Probabilistic branching bisimulation implies probabilistic weak bisimulation

(What I was doing in Eindhoven during August and September)

Matias D. Lee

Fac. de Matemática, Astronomía y Física (FaMAF)
Universidad Nacional de Córdoba

29 de Octubre del 2014
Charlas del grupo de Sistemas Confiables

Outline

Preliminaries

Why I started doing this

Our models: Probabilistic Automata

Weak transitions

Cones and fragments

Bisimulations

Probabilistic Weak bisimulation

Probabilistic branching bisimulation

$\approx_b$ is stronger than $\approx_w$

Branching bisimulation without schedulers

Logical characterization of the branching bisimilarity

The Logic

Outline

Deciding Bisimilarities on Distributions

Christian Eisentraut[1], Holger Hermanns[1], Julia Krämer[1],
Andrea Turrini[1], and Lijun Zhang[2,3,1]

[1] Saarland University – Computer Science, Saarbrücken, Germany
[2] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[3] DTU Informatics, Technical University of Denmark, Denmark

The important things of the paper:

- Definition of weak bisimulation for LTS over distributions and state based characterization of it: our definition of weak bisimulation.
- An exponential algorithm to check the relation.

Branching bisimulation congruence for probabilistic systems

Suzana Andova[a], Sonja Georgievska[a,*], Nikola Trčka[a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

- Definition of branching bisimulation (bb) for alternating models.
- If $\sim_b$ is the union of all bb relations, then $\sim_b$ is a bb relation.
- A probabilistic process language such that rooted bb is a congruence for all operator in the language.
- The logic pCTL: if two process are bb then they satisfy the same pCTL formula.
- A polinomial algorithm to check the relation.

Branching bisimulation congruence for probabilistic systems

Suzana Andova[a], Sonja Georgievska[a,*], Nikola Trčka[a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

- Definition of branching bisimulation (bb) for alternating models.
- If $\sim_b$ is the union of all bb relations, then $\sim_b$ is a bb relation.
- A probabilistic process language such that rooted bb is a congruence for all operator in the language.
- The logic pCTL: if two process are bb then they satisfy the same pCTL formula.
- A polinomial algorithm to check the relation.

Branching bisimulation congruence for probabilistic systems

Suzana Andova [a], Sonja Georgievska [a,*], Nikola Trčka [a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

- Definition of branching bisimulation (bb) for alternating models.
- If $\sim_b$ is the union of all bb relations, then $\sim_b$ is a bb relation.
- A probabilistic process language such that rooted bb is a congruence for all operator in the language.
- The logic pCTL: if two process are bb then they satisfy the same pCTL formula.
- A polinomial algorithm to check the relation.

Branching bisimulation congruence for probabilistic systems

Suzana Andova [a], Sonja Georgievska [a,*], Nikola Trčka [a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

- Definition of branching bisimulation (bb) for alternating models.
- If $\sim_b$ is the union of all bb relations, then $\sim_b$ is a bb relation.
- A probabilistic process language such that rooted bb is a congruence for all operator in the language.
- The logic pCTL: if two process are bb then they satisfy the same pCTL formula.
- A polinomial algorithm to check the relation.

Branching bisimulation congruence for probabilistic systems

Suzana Andova[a], Sonja Georgievska[a,*], Nikola Trčka[a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

- Definition of branching bisimulation (bb) for alternating models.
- If $\sim_b$ is the union of all bb relations, then $\sim_b$ is a bb relation.
- A probabilistic process language such that rooted bb is a congruence for all operator in the language.
- The logic pCTL: if two process are bb then they satisfy the same pCTL formula.
- A polinomial algorithm to check the relation.

Branching bisimulation congruence for probabilistic systems

Suzana Andova[a], Sonja Georgievska[a,*], Nikola Trčka[a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

- Definition of branching bisimulation (bb) for alternating models.
- If $\sim_b$ is the union of all bb relations, then $\sim_b$ is a bb relation.
- A probabilistic process language such that rooted bb is a congruence for all operator in the language.
- The logic pCTL: if two process are bb then they satisfy the same pCTL formula.
- A polinomial algorithm to check the relation.

