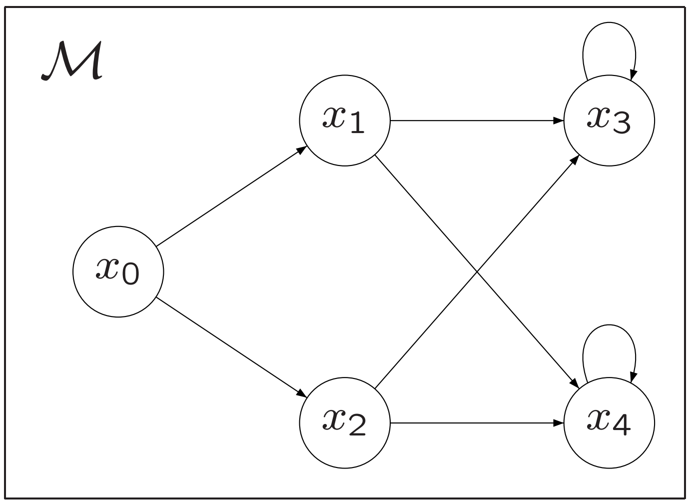
Equivalence of Labeled Markov Chains

Laurent Doyen
École Polytechnique Fédérale de Lausanne

Joint work with
Tom Henzinger, Jean-François Raskin

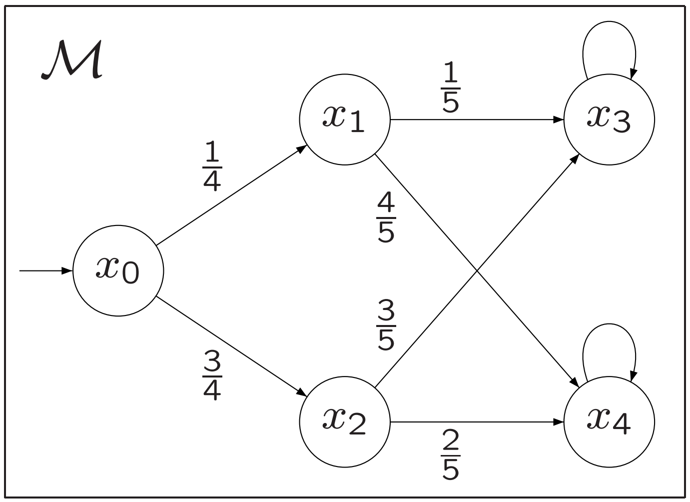
RWTH Aachen, June 2008

Labeled Markov Chains



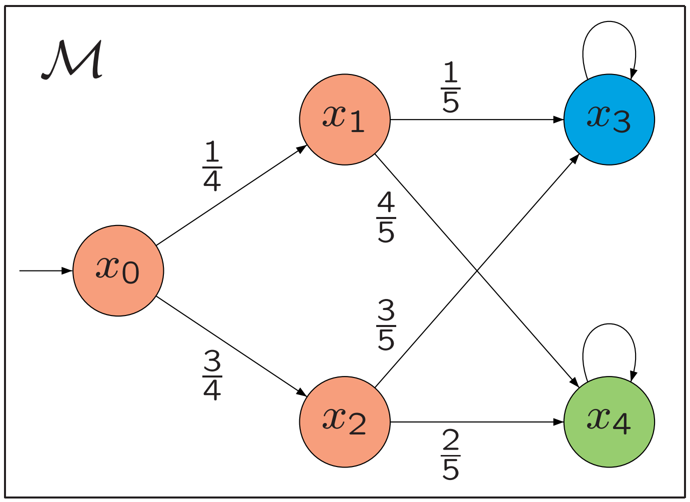
State space $\{x_0, \ldots, x_4\}$

Labeled Markov Chains



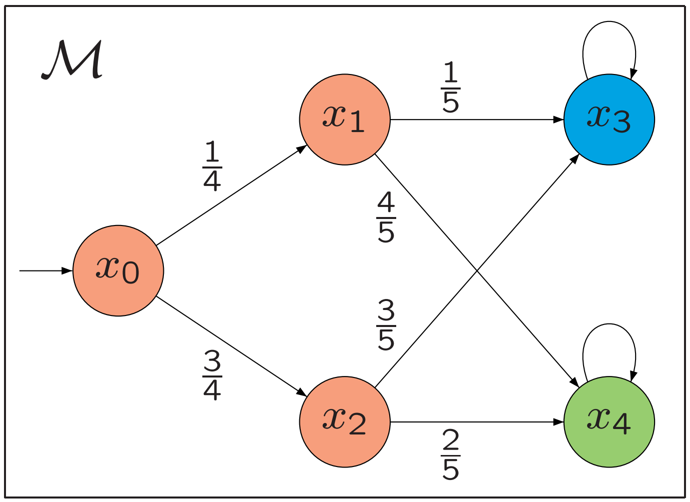
State space	$\{x_0, \ldots, x_4\}$
Initial dist.	$(1, 0, 0, 0, 0)$
Prob. trans.	$\delta(x_0, x_1) = \frac{1}{4}$, etc.

Labeled Markov Chains



State space	$\{x_0, \ldots, x_4\}$
Initial dist.	$(1, 0, 0, 0, 0)$
Prob. trans.	$\delta(x_0, x_1) = \frac{1}{4}$, etc.
Observations	$\Sigma = \{\bullet, \bullet, \bullet\}$
Labeling	$\mathcal{L}(x_0) = \bullet$, etc.

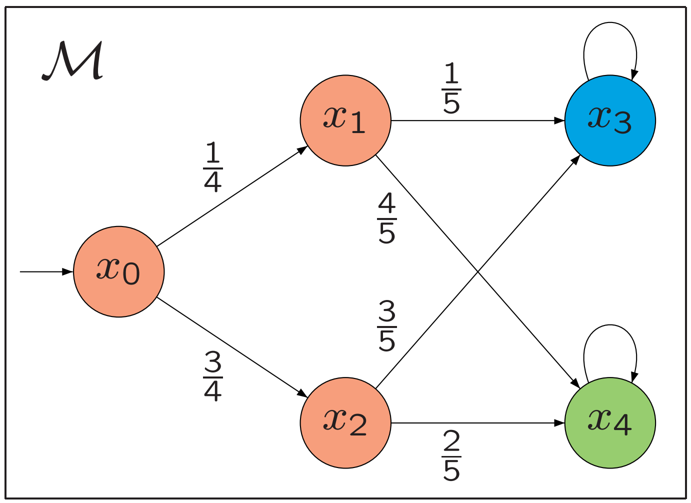
Labeled Markov Chains



State space	$\{x_0, \ldots, x_4\}$
Initial dist.	$(1, 0, 0, 0, 0)$
Prob. trans.	$\delta(x_0, x_1) = \frac{1}{4}$, etc.
Observations	$\Sigma = \{$●, ●, ●$\}$
Labeling	$\mathcal{L}(x_0) =$ ●, etc.

$$Pr(x_0 \cdot x_1 \cdot x_3) = 1 \times \frac{1}{4} \times \frac{1}{5} = \frac{1}{20}$$

Labeled Markov Chains

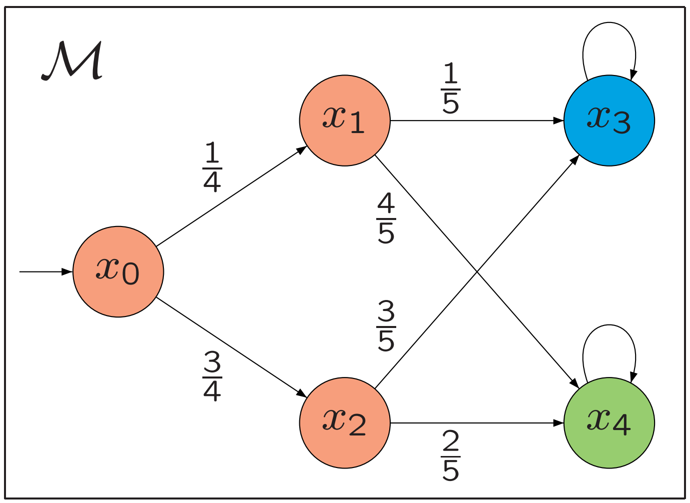


State space	$\{x_0, \ldots, x_4\}$
Initial dist.	$(1, 0, 0, 0, 0)$
Prob. trans.	$\delta(x_0, x_1) = \frac{1}{4}$, etc.
Observations	$\Sigma = \{\bullet, \bullet, \bullet\}$
Labeling	$\mathcal{L}(x_0) = \bullet$, etc.

$$Pr(x_0 \cdot x_1 \cdot x_3) = 1 \times \frac{1}{4} \times \frac{1}{5} = \frac{1}{20}$$

$$Pr(\bullet \cdot \bullet \cdot \bullet) = \frac{1}{20} + \frac{9}{20} = \frac{1}{2}$$

Labeled Markov Chains



State space	$\{x_0, \ldots, x_4\}$
Initial dist.	$(1, 0, 0, 0, 0)$
Prob. trans.	$\delta(x_0, x_1) = \frac{1}{4}$, etc.
Observations	$\Sigma = \{\textcolor{salmon}{\bullet}, \textcolor{cyan}{\bullet}, \textcolor{green}{\bullet}\}$
Labeling	$\mathcal{L}(x_0) = \textcolor{salmon}{\bullet}$, etc.

$$Pr(x_0 \cdot x_1 \cdot x_3) = 1 \times \frac{1}{4} \times \frac{1}{5} = \frac{1}{20}$$

$$Pr(\textcolor{salmon}{\bullet} \cdot \textcolor{salmon}{\bullet} \cdot \textcolor{cyan}{\bullet}) = \frac{1}{20} + \frac{9}{20} = \frac{1}{2}$$

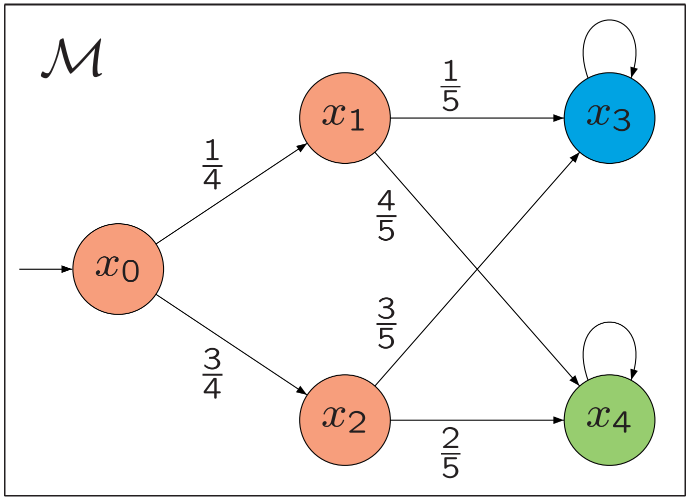
$\mathcal{M}$ defines the trace distribution $Pr^{\mathcal{M}} : \Sigma^+ \to [0, 1]$

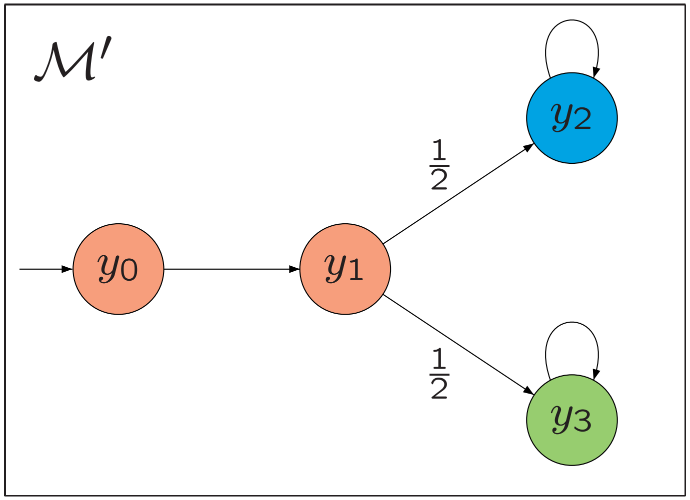
Trace equivalence

Definition: Two Markov chains are equivalent if they define the same trace distribution.

Trace equivalence

Definition: Two Markov chains are equivalent if they define the same trace distribution.





$\mathcal{M}$ and $\mathcal{M}'$ are equivalent.

LTL equivalence

LTL-formulas: $\varphi ::= \sigma \mid \neg\varphi \mid \varphi \wedge \varphi \mid X\varphi \mid \varphi_1 U \varphi_2 \qquad \sigma \in \Sigma$

Given an LTL-formula φ over Σ, $Pr^{\mathcal{M}}([\![\varphi]\!])$ is the probability that the trace generated by $\mathcal{M}$ satisfies φ.

