d, squiggly] \\
1
\end{tikzcd}$$

$$\begin{tikzcd}
s_0 \arrow[d, "a"] \\
s_0 \arrow[d, "a"] \\
\frac{1}{2}s_1 + \frac{1}{2}s_2 \arrow[d, squiggly] \\
\frac{1}{2}
\end{tikzcd}$$

$$\begin{tikzcd}
s_0 \arrow[d, "a"] \\
\frac{1}{2}s_1 + \frac{1}{2}s_2 \arrow[d, "a"] \\
\begin{array}{l} \frac{1}{2}\left(\frac{1}{2}s_1 + \frac{1}{2}s_2\right) + \frac{1}{2}s_1 \\ = \frac{3}{4}s_1 + \frac{1}{4}s_2 \end{array} \arrow[d, squiggly] \\
\frac{1}{4}
\end{tikzcd}$$

max
min
?

Language semantics formally

$$S \to \underset{\underset{\textbf{algebra}}{\uparrow}}{[0,1]} \times \underset{\underset{\textbf{monad}}{\uparrow}}{\mathcal{P}_c\mathcal{D}}(S)^A$$

$\mathcal{P}_c\mathcal{D}$ **is a composite monad**

Language semantics formally

$$S \to \underset{\substack{\big\uparrow \\ \textbf{algebra}}}{[0,1]} \times \underset{\substack{\big\uparrow \\ \textbf{monad}}}{\mathcal{P}_c\mathcal{D}}(S)^A$$

$\mathcal{P}_c\mathcal{D}$ **is a composite monad**

The semantics should be conservative set DPA

Language semantics formally

$$S \to \underset{\substack{\uparrow \\ \text{algebra}}}{[0,1]} \times \underset{\substack{\uparrow \\ \text{monad}}}{\mathcal{P}_c\mathcal{D}}(S)^A$$

$\mathcal{P}_c\mathcal{D}$ **is a composite monad**

The semantics should be conservative set DPA

$$\begin{tikzcd}
\mathcal{D}[0,1] \arrow[d, "\{-\}"'] \arrow[dr, "\mathbb{E}"] & \\
\mathcal{P}_c\mathcal{D}[0,1] \arrow[r, "o"] & {[0,1]}
\end{tikzcd}$$

Language semantics formally

$$\begin{tikzcd}
\mathcal{D}[0,1] \arrow[d, "\{-\}"'] \arrow[dr, "\mathbb{E}"] & \\
\mathcal{P}_c\mathcal{D}[0,1] \arrow[r, "o"] & {[0,1]}
\end{tikzcd}$$

Proposition 1. *Any* $\mathcal{P}_c\mathcal{D}$*-algebra on* $[0,1]$ *extending* $\mathbb{E}\colon \mathcal{D}[0,1] \to [0,1]$ *is of the form* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\alpha} [0,1]$, *where* α *is a* $\mathcal{P}_c$*-algebra.*

Proposition 2. *The only* $\mathcal{P}_c$*-algebras on the convex set* $[0,1]$ *are* min *and* max.

Corollary 1. *The only* $\mathcal{P}_c\mathcal{D}$*-algebras on* $[0,1]$ *extending* $\mathbb{E}$ *are* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\mathsf{min}} [0,1]$ *and* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\mathsf{max}} [0,1]$.

Language semantics formally

Proof:

Basic convex algebra facts

$\mathsf{Conv}(\{\{0\},\{1\},[0,1]\}) \;=\; \mathcal{P}_c[0,1]$

$\mathcal{P}_c[0,1] \xrightarrow{\;\alpha\;} [0,1]$

α is completely determined by $\alpha([0,1])$

$\alpha([0,1]) = 0$ or $\alpha([0,1]) = 1$

min or max

$$\begin{tikzcd}
\mathcal{D}[0,1] \arrow[d, "\{-\}"'] \arrow[dr, "\mathbb{E}"] & \\
\mathcal{P}_c\mathcal{D}[0,1] \arrow[r, "o"] & {[0,1]}
\end{tikzcd}$$

on $[0,1]$ *extending* $\mathbb{E}\colon \mathcal{D}[0,1] \to [0,1]$ *is of the form* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\alpha} [0,1]$, *where* α *is a* $\mathcal{P}_c$*-algebra.*

Proposition 2. *The only* $\mathcal{P}_c$*-algebras on the convex set* $[0,1]$ *are* min *and* max.