Outline

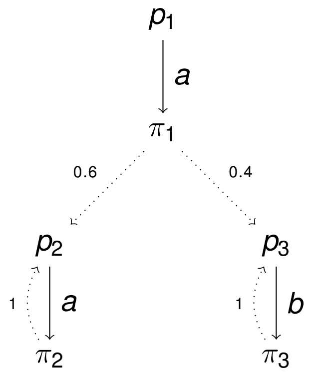
Preliminaries
- Why I started doing this
- Our models: Probabilistic Automata
- Weak transitions
- Cones and fragments

Bisimulations
- Probabilistic Weak bisimulation
- Probabilistic branching bisimulation
- $\approx_b$ is stronger than $\approx_w$
- Branching bisimulation without schedulers

Logical characterization of the branching bisimilarity
- The Logic

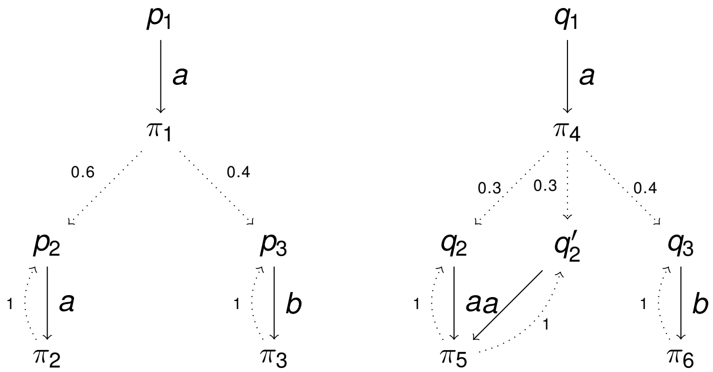
Probabilistic Automaton

Probabilistic transition systems without τ-steps



Probabilistic Automaton

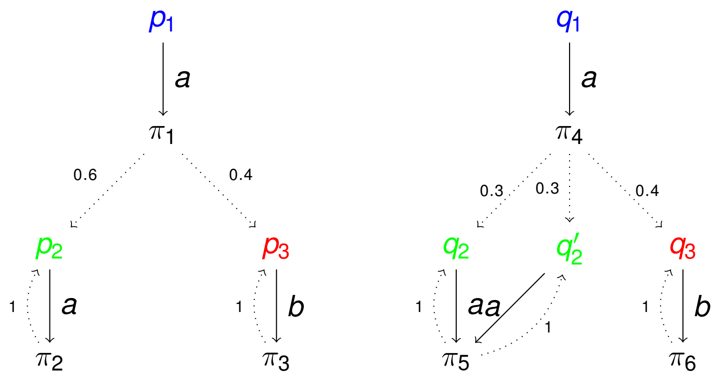
Probabilistic transition systems without τ-steps that are
bisimilar



Probabilistic Automaton

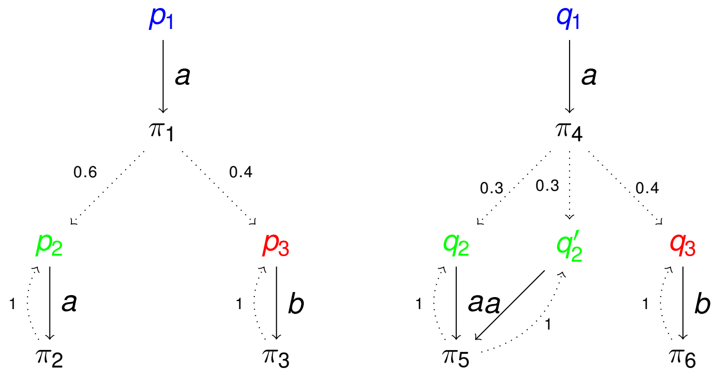
Probabilistic transition systems without τ-steps that are