(Note: A trace distribution can be uniquely extended to a probability measure over Borel subsets of Σ^{ω})

LTL equivalence

LTL-formulas: $\varphi ::= \sigma \mid \neg\varphi \mid \varphi \wedge \varphi \mid X\varphi \mid \varphi_1 U \varphi_2 \qquad \sigma \in \Sigma$

Given an LTL-formula φ over Σ, $Pr^{\mathcal{M}}([\![\varphi]\!])$ is the probability that the trace generated by $\mathcal{M}$ satisfies φ.

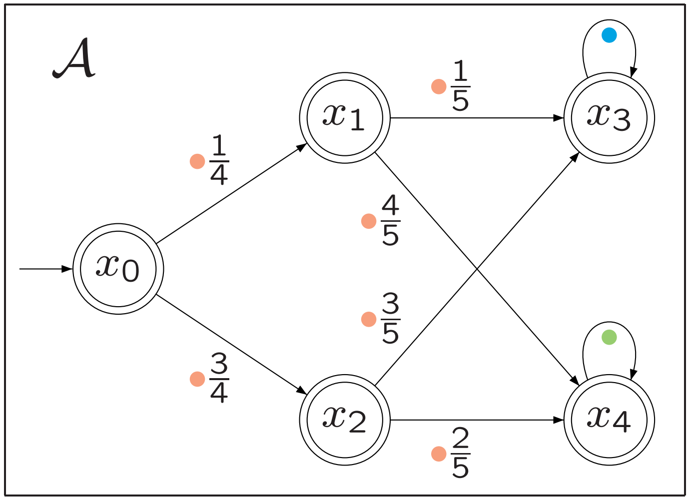
(Note: A trace distribution can be uniquely extended to a probability measure over Borel subsets of Σ^ω)

Theorem: Two Markov chains $\mathcal{M}, \mathcal{M}'$ are equivalent if and only if they are indistinguishable by LTL-formulas (*i.e.* $\forall\varphi : Pr^{\mathcal{M}}([\![\varphi]\!]) = Pr^{\mathcal{M}'}([\![\varphi]\!])$).

Outline

- Labeled Markov Chains
- Probabilistic Automata
- An algorithm for trace equivalence
- Markov Decision Processes

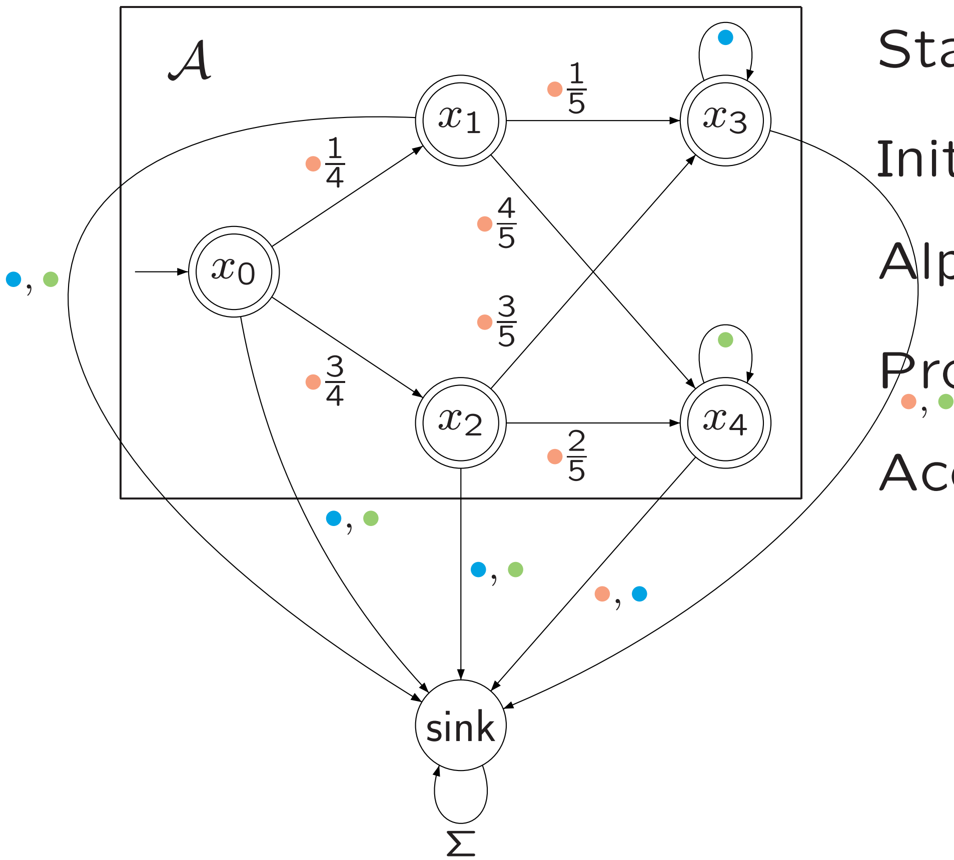
Probabilistic Automata



State space	$Q = \{x_0, \ldots, x_4\}$
Initial dist.	$\rho_0 \in \mathcal{D}(Q)$
Alphabet	$\Sigma = \{\bullet, \bullet, \bullet\}$
Prob. trans.	$\delta : Q \times \Sigma \rightarrow \mathcal{D}(Q)$
Accepting	$F \subseteq Q$

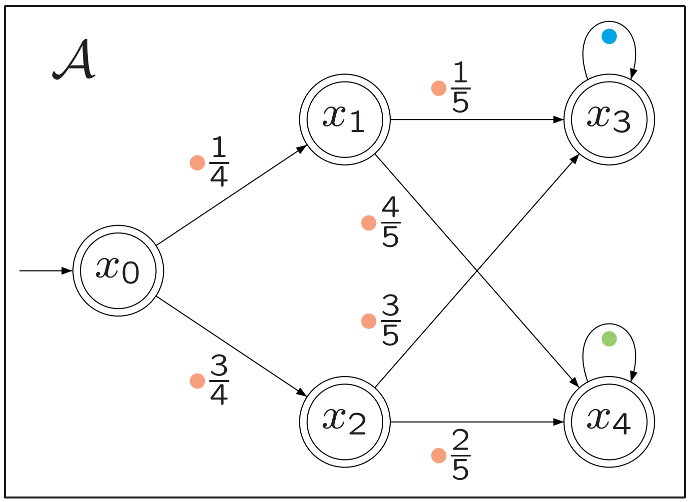
$$\mathcal{D}(Q) = \{f : Q \rightarrow [0, 1] \mid \sum_q f(q) = 1\}$$

Probabilistic Automata



State space	$Q = \{x_0, \ldots, x_4, \mathsf{sink}\}$
Initial dist.	$\rho_0 \in \mathcal{D}(Q)$
Alphabet	$\Sigma = \{$●, ●, ●$\}$
Prob. trans.	$\delta : Q \times \Sigma \to \mathcal{D}(Q)$
Accepting	$F \subseteq Q$

Probabilistic Automata



State space	$Q = \{x_0, \ldots, x_4\}$
Initial dist.	$\rho_0 \in \mathcal{D}(Q)$
Alphabet	$\Sigma = \{{\color{orange}\bullet}, {\color{blue}\bullet}, {\color{green}\bullet}\}$
Prob. trans.	$\delta : Q \times \Sigma \to \mathcal{D}(Q)$
Accepting	$F \subseteq Q$

The word $w = {\color{orange}\bullet} \cdot {\color{orange}\bullet} \cdot {\color{blue}\bullet}$ has (accepting) runs:

$$\left.\begin{array}{l} x_0 \xrightarrow{\color{orange}\bullet} x_1 \xrightarrow{\color{orange}\bullet} x_3 \xrightarrow{\color{blue}\bullet} x_3 \\ x_0 \xrightarrow{\color{orange}\bullet} x_2 \xrightarrow{\color{orange}\bullet} x_3 \xrightarrow{\color{blue}\bullet} x_3 \end{array}\right\} \frac{1}{20} + \frac{9}{20} = \frac{1}{2}$$

$\mathcal{A}$ defines the trace distribution $Pr^{\mathcal{A}} : \Sigma^+ \to [0,1]$

Probabilistic Automata

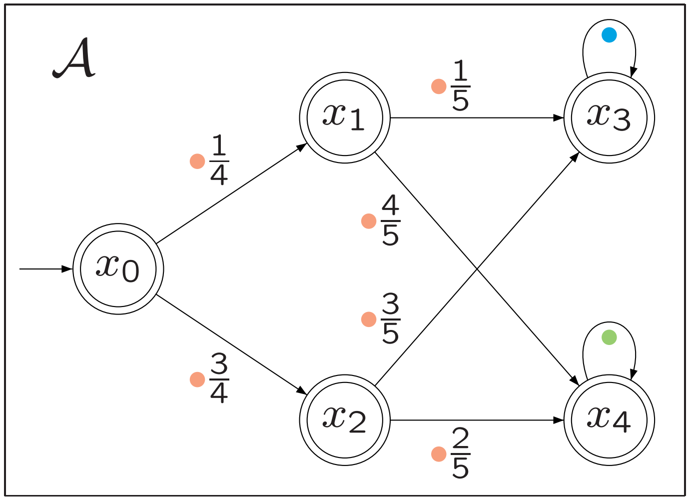
Definition: Two Probabilistic Automata are equivalent if they define the same trace distribution.

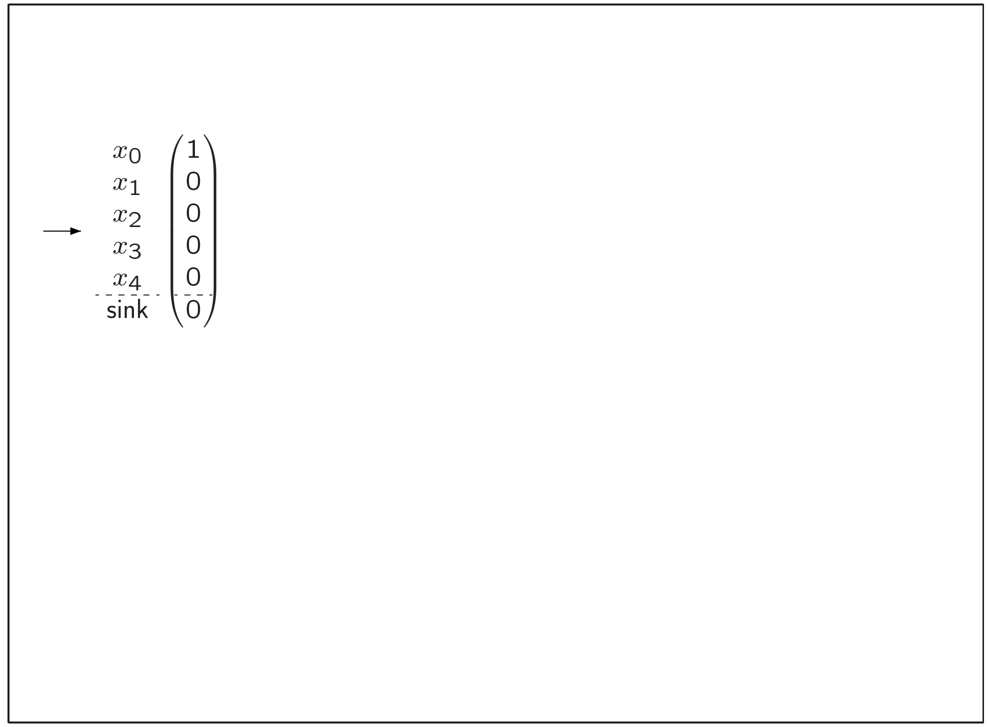
Straightforward (linear-time) transformation of MC to PA that preserves trace distribution.

Decidability of equivalence for PA entails decidability of equivalence for MC.

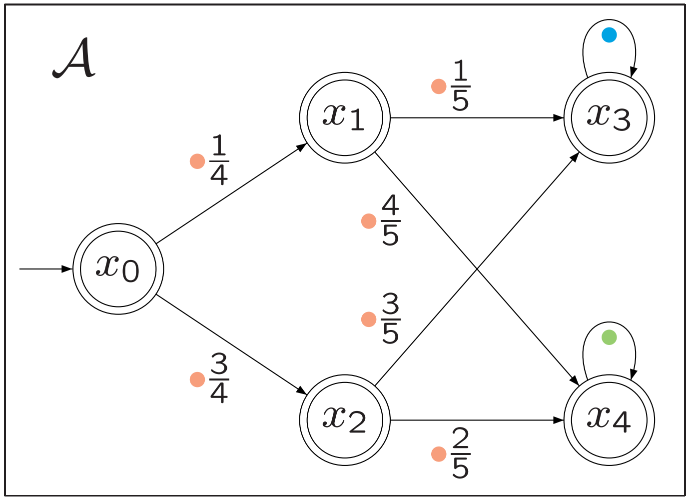
Equivalence for PA is decidable [Tze92].