Corollary 1. *The only* $\mathcal{P}_c\mathcal{D}$*-algebras on* $[0,1]$ *extending* $\mathbb{E}$ *are* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\mathsf{min}} [0,1]$ *and* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\mathsf{max}} [0,1]$.

Language semantics formally

Proof:

Basic convex algebra facts

$$\mathsf{Conv}(\{\{0\},\{1\},[0,1]\}) = \mathcal{P}_c[0,1]$$

$$\mathcal{P}_c[0,1] \xrightarrow{\alpha} [0,1]$$

α is completely determined by $\alpha([0,1])$

$\alpha([0,1]) = 0$ or $\alpha([0,1]) = 1$

min **or** **max**

$$\begin{tikzcd}
\mathcal{D}[0,1] \arrow[d, "\{-\}"'] \arrow[dr, "\mathbb{E}"] & \\
\mathcal{P}_c\mathcal{D}[0,1] \arrow[r, "o"] & {[0,1]}
\end{tikzcd}$$

This formally explains why in the literature min and max are the only functions used

on $[0,1]$ *extendi*

form $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\alpha} [0,1]$, *where* α *is a*

Proposition 2. *The only* $\mathcal{P}_c$*-algebras on the convex set* $[0,1]$ *are* min *and* max.

Corollary 1. *The only* $\mathcal{P}_c\mathcal{D}$*-algebras on* $[0,1]$ *extending* $\mathbb{E}$ *are* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\mathsf{min}} [0,1]$ *and* $\mathcal{P}_c\mathcal{D}[0,1] \xrightarrow{\mathcal{P}_c\mathbb{E}} \mathcal{P}_c[0,1] \xrightarrow{\mathsf{max}} [0,1]$.

Summary, so far

$$\begin{tikzcd} S \to [0,1] \times \mathcal{P}_c\mathcal{D}(S)^A \arrow[d, squiggly, "\text{generalised powerset construction}"] \\ \mathcal{P}_c\mathcal{D}(S) \to [0,1] \times \mathcal{P}_c\mathcal{D}(S)^A \end{tikzcd}$$

min and max are the only algebras

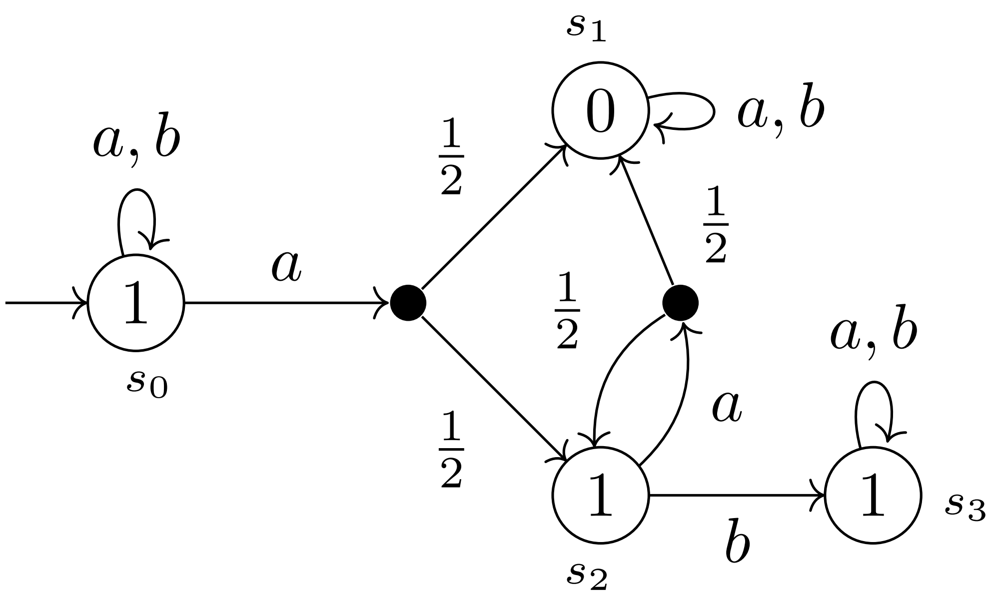
Deterministic automaton

Language semantics as usual using min/max to combine set of output results

Next

- Non-determinism brings a lot of trouble… is it necessary?
- Given two NPAs can we decide whether they are equivalent?