bisimilar



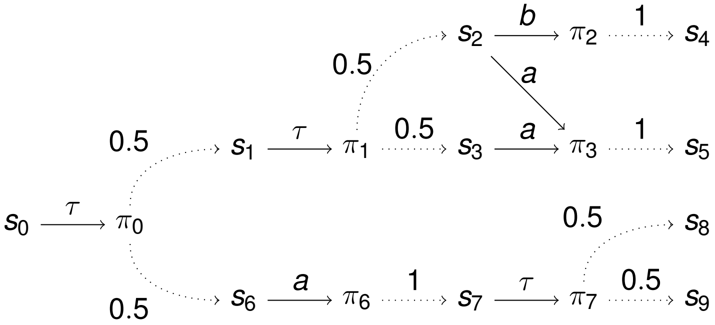
Probabilistic Automaton

Probabilistic transition systems without τ-steps that are **bisimilar**



$$\pi_1 \, \mathcal{L}(\mathcal{B}) \, \pi_4$$

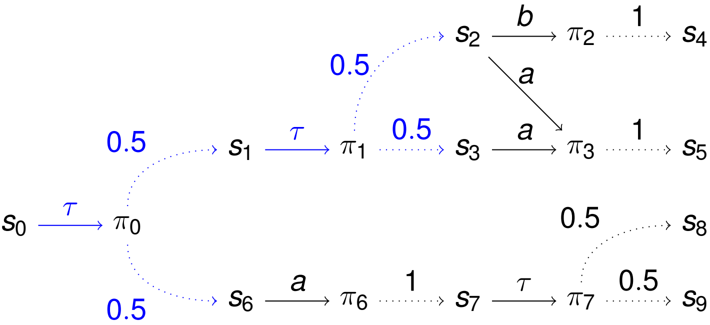
Probabilistic Automata with τ-steps



Outline

Weak transitions - Ex 1

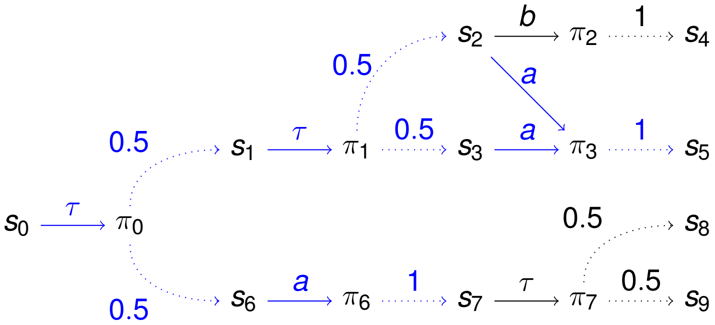
Let σ be a scheduler s.t:



$$s_0 \stackrel{\tau}{\Longrightarrow}_{\sigma} [0.25]s_2 + [0.25]s_3 + [0.25]s_6$$

Weak transitions - Ex 2

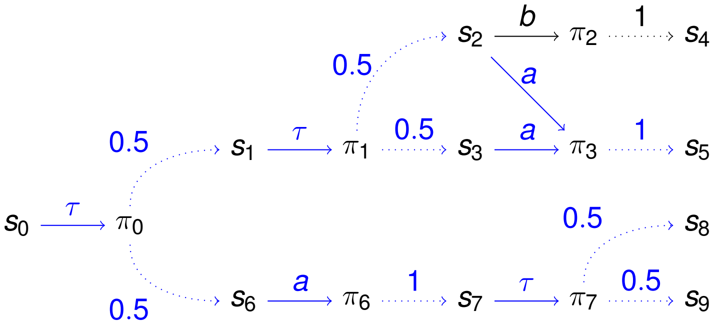
Let σ be a scheduler s.t:



$$s_0 \stackrel{a}{\Longrightarrow}_{\sigma} [0.5]s_5 + [0.5]s_7$$

Weak transitions - Ex 3

Let σ be a scheduler s.t:



$$s_0 \overset{a}{\Longrightarrow}_{\sigma} [0.5]s_5 + [0.25]s_8 + [0.25]s_9$$

Allowed weak combined transtions

$$s \xRightarrow{a \downharpoonright P}_{\sigma} \mu \text{ if and only if } s \xRightarrow{a}_{\sigma} \mu \text{ and } Im(\sigma) \subseteq P$$

Weak hyper transtions

$\mu \stackrel{a}{\Longrightarrow}_c \nu$ if and only iff

- for each $s \in Supp(\mu)$, there is σ_s s.t $s \stackrel{a}{\Longrightarrow}_{\sigma_s} \mu_s$ and