State Distribution Graph



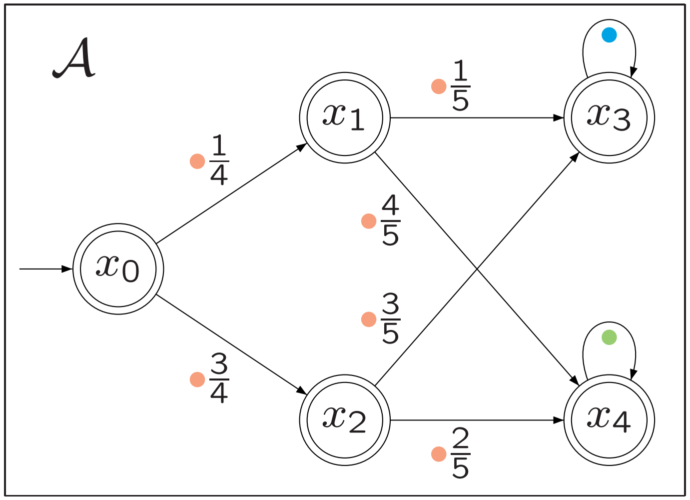


State Distribution Graph



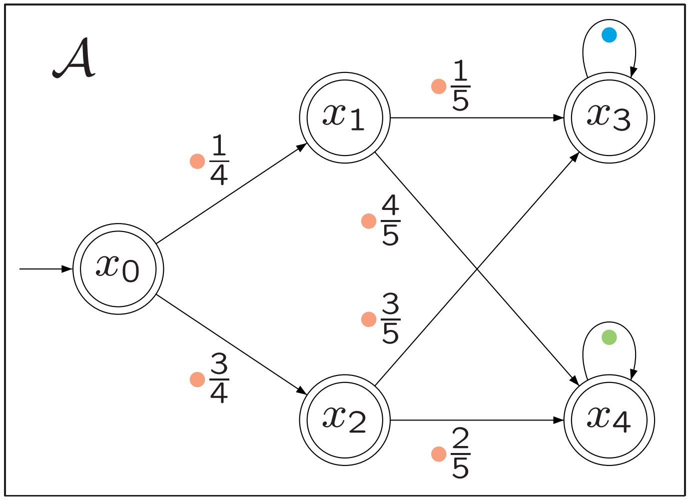
$$\longrightarrow \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \text{sink} \end{matrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 1/4 \\ 3/4 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

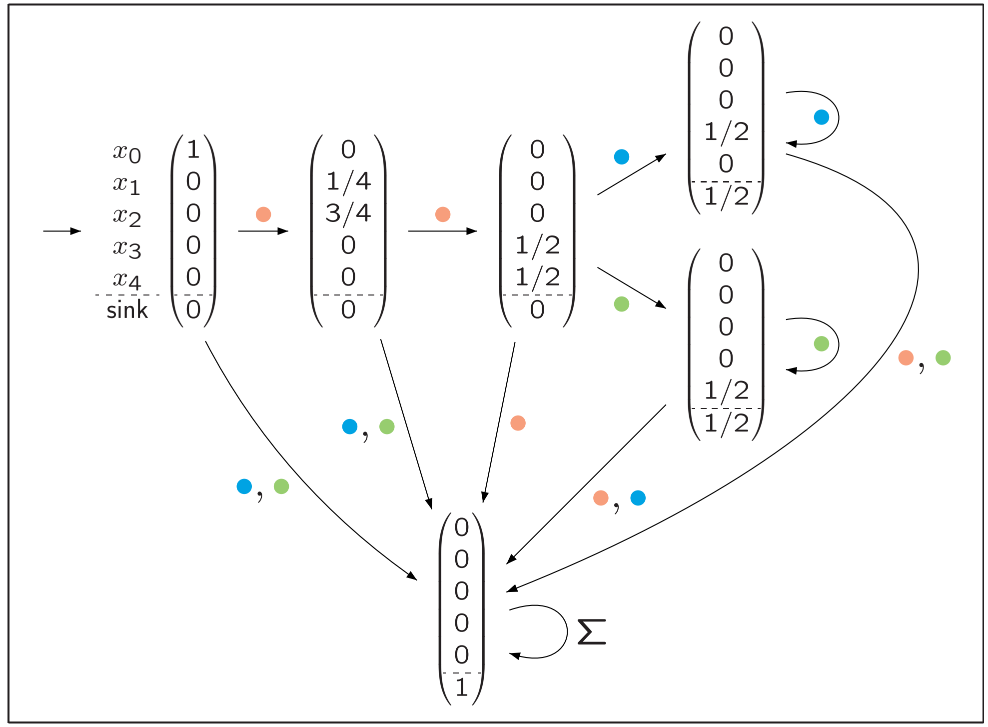
State Distribution Graph



$$\longrightarrow \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \hdashline \text{sink} \end{matrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \hdashline 0 \end{pmatrix} \xrightarrow{\bullet} \begin{pmatrix} 0 \\ 1/4 \\ 3/4 \\ 0 \\ 0 \\ \hdashline 0 \end{pmatrix} \xrightarrow{\bullet} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/2 \\ 1/2 \\ \hdashline 0 \end{pmatrix}$$

State Distribution Graph

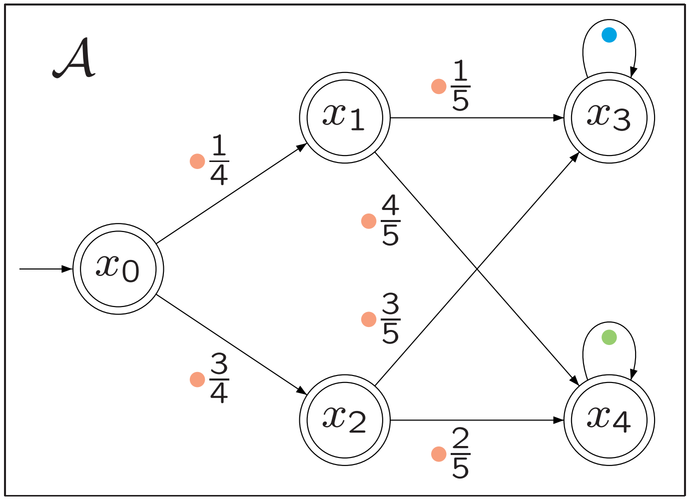


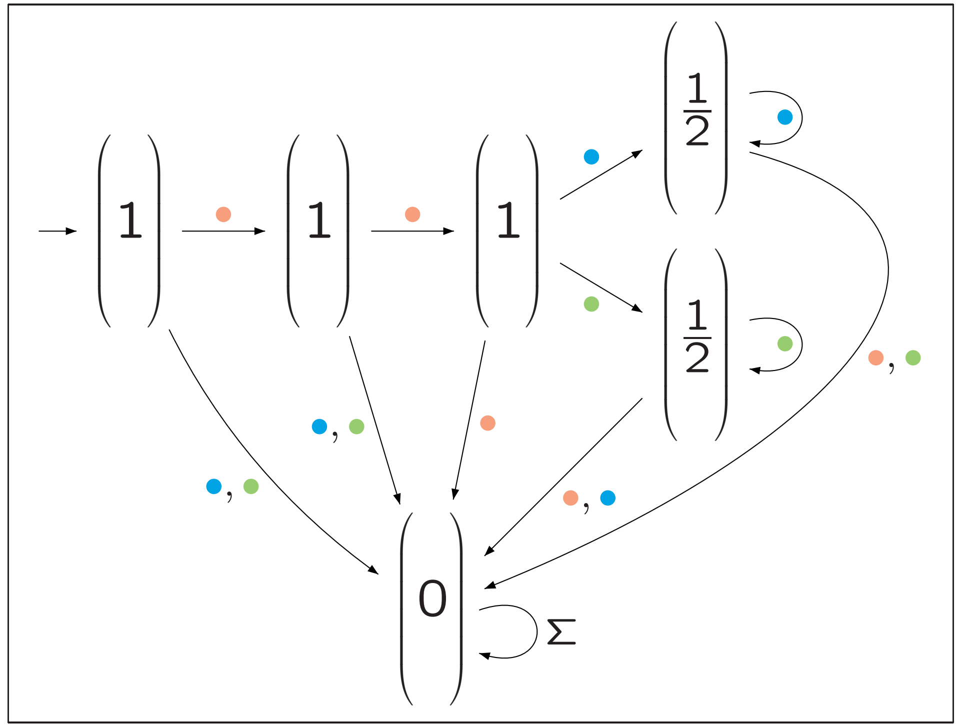


Finite state space: Q

Infinite state space: $\mathcal{D}(Q)$

State Distribution Graph





Finite state space: Q

Infinite labeling alphabet: $[0, 1]$

Bisimulation for probabilistic automata

Definition: $\approx \subseteq \mathcal{D}(Q^1) \times \mathcal{D}(Q^2)$ is a bisimulation for $\mathcal{A}_1$, $\mathcal{A}_2$ if for all $X \approx Y$:

(i) $\sum_{q \in F^1} X(q) = \sum_{q \in F^2} Y(q)$ and

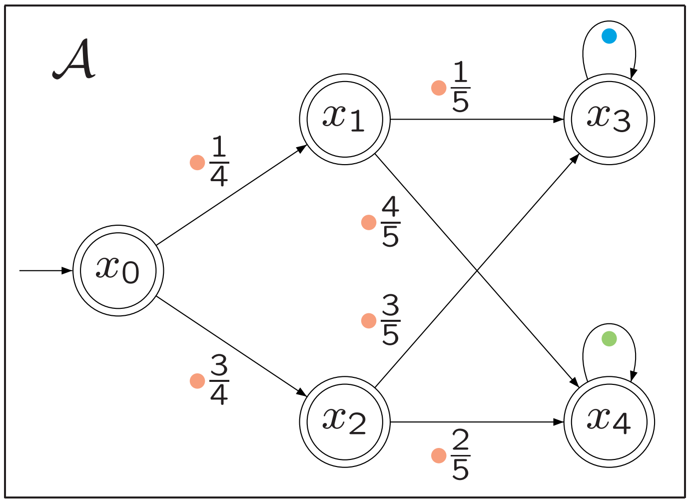
(ii) $\delta^1(X, \sigma) \approx \delta^2(Y, \sigma)$ for all $\sigma \in \Sigma$.

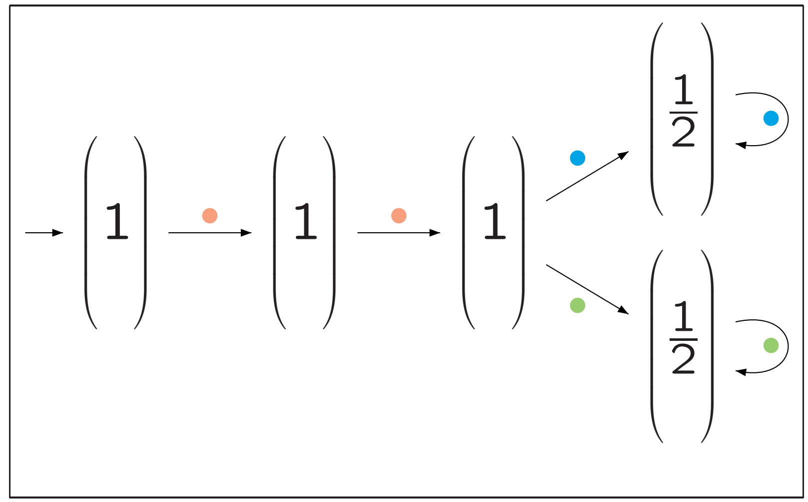
where the distribution $X' = \delta^1(X, \sigma)$ is defined by

$$X'(q') = \sum_q X(q) \times \delta^1(q, \sigma)(q')$$

We say that $\mathcal{A}_1$, $\mathcal{A}_2$ are bisimilar if there exists a bisimulation for them that contains (ρ_0^1, ρ_0^2).

State Distribution Graph





Theorem: Bisimulation and trace equivalence coincide.

Proof: properties of deterministic automata.

Outline

- Labeled Markov Chains
- Probabilistic Automata
- An algorithm for trace equivalence
- Markov Decision Processes

Deciding bisimulation

Question: Does there exist a bisimulation $\approx$ such that $\rho_0^1 \approx \rho_0^2$?