Is non-determinism really important?



$\mathcal{L}_a \colon \{a, b\}^* \to [0, 1]$ by $\mathcal{L}_a(u) = 2^{-n}$

n - length longest sequence of a's in u

Theorem 1. *NPAs are more expressive than DPAs. Specifically, there is no DPA, or even WFA, accepting* $\mathcal{L}_a$.

Is non-determinism really important?

Theorem *There is a language over a unary alphabet that is recognizable by an NPA but not by any WFA (and in particular any DPA).*

Is non-determinism really important?

Theorem *There is a language over a unary alphabet that is recognizable by an NPA but not by any WFA (and in particular any DPA).*

Proof uses results of Linear recurrence sequences

Skolem-Macher-Lech Theorem

Cayley-Hamilton Theorem

(Un)Decidability

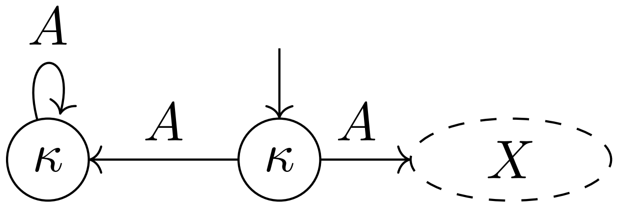
Theorem *Equivalence of NPAs is undecidable when* $|A| \geq 2$ *and the* $\mathcal{P}_c\mathcal{D}$-*algebra on* $[0,1]$ *extends the usual* $\mathcal{D}$-*algebra on* $[0,1]$.

(Un)Decidability

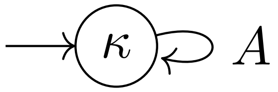
Theorem *Equivalence of NPAs is undecidable when* $|A| \geq 2$ *and the* $\mathcal{P}_c\mathcal{D}$*-algebra on* $[0, 1]$ *extends the usual* $\mathcal{D}$*-algebra on* $[0, 1]$.

Proof. Let X be a DPA and $\kappa \in [0, 1]$. We define NPAs Y and Z as follows:

$Y =$



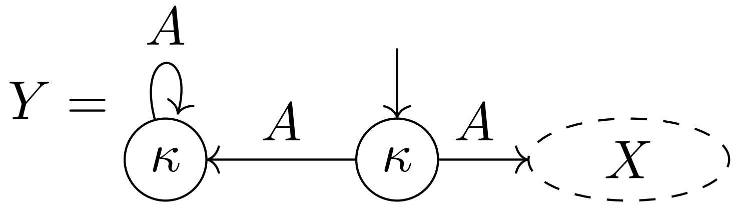
$Z =$

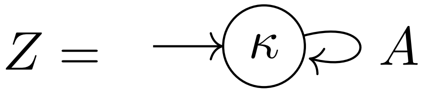


(Un)Decidability

Theorem *Equivalence of NPAs is undecidable when* $|A| \geq 2$ *and the* $\mathcal{P}_c\mathcal{D}$*-algebra on* $[0,1]$ *extends the usual* $\mathcal{D}$*-algebra on* $[0,1]$.

Proof. Let X be a DPA and $\kappa \in [0,1]$. We define NPAs Y and Z as follows:





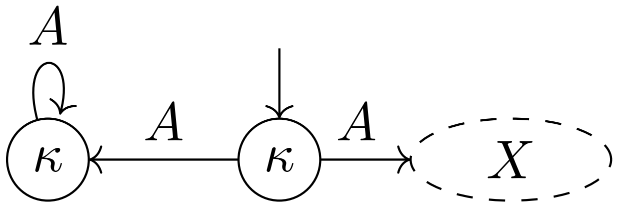
$$\mathcal{L}_Y(\varepsilon) = \kappa = \mathcal{L}_Z(\varepsilon)$$

(Un)Decidability

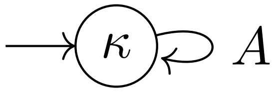
Theorem *Equivalence of NPAs is undecidable when $|A| \geq 2$ and the $\mathcal{P}_c\mathcal{D}$-algebra on $[0,1]$ extends the usual $\mathcal{D}$-algebra on $[0,1]$.*