$$\sum_{s \in Supp(\mu)} \mu(s) \cdot \mu_s = \nu$$

Weak hyper transtions

$\mu \stackrel{a}{\Longrightarrow}_c \nu$ if and only iff

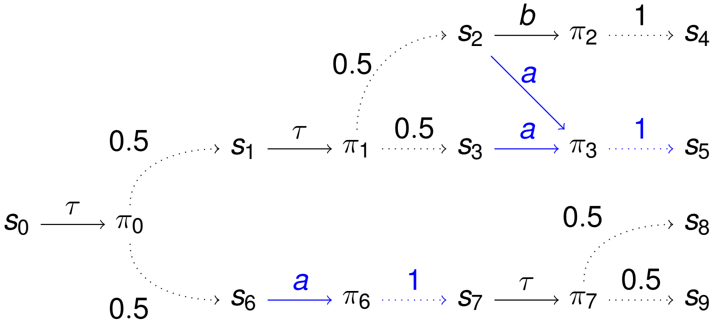
- for each $s \in Supp(\mu)$, there is σ_s s.t $s \stackrel{a}{\Longrightarrow}_{\sigma_s} \mu_s$ and

$$\sum_{s \in Supp(\mu)} \mu(s) \cdot \mu_s = \nu$$

Similar definition for $\mu \stackrel{a \downharpoonleft P}{\Longrightarrow}_c \nu$

Weak (hyper) transitions of length 1

Let π and σ s.t. $Supp(\pi) = \{s_2, s_3, s_6\}$ and

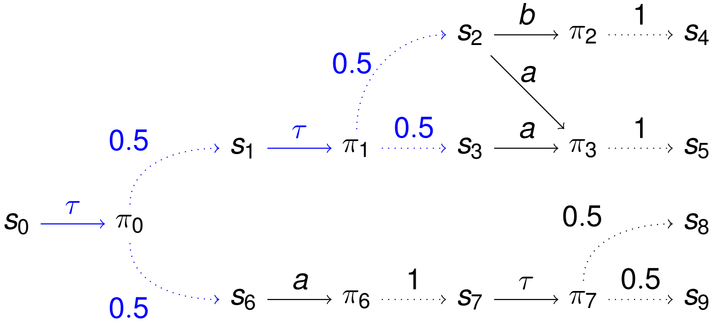


$$\pi \xrightarrow{a}_{\sigma} \pi(s_2)\pi_3 + \pi(s_3)\pi_3 + \pi(s_6)\pi_6$$

Outline

Probabilities of Cones and Fragmets

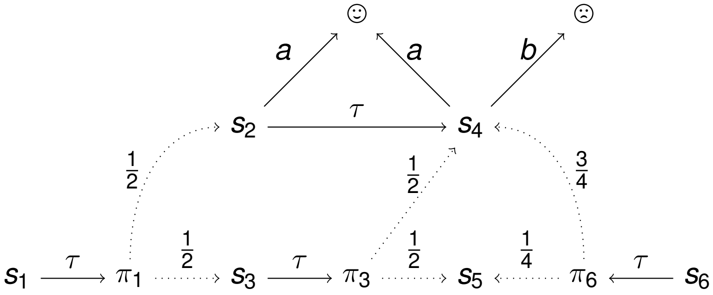
Let σ be a scheduler s.t:



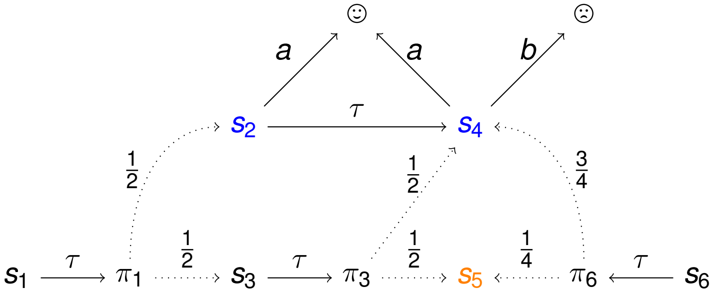
Prob. of Cones	Prob. of Fragmets
$\mu_{\sigma,s_0}(C_{s_0\tau s_6}) = 0.5$	$\mu_{\sigma,s_0}(s_0\tau s_6) = 0.5$
$\mu_{\sigma,s_0}(C_{s_0\tau s_1}) = 0.5$	$\mu_{\sigma,s_0}(s_0\tau s_1) = 0$
$\mu_{\sigma,s_0}(C_{s_0\tau s_1\tau s_3 a s_5}) = 0$	$\mu_{\sigma,s_0}(s_0\tau s_1\tau s_3 a s_5) = 0$