Solution Compute the largest bisimulation $\approx$ and check $\rho_0^1 \approx \rho_0^2$.

a) Symbolic representation of state distribution

b) Fixpoint iteration

Deciding bisimulation

a) **Symbolic representation of state distributions**

$X = (x_1, \ldots, x_{n_1})$ and $Y = (y_1, \ldots, y_{n_2})$ where $n_1 = |Q^1|$ and $n_2 = |Q^2|$.

$$\phi_0 \equiv \sum_{q_i \in F^1} x_i = \sum_{q_i \in F^2} y_i$$
$$\Big(\wedge \bigwedge_i x_i \geq 0 \wedge \sum_i x_i = 1 \wedge \bigwedge_j y_j \geq 0 \wedge \sum_j y_j = 1 \Big)$$

Deciding bisimulation

a) **Symbolic representation of state distributions**

$X = (x_1, \dots, x_{n_1})$ and $Y = (y_1, \dots, y_{n_2})$ where $n_1 = |Q^1|$ and $n_2 = |Q^2|$.

$$\phi_0 \equiv \sum_{q_i \in F^1} x_i = \sum_{q_i \in F^2} y_i$$
$$\Big(\wedge \bigwedge_i x_i \geq 0 \wedge \sum_i x_i = 1 \wedge \bigwedge_j y_j \geq 0 \wedge \sum_j y_j = 1 \Big)$$

b) **Fixpoint iteration**

$$\phi_{i+1} \equiv \phi_i \wedge \bigwedge_{\sigma \in \Sigma} \phi_i[X, Y \leftarrow \delta^1(X, \sigma), \delta^2(Y, \sigma)]$$

Deciding bisimulation

Correctness: if $\phi_{i+1} \equiv \phi_i$, then $\{(X,Y) \mid \phi_i(X,Y) \text{ holds}\}$ is the largest bisimulation for $\mathcal{A}_1$, $\mathcal{A}_2$.

Termination: if $\phi_{i+1} \not\equiv \phi_i$, then ϕ_{i+1} contains one more equality constraint than ϕ_i.

Hence, $\phi_{i+1} \equiv \phi_i$ for some $i \leq n_1 + n_2$.

Deciding bisimulation

Correctness: if $\phi_{i+1} \equiv \phi_i$, then $\{(X, Y) \mid \phi_i(X, Y) \text{ holds}\}$ is the largest bisimulation for $\mathcal{A}_1$, $\mathcal{A}_2$.

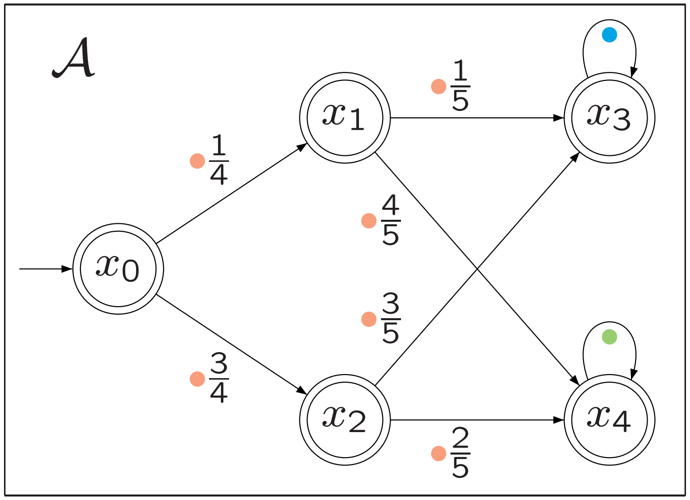
Termination: if $\phi_{i+1} \not\equiv \phi_i$, then ϕ_{i+1} contains one more equality constraint than ϕ_i.

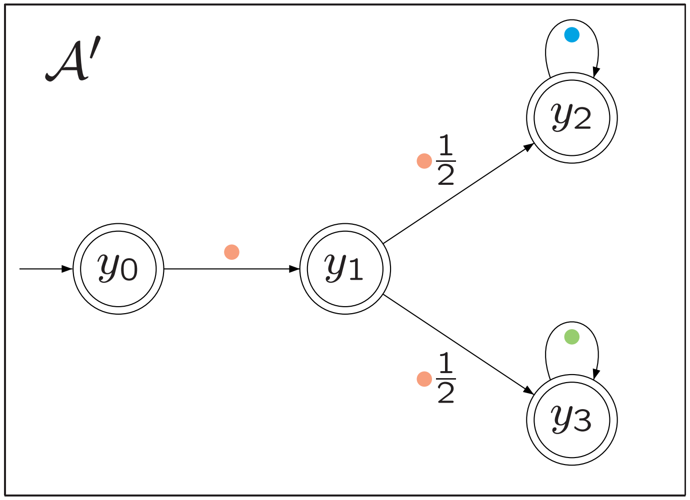
Hence, $\phi_{i+1} \equiv \phi_i$ for some $i \leq n_1 + n_2$.

Complexity: $O((n_1 + n_2)^4)$

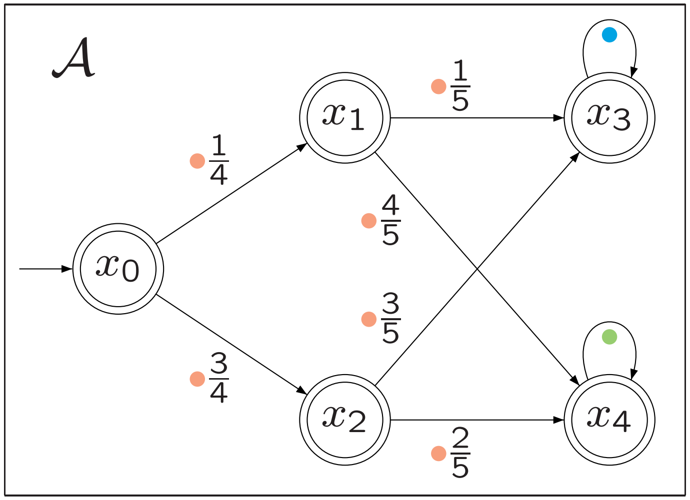
- $n_1 + n_2$ iterations
- matrix mult., linear independence: $O((n_1 + n_2)^3)$

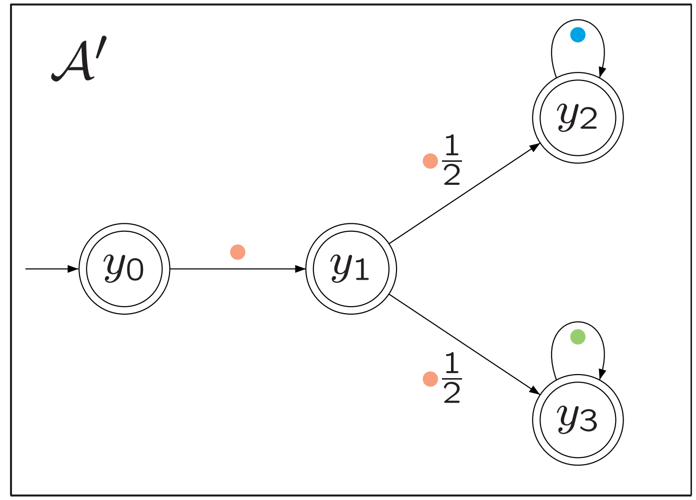
Example


$\mathcal{A}$
x_0
x_1
x_2
x_3
x_4
$\frac{1}{4}$
$\frac{3}{4}$
$\frac{1}{5}$
$\frac{4}{5}$
$\frac{3}{5}$
$\frac{2}{5}$


$\mathcal{A}'$
y_0
y_1
y_2
y_3
$\frac{1}{2}$
$\frac{1}{2}$

Example





$$\delta(X, {\color{salmon}\bullet}) = (0, \frac{x_0}{4}, \frac{3x_0}{4}, \frac{x_1}{5} + \frac{3x_2}{5}, \frac{4x_1}{5} + \frac{2x_2}{5})$$

$$\delta(Y, {\color{salmon}\bullet}) = (0, y_0, \frac{y_1}{2}, \frac{y_1}{2})$$

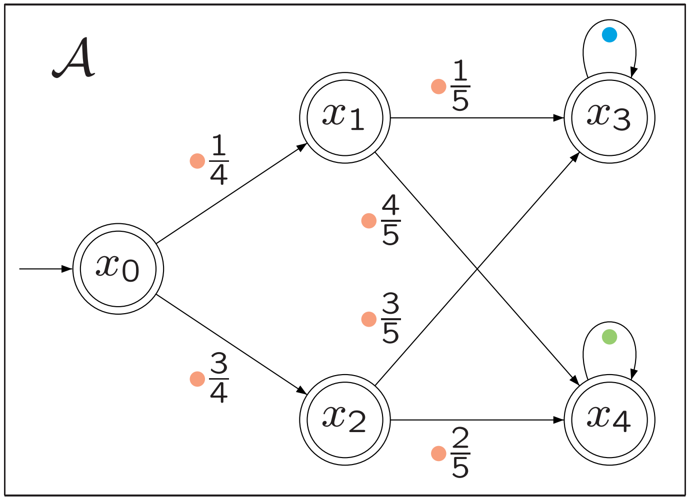
$$\delta(X, {\color{cyan}\bullet}) = (0, 0, 0, x_3, 0)$$

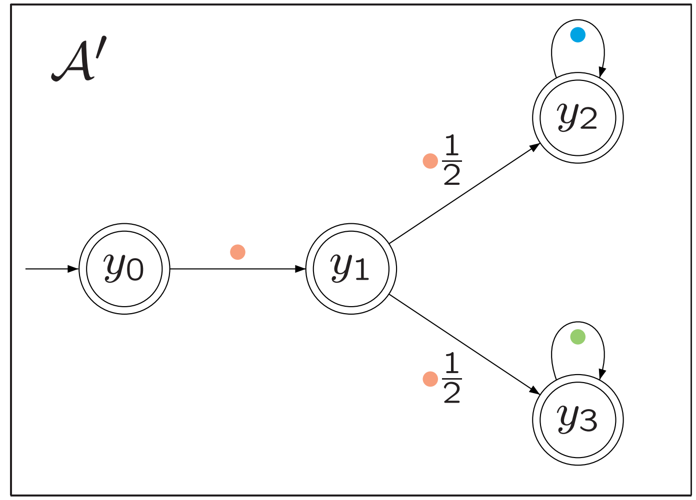
$$\delta(Y, {\color{cyan}\bullet}) = (0, 0, y_2, 0)$$

$$\delta(X, {\color{green}\bullet}) = (0, 0, 0, 0, x_4)$$

$$\delta(Y, {\color{green}\bullet}) = (0, 0, 0, y_3)$$

Example





$$\delta(X, {\color{orange}\bullet}) = \left(0, \frac{x_0}{4}, \frac{3x_0}{4}, \frac{x_1}{5} + \frac{3x_2}{5}, \frac{4x_1}{5} + \frac{2x_2}{5}\right)$$

$$\delta(Y, {\color{orange}\bullet}) = \left(0, y_0, \frac{y_1}{2}, \frac{y_1}{2}\right)$$

$$\delta(X, {\color{cyan}\bullet}) = (0, 0, 0, x_3, 0)$$

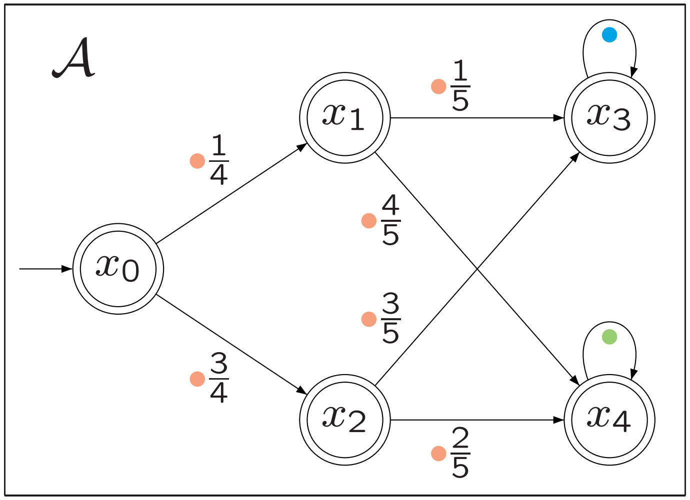
$$\delta(Y, {\color{cyan}\bullet}) = (0, 0, y_2, 0)$$