Proof. Let X be a DPA and $\kappa \in [0,1]$. We define NPAs Y and Z as follows:

$Y =$



$Z =$



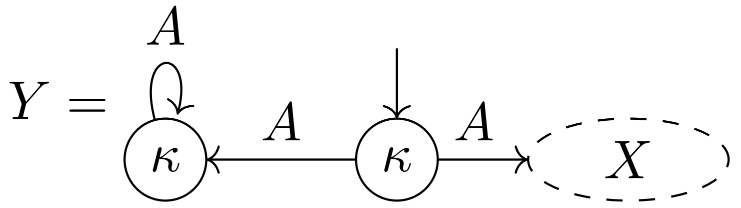
$$\mathcal{L}_Y(\varepsilon) = \kappa = \mathcal{L}_Z(\varepsilon)$$

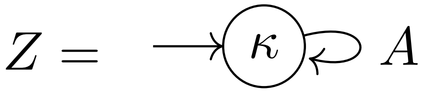
$$\mathcal{L}_Y(av) = (\alpha \circ \mathsf{Conv})(\{\kappa, \mathcal{L}_X(v)\}) \qquad\qquad \mathcal{L}_Z(av) = \kappa.$$

(Un)Decidability

Theorem *Equivalence of NPAs is undecidable when* $|A| \geq 2$ *and the* $\mathcal{P}_c\mathcal{D}$*-algebra on* $[0,1]$ *extends the usual* $\mathcal{D}$*-algebra on* $[0,1]$.

Proof. Let X be a DPA and $\kappa \in [0,1]$. We define NPAs Y and Z as follows:





$$\mathcal{L}_Y(\varepsilon) = \kappa = \mathcal{L}_Z(\varepsilon)$$

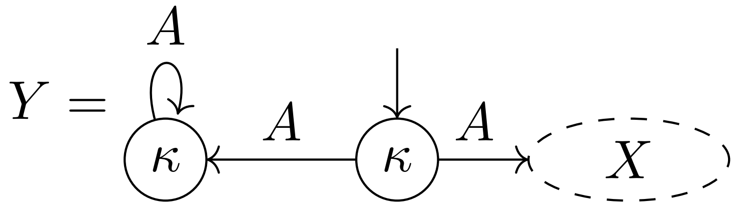
$$\mathcal{L}_Y(av) = (\alpha \circ \mathsf{Conv})(\{\kappa, \mathcal{L}_X(v)\}) \qquad\qquad \mathcal{L}_Z(av) = \kappa.$$

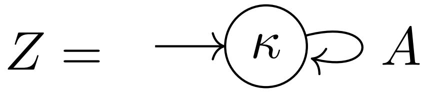
if $\alpha = \mathsf{min}$, then $\mathcal{L}_Y = \mathcal{L}_Z$ if and only if $\mathcal{L}_X(v) \geq \kappa$ for all $v \in A^*$

(Un)Decidability

Theorem *Equivalence of NPAs is undecidable when* $|A| \geq 2$ *and the* $\mathcal{P}_c\mathcal{D}$*-algebra on* $[0,1]$ *extends the usual* $\mathcal{D}$*-algebra on* $[0,1]$.

Proof. Let X be a DPA and $\kappa \in [0,1]$. We define NPAs Y and Z as follows:





$$\mathcal{L}_Y(\varepsilon) = \kappa = \mathcal{L}_Z(\varepsilon)$$

$$\mathcal{L}_Y(av) = (\alpha \circ \mathsf{Conv})(\{\kappa, \mathcal{L}_X(v)\}) \qquad \mathcal{L}_Z(av) = \kappa.$$