Outline

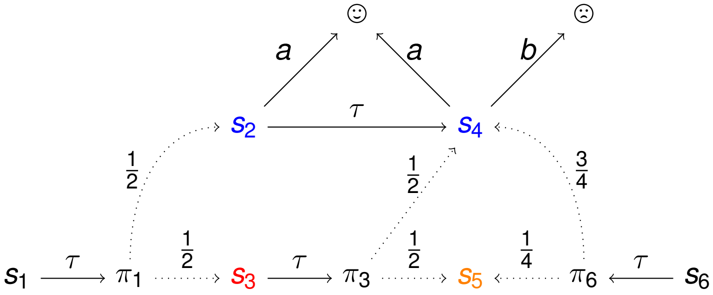
Motivation



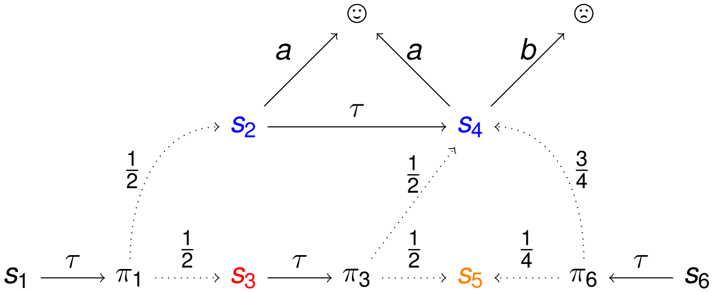
Motivation



Motivation

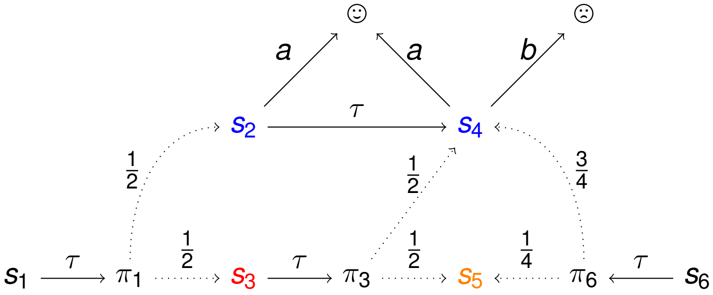


Motivation



- $s_6 \xrightarrow{\tau} \pi_6$ then $s_1 \overset{\tau}{\Longrightarrow}_c [1/2]s_2 + [1/4]s_4 + [1/4]s_5$

Motivation

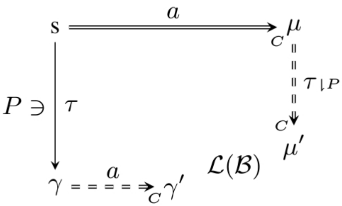


- $s_6 \xrightarrow{\tau} \pi_6$ then $s_1 \stackrel{\tau}{\Longrightarrow}_c [1/2]s_2 + [1/4]s_4 + [1/4]s_5$
- $s_1 \xrightarrow{\tau} \pi_1$ then ...

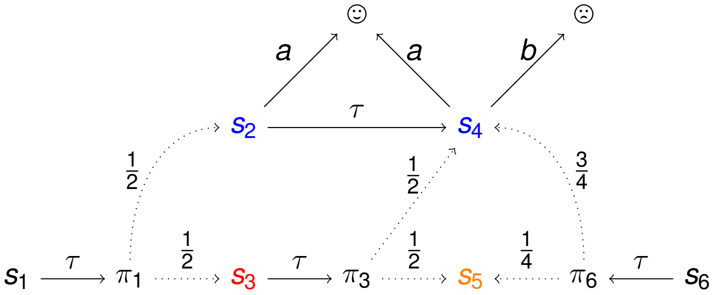
Preserving Transitions

Definition

Let $\mathcal{B}$ be an equivalence relation on S and $P \subseteq D$ a set of τ-transition. The set P is *preserving with respect to* $\mathcal{B}$ if for all $s \xrightarrow{\tau} \gamma \in P$, whenever $s \overset{a}{\Longrightarrow}_c \mu$ then there exist μ' and γ' such that $\mu \overset{\tau\downharpoonright P}{\Longrightarrow}_c \mu', \gamma \overset{a}{\Longrightarrow}_c \gamma'$ and $\mu' \, \mathcal{L}(\mathcal{B}) \, \gamma'$.