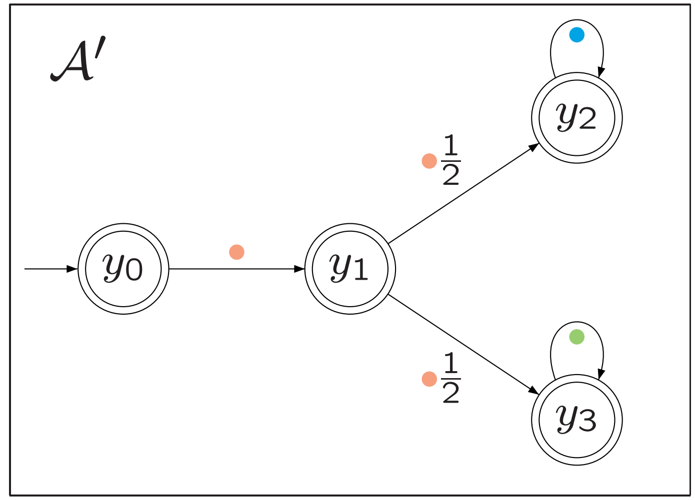
$$\delta(X, {\color{green}\bullet}) = (0, 0, 0, 0, x_4)$$

$$\delta(Y, {\color{green}\bullet}) = (0, 0, 0, y_3)$$

$$\phi_0 \equiv x_0 + x_1 + x_2 + x_3 + x_4 = y_0 + y_1 + y_2 + y_3$$

Example





$$\delta(X, \bullet) = (0, \frac{x_0}{4}, \frac{3x_0}{4}, \frac{x_1}{5} + \frac{3x_2}{5}, \frac{4x_1}{5} + \frac{2x_2}{5})$$

$$\delta(X, \bullet) = (0, 0, 0, x_3, 0)$$

$$\delta(X, \bullet) = (0, 0, 0, 0, x_4)$$

$$\delta(Y, \bullet) = (0, y_0, \frac{y_1}{2}, \frac{y_1}{2})$$

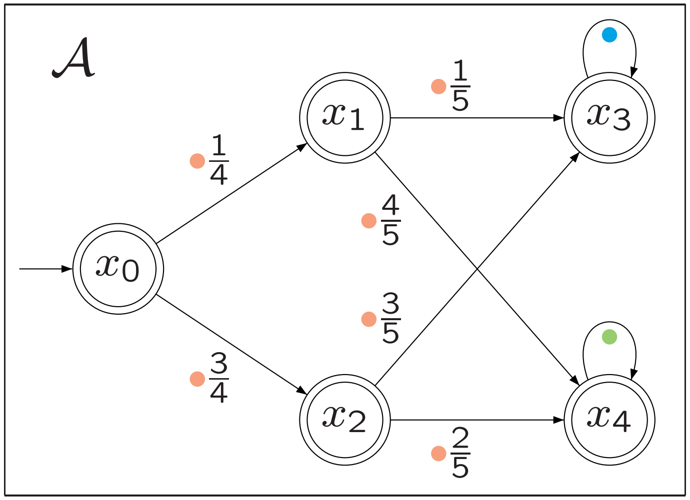
$$\delta(Y, \bullet) = (0, 0, y_2, 0)$$

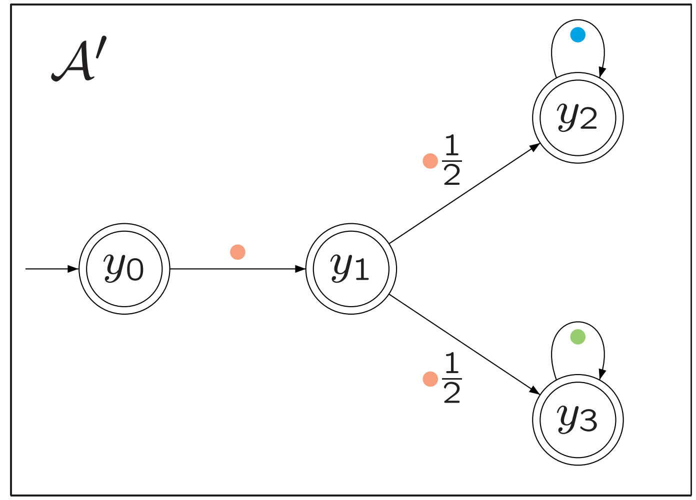
$$\delta(Y, \bullet) = (0, 0, 0, y_3)$$

$$\phi_0 \equiv x_0 + x_1 + x_2 + x_3 + x_4 = y_0 + y_1 + y_2 + y_3$$

$$\phi_1 \equiv \phi_0 \wedge x_0 + x_1 + x_2 = y_0 + y_1 \wedge x_3 = y_2 \wedge x_4 = y_3$$

Example





$$\delta(X, {\color{orange}\bullet}) = (0, \frac{x_0}{4}, \frac{3x_0}{4}, \frac{x_1}{5} + \frac{3x_2}{5}, \frac{4x_1}{5} + \frac{2x_2}{5})$$

$$\delta(X, {\color{cyan}\bullet}) = (0, 0, 0, x_3, 0)$$

$$\delta(X, {\color{green}\bullet}) = (0, 0, 0, 0, x_4)$$

$$\delta(Y, {\color{orange}\bullet}) = (0, y_0, \frac{y_1}{2}, \frac{y_1}{2})$$

$$\delta(Y, {\color{cyan}\bullet}) = (0, 0, y_2, 0)$$

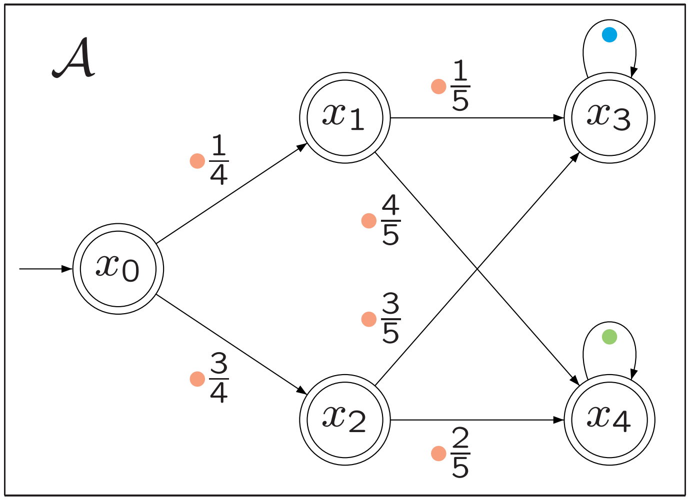
$$\delta(Y, {\color{green}\bullet}) = (0, 0, 0, y_3)$$

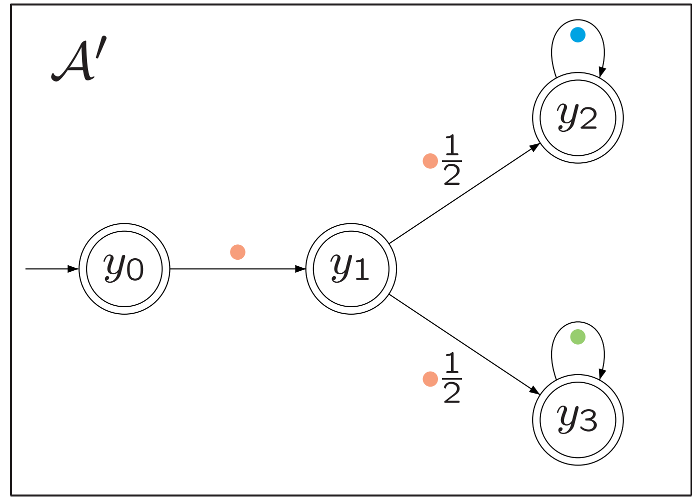
$$\phi_0 \equiv x_0 + x_1 + x_2 + x_3 + x_4 = y_0 + y_1 + y_2 + y_3$$

$$\phi_1 \equiv \phi_0 \wedge x_0 + x_1 + x_2 = y_0 + y_1 \wedge x_3 = y_2 \wedge \cancel{x_4 = y_3}$$

redundant

Example





$$\delta(X, \textcolor{orange}{\bullet}) = \left(0, \frac{\textcolor{red}{x_0}}{4}, \frac{3\textcolor{red}{x_0}}{4}, \frac{\textcolor{red}{x_1}}{5} + \frac{3\textcolor{red}{x_2}}{5}, \frac{4\textcolor{red}{x_1}}{5} + \frac{2\textcolor{red}{x_2}}{5}\right)$$

$$\delta(Y, \textcolor{orange}{\bullet}) = \left(0, \textcolor{red}{y_0}, \frac{\textcolor{red}{y_1}}{2}, \frac{\textcolor{red}{y_1}}{2}\right)$$

$$\delta(X, \textcolor{cyan}{\bullet}) = (0, 0, 0, \textcolor{blue}{x_3}, 0)$$

$$\delta(Y, \textcolor{cyan}{\bullet}) = (0, 0, \textcolor{blue}{y_2}, 0)$$

$$\delta(X, \textcolor{green}{\bullet}) = (0, 0, 0, 0, \textcolor{green}{x_4})$$

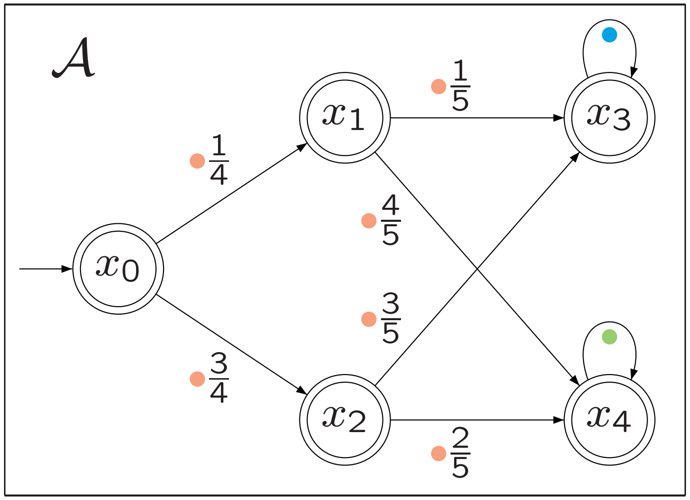
$$\delta(Y, \textcolor{green}{\bullet}) = (0, 0, 0, \textcolor{green}{y_3})$$

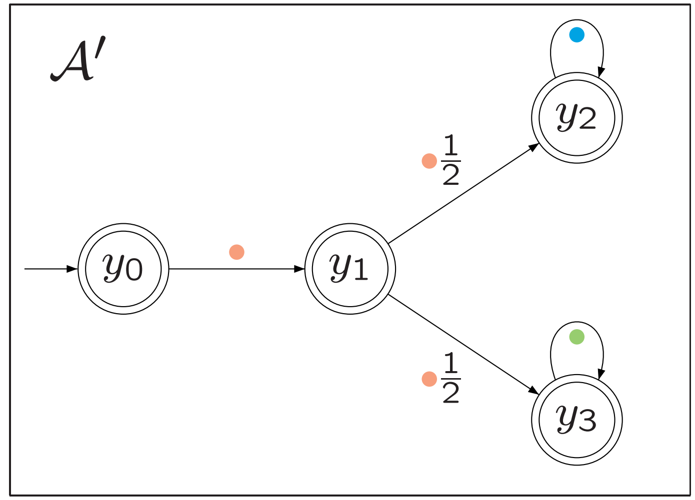
$$\phi_0 \equiv \textcolor{red}{x_0} + \textcolor{red}{x_1} + \textcolor{red}{x_2} + \textcolor{blue}{x_3} + \textcolor{green}{x_4} = \textcolor{red}{y_0} + \textcolor{red}{y_1} + \textcolor{blue}{y_2} + \textcolor{green}{y_3}$$

$$\phi_1 \equiv \phi_0 \wedge \textcolor{red}{x_0} + \textcolor{red}{x_1} + \textcolor{red}{x_2} = \textcolor{red}{y_0} + \textcolor{red}{y_1} \wedge \textcolor{blue}{x_3} = \textcolor{blue}{y_2}$$

$$\phi_2 \equiv \phi_1 \wedge \textcolor{red}{x_0} = \textcolor{red}{y_0} \wedge \frac{\textcolor{red}{x_1}}{5} + \frac{3\textcolor{red}{x_2}}{5} = \frac{\textcolor{red}{y_1}}{2}$$

Example





$$\delta(X, \bullet) = \left(0, \frac{x_0}{4}, \frac{3x_0}{4}, \frac{x_1}{5} + \frac{3x_2}{5}, \frac{4x_1}{5} + \frac{2x_2}{5}\right)$$

$$\delta(Y, \bullet) = \left(0, y_0, \frac{y_1}{2}, \frac{y_1}{2}\right)$$

$$\delta(X, \bullet) = (0, 0, 0, x_3, 0)$$

$$\delta(Y, \bullet) = (0, 0, y_2, 0)$$

$$\delta(X, \bullet) = (0, 0, 0, 0, x_4)$$

$$\delta(Y, \bullet) = (0, 0, 0, y_3)$$

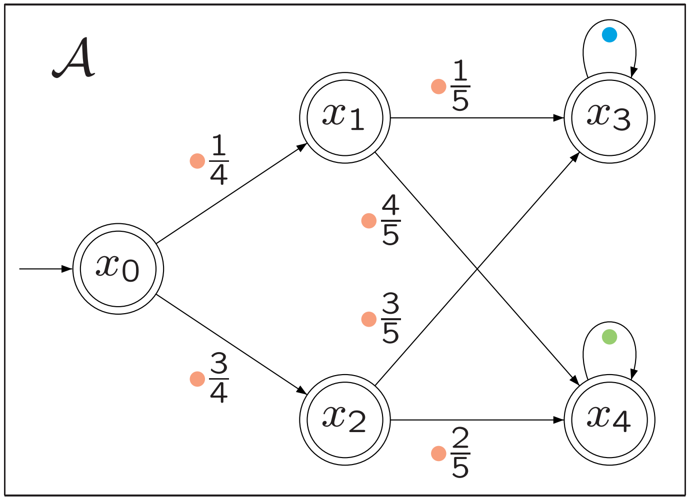
$$\phi_0 \equiv x_0 + x_1 + x_2 + x_3 + x_4 = y_0 + y_1 + y_2 + y_3$$