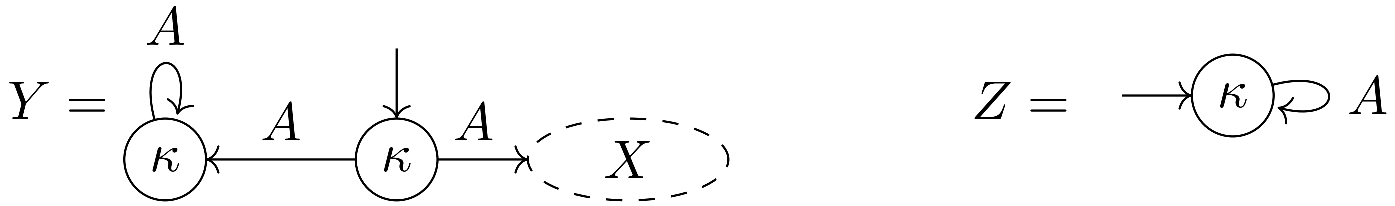
if $\alpha = \mathsf{min}$, then $\mathcal{L}_Y = \mathcal{L}_Z$ if and only if $\mathcal{L}_X(v) \geq \kappa$ for all $v \in A^*$

$\alpha = \mathsf{max}$ $\qquad \mathcal{L}_X(v) \leq \kappa$

(Un)Decidability

Theorem *Equivalence of NPAs is undecidable when* $|A| \geq 2$ *and the* $\mathcal{P}_c\mathcal{D}$-*algebra on* $[0,1]$ *extends the usual* $\mathcal{D}$-*algebra on* $[0,1]$.

Proof. Let X be a DPA and $\kappa \in [0,1]$. We define NPAs Y and Z as follows:



$$\mathcal{L}_Y(\varepsilon) = \kappa = \mathcal{L}_Z(\varepsilon)$$

$$\mathcal{L}_Y(av) = (\alpha \circ \mathsf{Conv})(\{\kappa, \mathcal{L}_X(v)\}$$

Threshold problems undecidable for alphabets of size at least 2 [Blondel, Canterini'03]

if $\alpha = \mathsf{min}$, then $\mathcal{L}_Y = \mathcal{L}_Z$ if and only if $\mathcal{L}_X(v) \geq \kappa$ for all $v \in A^*$

$\alpha = \mathsf{max}$ $\mathcal{L}_X(v) \leq \kappa$

(Un)Decidability

- (Un)decidability of threshold problem not know for |A|=1
- Reduction to the *Positivity problem* for LRS
- Decidability of *Positivity* open for >80 years
- Decision procedure would entail breakthroughs in open problems in number theory (algorithm to compute the homogeneous Diophantine approximation type for a class of transcendental numbers)

Corollary 2. *The Positivity problem for linear recurrence sequences can be reduced to the equivalence problem of NPAs over a unary alphabet.*

Approximation

Approximation

- Deciding exact equivalence of NPAs is computationally intractable (or at least very difficult, for a unary alphabet)

Approximation

- Deciding exact equivalence of NPAs is computationally intractable (or at least very difficult, for a unary alphabet)
- Approximate a *discounted metric* on weighted languages

Approximation

- Deciding exact equivalence of NPAs is computationally intractable (or at least very difficult, for a unary alphabet)
- Approximate a *discounted metric* on weighted languages
- Longer words contribute less than shorter words:

Given $c \in [0,1)$ and two weighted languages $l_1, l_2 \colon A^* \to [0,1]$, we define

$$d_c(l_1, l_2) = \sum_{u \in A^*} |l_1(u) - l_2(u)| \cdot \left(\frac{c}{|A|}\right)^{|u|}.$$

Approximation

- Deciding exact equivalence of NPAs is computationally intractable (or at least very difficult, for a unary alphabet)

- Approximate a *discounted metric* on weighted languages

- Longer words contribute less than shorter words:

Given $c \in [0,1)$ and two weighted languages $l_1, l_2 \colon A^* \to [0,1]$, we define

$$d_c(l_1, l_2) = \sum_{u \in A^*} |l_1(u) - l_2(u)| \cdot \left(\frac{c}{|A|}\right)^{|u|}.$$

- Approximate the metric for any desired non-zero error:

Theorem *There is a procedure that given $c \in [0,1)$, $\kappa > 0$, and computable functions $l_1, l_2 \colon A^* \to [0,1]$ outputs $x \in \mathbb{R}_+$ such that $|d_c(l_1, l_2) - x| \leq \kappa$.*

Conclusions

- Non-determinism + probabilities subtle from a semantics and algorithmic perspective
- Connections between and interplay of convex algebra, number theory, category theory
- Approximation techniques interesting for verification applications - different metrics?

Questions?