Probabilistic weak bisimulation



- $P = \{ s_1 \xrightarrow{\tau} \pi_1, s_3 \xrightarrow{\tau} \pi_3, s_2 \xrightarrow{\tau} \delta_{s_4} \}$

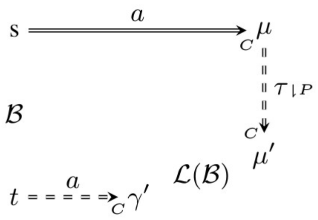
Probabilistic weak bisimulation

Definition

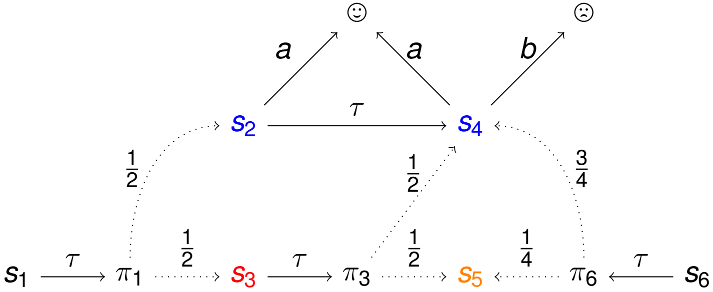
An equivalence relation $\mathcal{B}$ on S is called a *probabilistic weak bisimulation*, if there is a set $P \subseteq D$ that is preserving with respect to $\mathcal{B}$ and whenever $s \, \mathcal{B} \, t$,

- if $s \overset{a}{\Longrightarrow}_c \mu$ for some μ, then $t \overset{a}{\Longrightarrow}_c \nu$ for some ν and there exists μ' such that $\mu \overset{\tau\downharpoonright P}{\Longrightarrow} \mu'$ and $\mu' \, \mathcal{L}(\mathcal{B}) \, \nu$.

We write $s \approx_w t$ if there is a probabilistic weak bisimulation relating s and t.



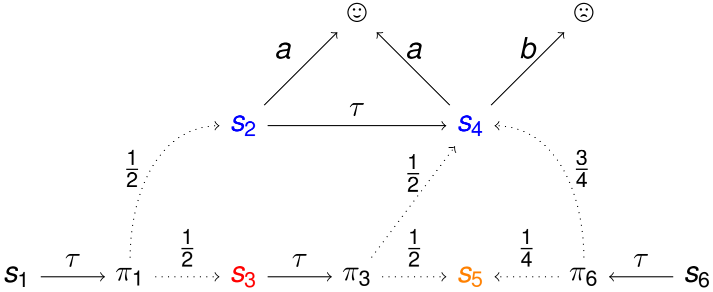
Probabilistic weak bisimulation



- $P = \{s_1 \xrightarrow{\tau} \pi_1, s_3 \xrightarrow{\tau} \pi_3, s_2 \xrightarrow{\tau} \delta_{s_4}\}$
- $s_1 \overset{\tau}{\Longrightarrow}_c \pi_1$ then $s_6 \overset{a}{\Longrightarrow}_c \pi_6$ and

$$\pi_1 \overset{\tau \downarrow P}{\Longrightarrow} [1/2]s_2 + [1/4]s_4 + [1/4]s_5 \approx_w \pi_6$$

Probabilistic weak bisimulation

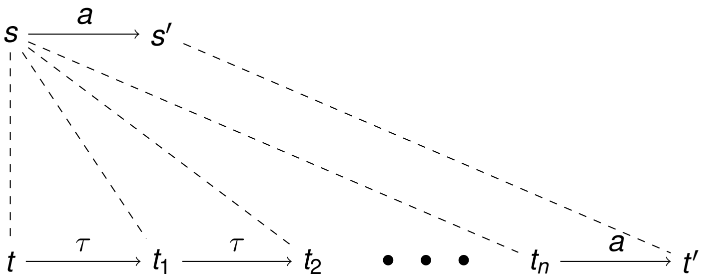


- $P = \{s_1 \xrightarrow{\tau} \pi_1, s_3 \xrightarrow{\tau} \pi_3, s_2 \xrightarrow{\tau} \delta_{s_4}\}$
- $s_1 \stackrel{\tau}{\Longrightarrow}_c \pi_1$ then $s_6 \stackrel{a}{\Longrightarrow}_c \pi_6$ and