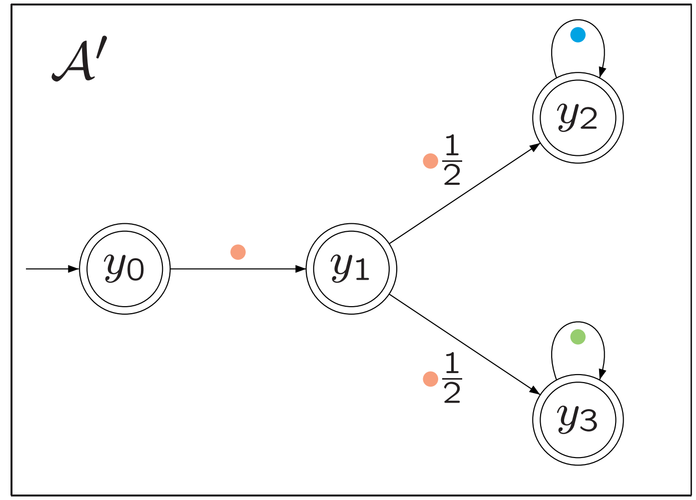
$$\phi_1 \equiv \phi_0 \wedge x_0 + x_1 + x_2 = y_0 + y_1 \wedge x_3 = y_2$$

$$\phi_2 \equiv \phi_1 \wedge x_0 = y_0 \wedge \frac{x_1}{5} + \frac{3x_2}{5} = \frac{y_1}{2}$$

$$\phi_3 \equiv \phi_2 \wedge \frac{x_0}{20} + \frac{9x_0}{20} = \frac{y_0}{2}$$

Example





$$\delta(X, {\color{orange}\bullet}) = (0, \tfrac{x_0}{4}, \tfrac{3x_0}{4}, \tfrac{x_1}{5} + \tfrac{3x_2}{5}, \tfrac{4x_1}{5} + \tfrac{2x_2}{5})$$

$$\delta(X, {\color{cyan}\bullet}) = (0, 0, 0, x_3, 0)$$

$$\delta(X, {\color{green}\bullet}) = (0, 0, 0, 0, x_4)$$

$$\delta(Y, {\color{orange}\bullet}) = (0, y_0, \tfrac{y_1}{2}, \tfrac{y_1}{2})$$

$$\delta(Y, {\color{cyan}\bullet}) = (0, 0, y_2, 0)$$

$$\delta(Y, {\color{green}\bullet}) = (0, 0, 0, y_3)$$

$$\phi_0 \equiv x_0 + x_1 + x_2 + x_3 + x_4 = y_0 + y_1 + y_2 + y_3$$

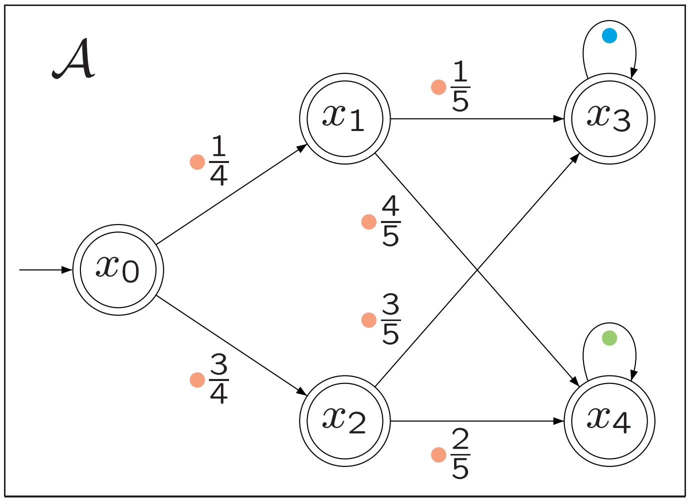
$$\phi_1 \equiv \phi_0 \wedge x_0 + x_1 + x_2 = y_0 + y_1 \wedge x_3 = y_2$$

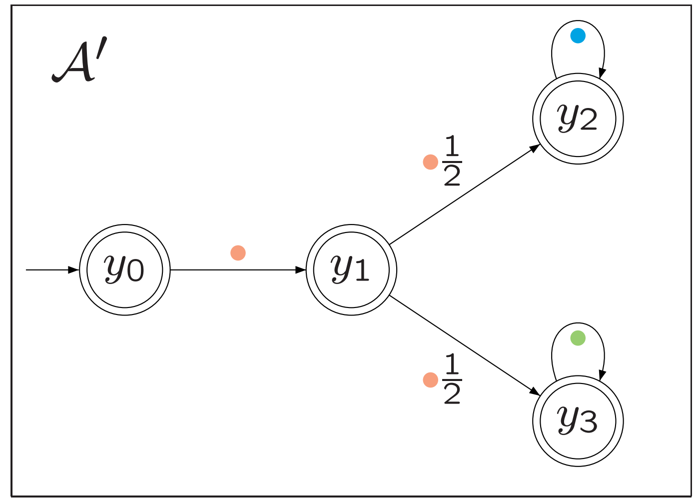
$$\phi_2 \equiv \phi_1 \wedge x_0 = y_0 \wedge \frac{x_1}{5} + \frac{3x_2}{5} = \frac{y_1}{2}$$

$$\phi_3 \equiv \phi_2 \wedge \frac{x_0}{20} + \frac{9x_0}{20} = \frac{y_0}{2}$$

redundant

Example





$$\phi^* \equiv \begin{cases} x_0 = y_0 \\ \frac{x_1}{5} + \frac{3x_2}{5} = \frac{y_1}{2} \\ \frac{4x_1}{5} + \frac{2x_2}{5} = \frac{y_1}{2} \\ x_3 = y_2 \\ x_4 = y_3 \end{cases} \quad \wedge \quad \begin{array}{l} \sum x_i = \sum y_i = 1 \\ x_i, y_i \geq 0 \end{array}$$

$\mathcal{A}_{(1,0,0,0,0)}$ and $\mathcal{A}'_{(1,0,0,0)}$ are bisimilar.

An algorithm for trace equivalence

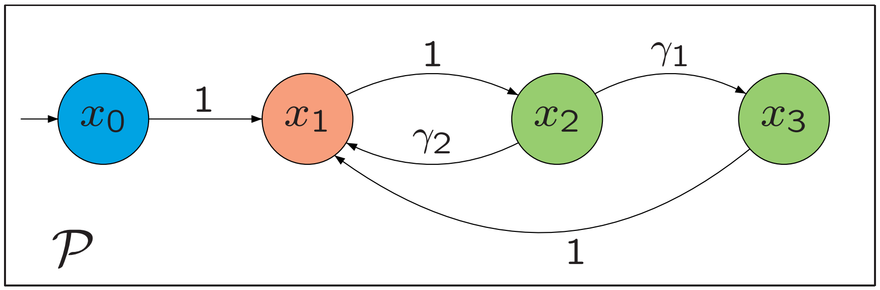
- Polynomial-time algorithm $O((n_1+n_2)^4)$ (same as [Tze92]).
- $O((n_1 + n_2)^2)$ for different initial distributions.
- Open problem #1: efficient minimization algorithm.
- Open problem #2: compute $\sup_{w\in\Sigma^+}|Pr^{\mathcal{A}_1}(w) - Pr^{\mathcal{A}_2}(w)|$.

Outline

- Labeled Markov Chains
- Probabilistic Automata
- An algorithm for trace equivalence
- Markov Decision Processes

Labeled Markov decision process

MDP $\equiv$ Markov chain + Decisions (nondeterminism)

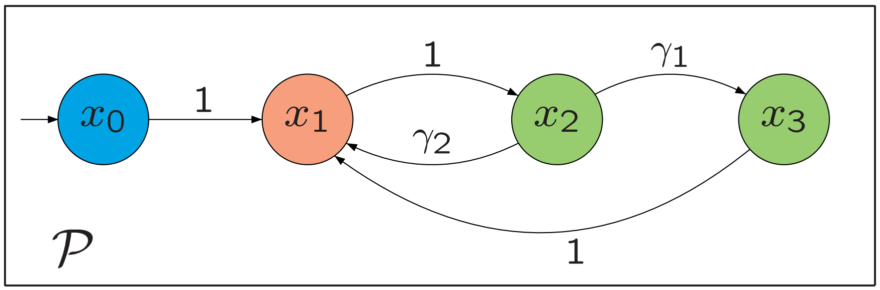


Choices $\Gamma = \{\gamma_1, \gamma_2\}$

Prob. trans. $\delta : Q \times \Gamma \to \mathcal{D}(Q)$

Labeled Markov decision process

MDP $\equiv$ Markov chain + Decisions (nondeterminism)



Choices $\Gamma = \{\gamma_1, \gamma_2\}$

Prob. trans. $\delta : Q \times \Gamma \to \mathcal{D}(Q)$

Decisions are resolved by a scheduler $\sigma : \underbrace{Q^+ \to \mathcal{D}(\Gamma)}_{\mathcal{S}}$

An MDP $\mathcal{P}$ executed under scheduler σ defines a Markov chain $\mathcal{P}(\sigma)$, and thus a trace distribution Pr_σ.

Trace refinement - Trace equivalence

Definition: Given two MDP $\mathcal{P}_1, \mathcal{P}_2$, we say that $\mathcal{P}_1$ refines $\mathcal{P}_2$ (written $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2$) if

$$\forall \sigma \in \mathcal{S}_1 \cdot \exists \tau \in \mathcal{S}_2 : \mathcal{P}_1(\sigma) \text{ and } \mathcal{P}_2(\tau) \text{ are equivalent.}$$

We define $\mathcal{P}_1 \equiv \mathcal{P}_2$ as $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2 \wedge \mathcal{P}_2 \sqsubseteq \mathcal{P}_1$

Trace refinement - Trace equivalence

Definition: Given two MDP $\mathcal{P}_1, \mathcal{P}_2$, we say that $\mathcal{P}_1$ refines $\mathcal{P}_2$ (written $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2$) if

$$\forall \sigma \in \mathcal{S}_1 \cdot \exists \tau \in \mathcal{S}_2 : \mathcal{P}_1(\sigma) \text{ and } \mathcal{P}_2(\tau) \text{ are equivalent.}$$

We define $\mathcal{P}_1 \equiv \mathcal{P}_2$ as $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2 \wedge \mathcal{P}_2 \sqsubseteq \mathcal{P}_1$

Open questions: Decide $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2$, decide $\mathcal{P}_1 \equiv \mathcal{P}_2$.

These questions are open even if $\mathcal{P}_1$ (or $\mathcal{P}_2$) is a Markov chain.

Trace refinement - Trace equivalence

Remark 1: Observation-based schedulers are sufficient.

A scheduler σ is observation-based if

$$\sigma(\overline{q}) = \sigma(\overline{q}\,') \text{ for all } \overline{q}, \overline{q}' \text{ such that } \mathcal{L}(\overline{q}) = \mathcal{L}(\overline{q}')$$

Trace refinement - Trace equivalence

Remark 1: Observation-based schedulers are sufficient.

A scheduler σ is observation-based if

$$\sigma(\bar{q}) = \sigma(\bar{q}') \text{ for all } \bar{q}, \bar{q}' \text{ such that } \mathcal{L}(\bar{q}) = \mathcal{L}(\bar{q}')$$

Let $\mathcal{P}$ be a labeled MDP.

Theorem: For all schedulers $\sigma \in \mathcal{S}$, there exists an observation-based scheduler $\sigma^{\mathsf{o}} \in \mathcal{S}$ such that $\mathcal{P}(\sigma)$ and $\mathcal{P}(\sigma^{\mathsf{o}})$ are equivalent.

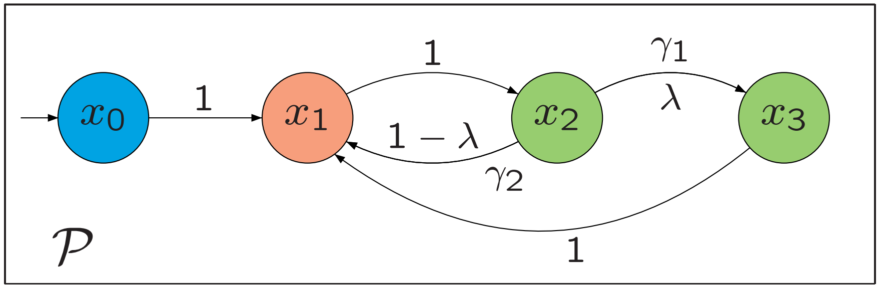
Trace refinement - Trace equivalence

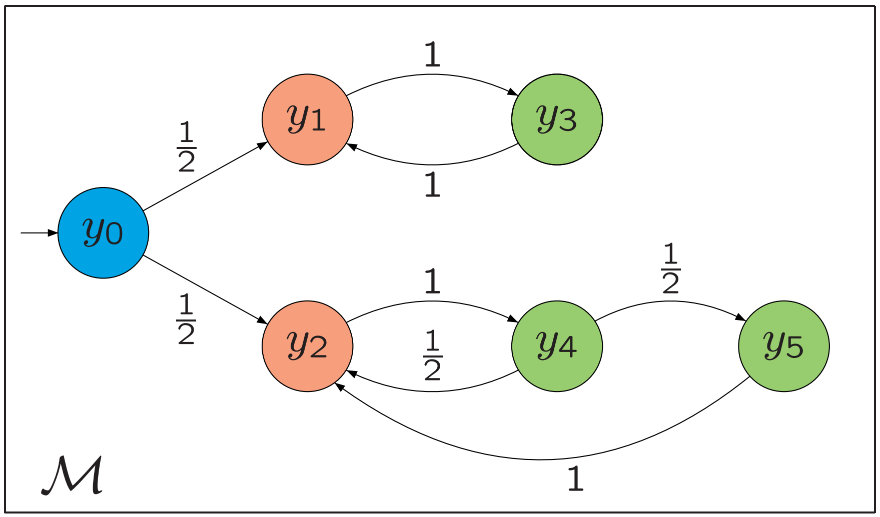
Remark 2: Finite-memory schedulers are not sufficient. A scheduler σ is finite-memory if there exist

- a finite set M (the memory),
- $m_0 \in M$,
- $update : M \times Q \to M$,
- $\mu : M \to \mathcal{D}(\Gamma)$

such that $\sigma(\bar{q}) = \mu(update(m_0, \bar{q}))$ for all $\bar{q} \in Q^+$.

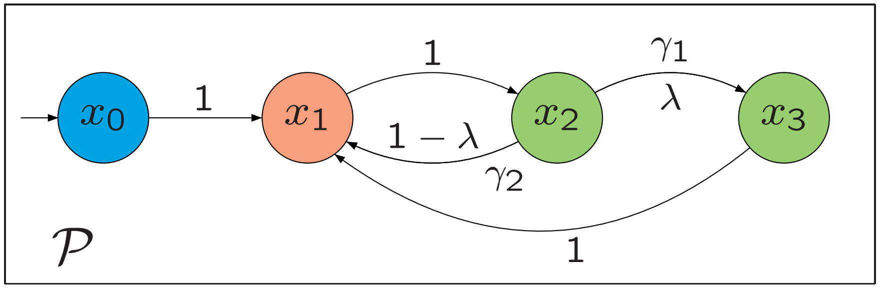
Finite-memory is not sufficient

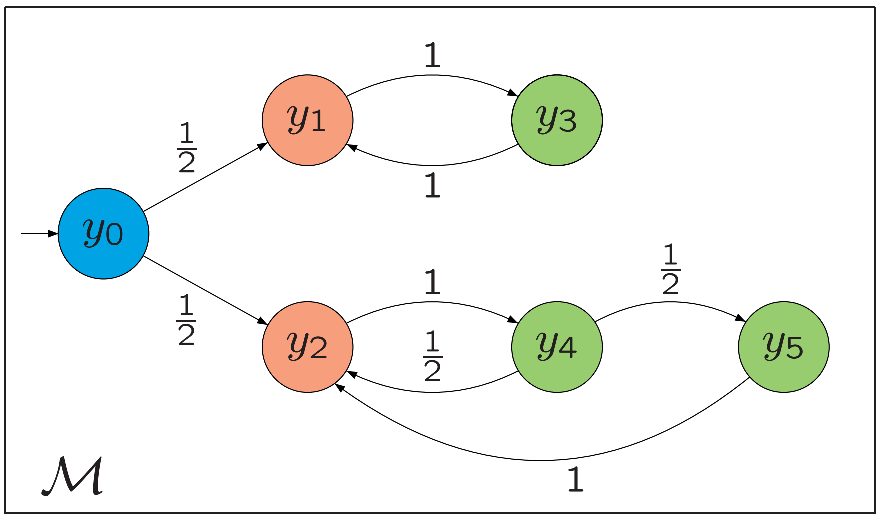




Does $\mathcal{M} \sqsubseteq \mathcal{P}$?

Finite-memory is not sufficient



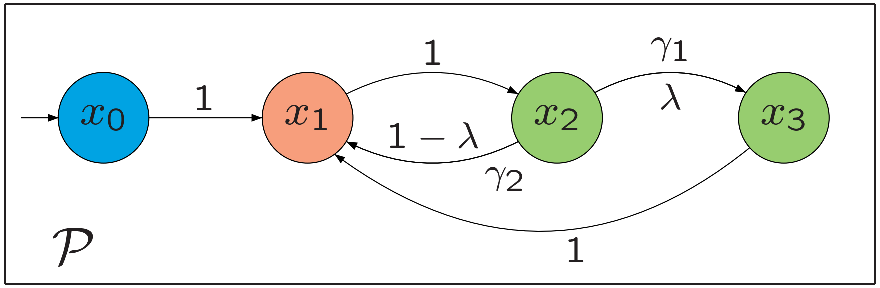


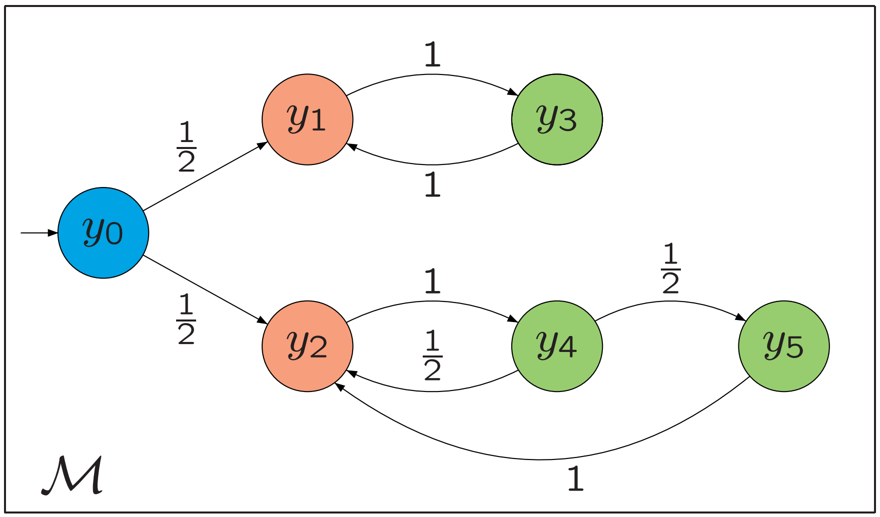
Does $\mathcal{M} \sqsubseteq \mathcal{P}$?

$\lambda(x_0\, x_1\, x_2) = \frac{1}{4}$

Yes ! but with infinite memory.

Finite-memory is not sufficient



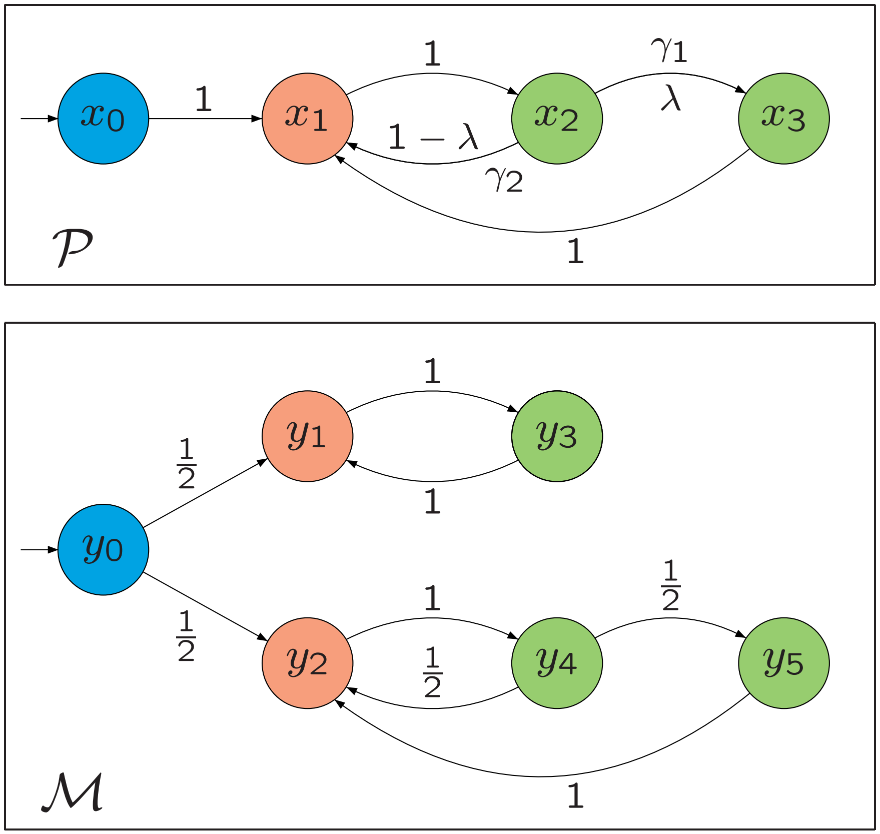


Does $\mathcal{M} \sqsubseteq \mathcal{P}$? Yes ! but with infinite memory.

$\lambda(x_0\, x_1\, x_2) = \frac{1}{4}$

$\lambda(x_0\,(Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \frac{1}{2}$

Finite-memory is not sufficient



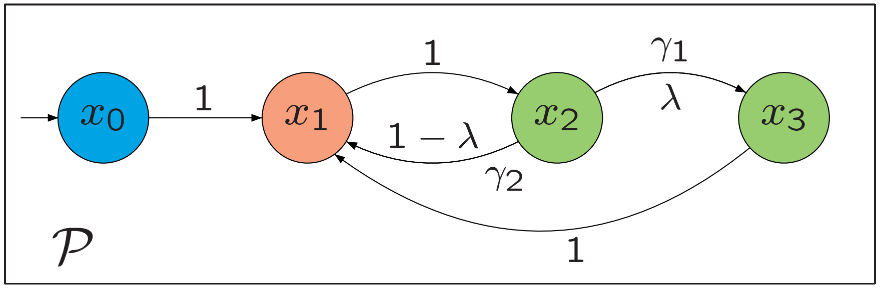
Does $\mathcal{M} \sqsubseteq \mathcal{P}$?   Yes ! but with infinite memory.

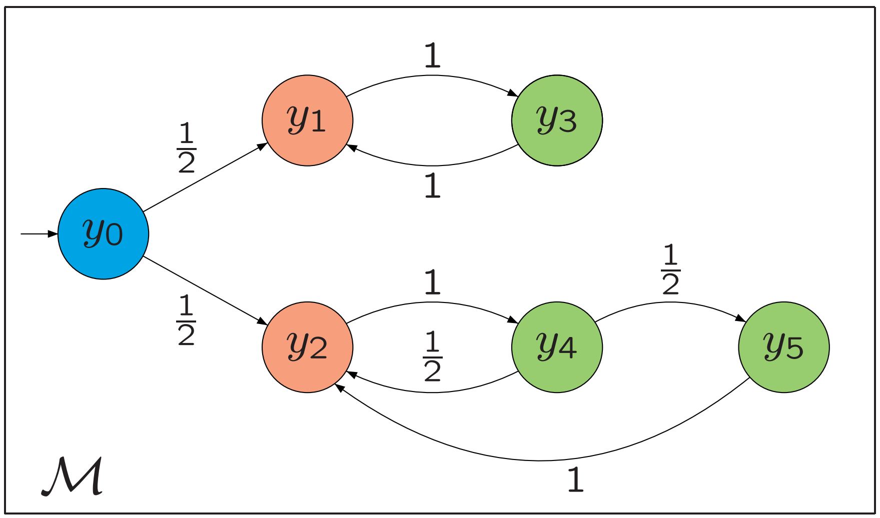
$$\lambda(x_0\, x_1\, x_2) = \tfrac{1}{4}$$

$$\lambda(x_0\, (Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \tfrac{1}{2}$$

$$\lambda(x_0\, (x_1\, x_2)^n) = ?$$

Finite-memory is not sufficient





Does $\mathcal{M} \sqsubseteq \mathcal{P}$? Yes ! but with infinite memory.

$\lambda(x_0\, x_1\, x_2) = \frac{1}{4}$

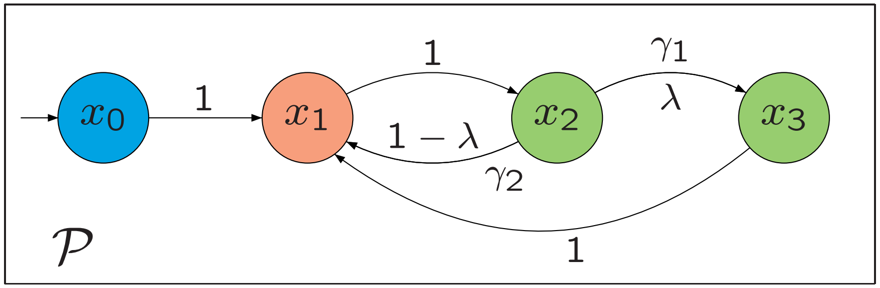
$\lambda(x_0\, (Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \frac{1}{2}$

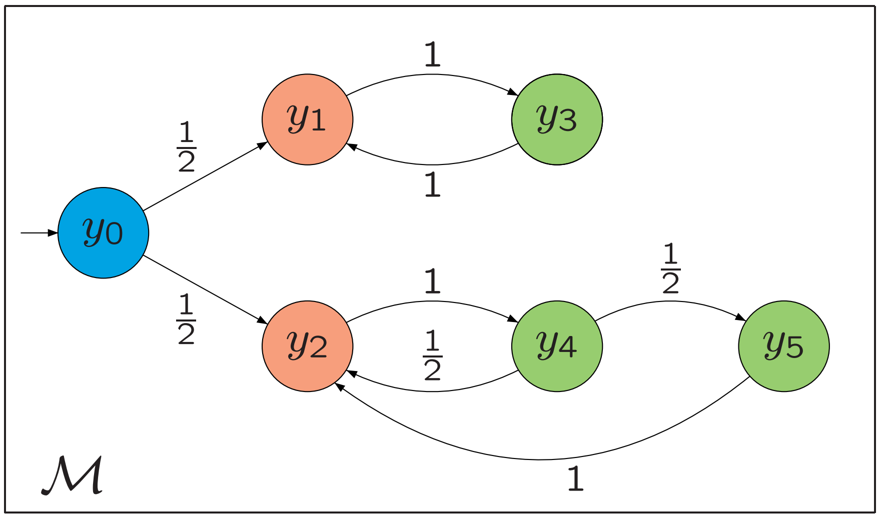
$\lambda(x_0\, (x_1\, x_2)^n) = ?$

$P^{\mathcal{M}}(\bullet \mid \bullet\, (\bullet\, \bullet)^n) = ?$

$P^{\mathcal{M}}(\bullet\, (\bullet\, \bullet)^n) = ?$

Finite-memory is not sufficient





Does $\mathcal{M} \sqsubseteq \mathcal{P}$? Yes ! but with infinite memory.

$\lambda(x_0\, x_1\, x_2) = \frac{1}{4}$

$\lambda(x_0\,(Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \frac{1}{2}$

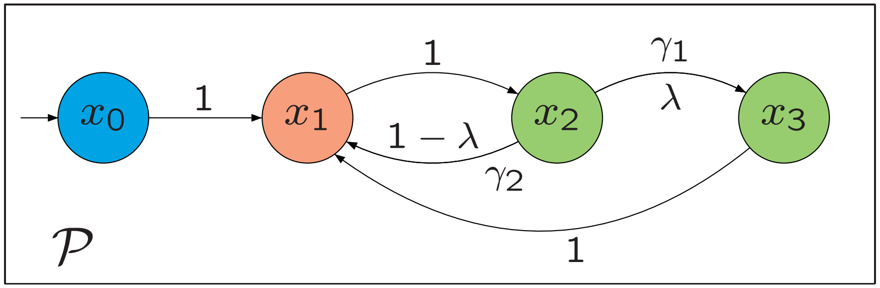
$\lambda(x_0\,(x_1\, x_2)^n) = ?$

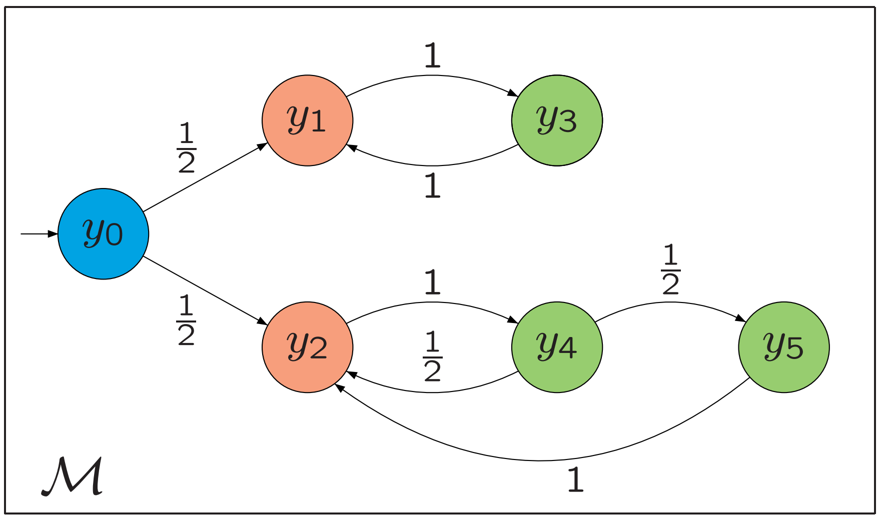
$P^{\mathcal{M}}(\bullet \mid \bullet\,(\bullet\,\bullet)^n) = ?$

$P^{\mathcal{M}}(\bullet\,(\bullet\,\bullet)^n) = ?$

- $P^{\mathcal{M}}(y_0\,(y_1\, y_3)^n) = \frac{1}{2}$

Finite-memory is not sufficient





Does $\mathcal{M} \sqsubseteq \mathcal{P}$? Yes ! but with infinite memory.

$\lambda(x_0\, x_1\, x_2) = \frac{1}{4}$

$\lambda(x_0\, (Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \frac{1}{2}$

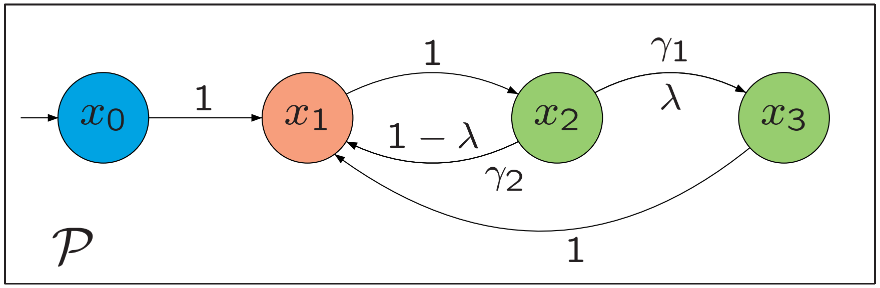
$\lambda(x_0\, (x_1\, x_2)^n) =?$

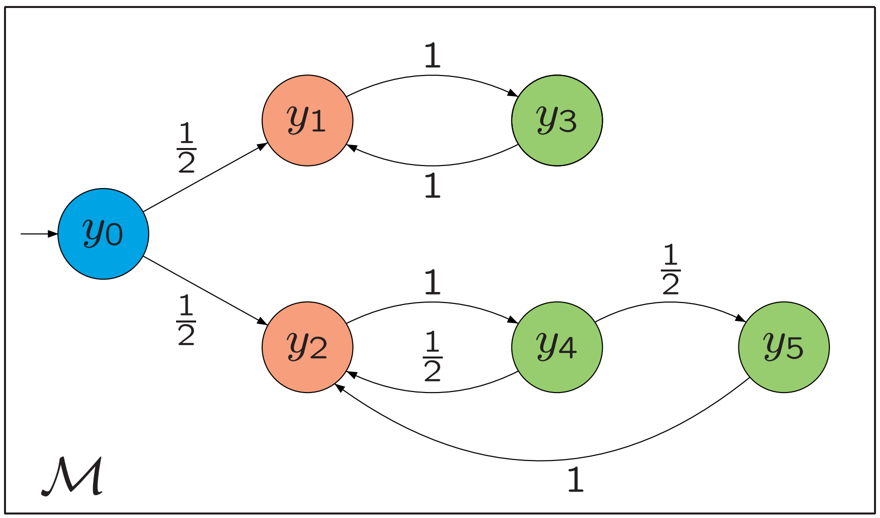
$P^{\mathcal{M}}(\bullet \mid \bullet\, (\bullet\, \bullet)^n) =?$

$P^{\mathcal{M}}(\bullet\, (\bullet\, \bullet)^n) =?$

- $P^{\mathcal{M}}(y_0\, (y_1\, y_3)^n) = \frac{1}{2}$
- $P^{\mathcal{M}}(y_0\, (y_2\, y_4)^n) = \frac{1}{2^n}$

Finite-memory is not sufficient





Does $\mathcal{M} \sqsubseteq \mathcal{P}$? Yes ! but with infinite memory.

$$\lambda(x_0\, x_1\, x_2) = \tfrac{1}{4}$$

$$\lambda(x_0\, (Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \tfrac{1}{2}$$

$$\lambda(x_0\, (x_1\, x_2)^n) = ?$$

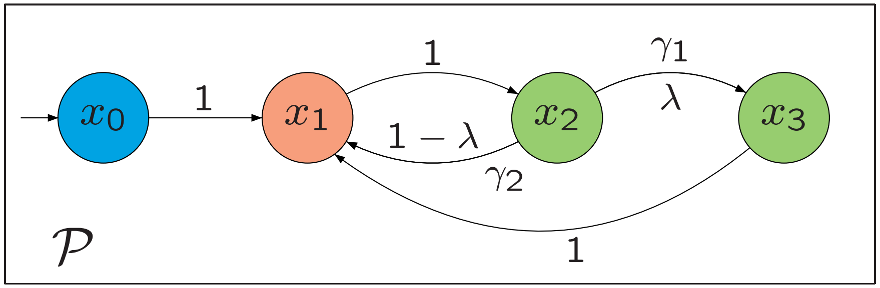
$$P^{\mathcal{M}}(\bullet \mid \bullet\, (\bullet\, \bullet)^n) = ?$$

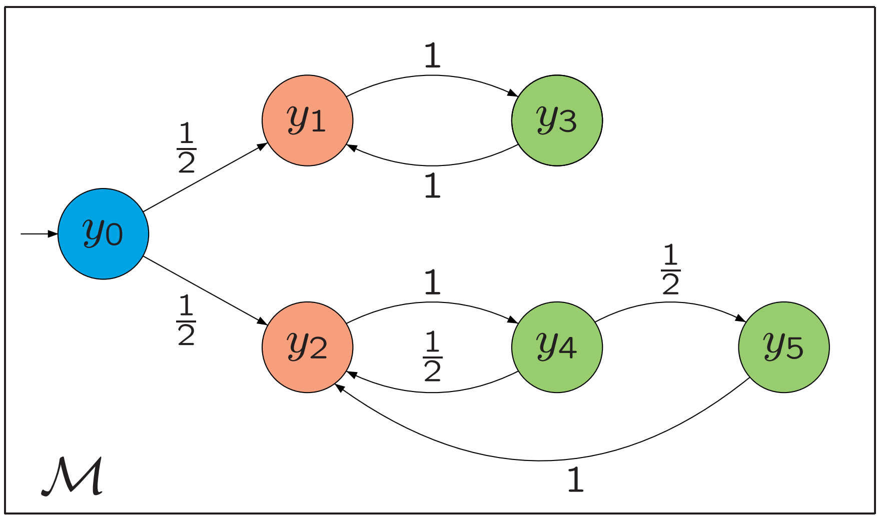
$$P^{\mathcal{M}}(\bullet\, (\bullet\, \bullet)^n) = ?$$

- $P^{\mathcal{M}}(y_0\, (y_1\, y_3)^n) = \frac{1}{2}$
- $P^{\mathcal{M}}(y_0\, (y_2\, y_4)^n) = \frac{1}{2^n}$

$$\lambda(x_0\, (x_1\, x_2)^n) = \frac{\frac{1}{2^n}\cdot\frac{1}{2}}{\frac{1}{2}+\frac{1}{2^n}} = \frac{1}{2+2^n}$$

Finite-memory is not sufficient





Does $\mathcal{M} \sqsubseteq \mathcal{P}$? Yes ! but with infinite memory.

$\lambda(x_0\, x_1\, x_2) = \frac{1}{4}$

$\lambda(x_0\, (Q^*\, x_2\, x_3\, Q^*)\, x_1\, x_2) = \frac{1}{2}$

$\lambda(x_0\, (x_1\, x_2)^n) =?$

$P^{\mathcal{M}}(\bullet \mid \bullet\, (\bullet\, \bullet)^n) =?$

$P^{\mathcal{M}}(\bullet\, (\bullet\, \bullet)^n) =?$

- $P^{\mathcal{M}}(y_0\, (y_1\, y_3)^n) = \frac{1}{2}$
- $P^{\mathcal{M}}(y_0\, (y_2\, y_4)^n) = \frac{1}{2^n}$

$$\lambda(x_0\, (x_1\, x_2)^n) = \frac{\frac{1}{2^n} \cdot \frac{1}{2}}{\frac{1}{2} + \frac{1}{2^n}} = \frac{1}{2 + 2^n}$$

infinite memory

Summary

- An algorithm for bisimulation and trace equivalence of labeled Markov chains.

- Open questions:
  - Minimization, Distance for LMC.
  - Refinement, Equivalence for LMDP.

Thank you

Questions ???

Bibliography

[Tze92] Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. SIAM J. Comput., 21(2):216–227, 1992.