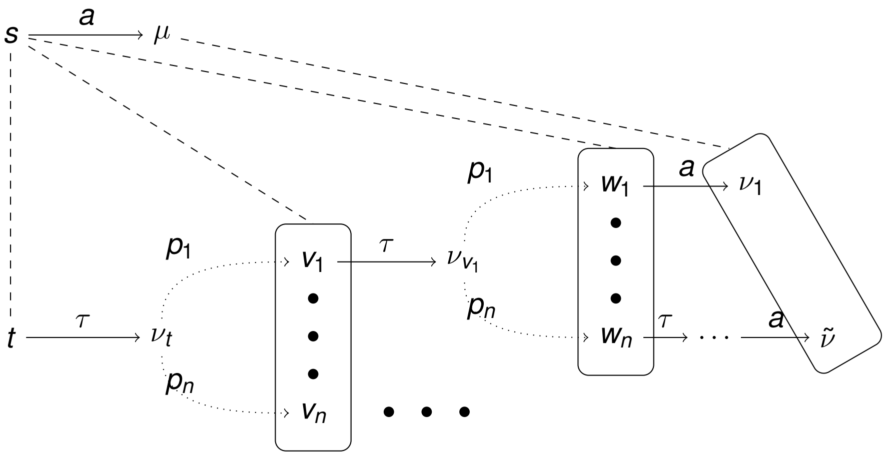
$$\pi_1 \stackrel{\tau \downharpoonright P}{\Longrightarrow} [1/2]s_2 + [1/4]s_4 + [1/4]s_5 \approx_w \pi_6$$

Outline

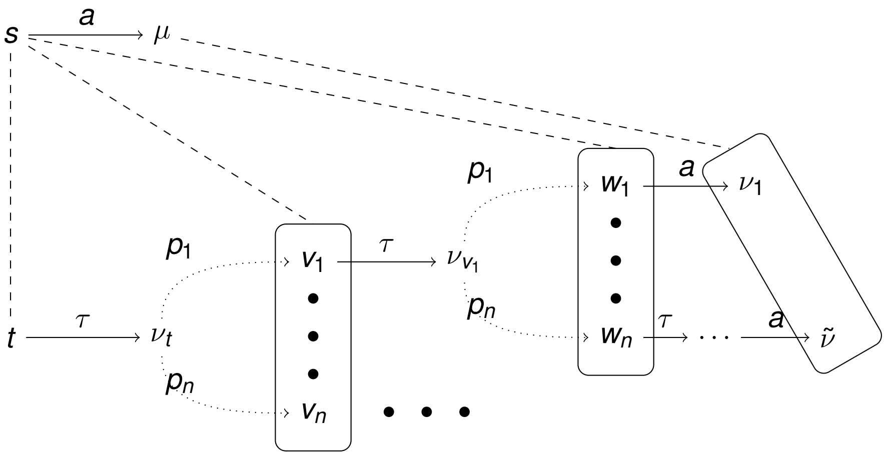
Branching bisimulation



Branching bisimulation for PA



Branching bisimulation for PA



Given a equiv. rel. $\mathcal{B}$, $t \xrightarrow{\tau} \nu$ is *branching preserving* iff $\delta_t \, \mathcal{L}(\mathcal{B}) \, \nu$

Probabilistic branching bisimulation

Definition

An equivalence relation $\mathcal{B}$ on S is called *probabilistic branching bisimulation* if there is a set $P \subseteq D(\tau)$ that is branching preserving w.r.t. $\mathcal{B}$ and whenever $s \, \mathcal{B} \, t$, if $s \xrightarrow{a} \mu$ then either

1. $a = \tau$ and $s \xrightarrow{a} \mu \in P$
2. $t \xRightarrow{\tau \downharpoonright P}_c \tilde{\nu} \xrightarrow{a}_c \nu$ and $\mu \, \mathcal{L}(\mathcal{B}) \, \nu$.

We write $s \approx_b t$ is there exists a probabilistic branching bisimulation relating s and t.

Outline

$\approx_b$ is stronger than $\approx_w$

Proof sketch: take a branching bisimulation $\mathcal{B}$ and prove that it is also a weak bisimulation.

Things that we need to prove the results

1. if $s \stackrel{a}{\Longrightarrow}_c \mu$ then there is ν and ν' such that:

$$s \stackrel{\tau}{\Longrightarrow}_c \nu \stackrel{a}{\longrightarrow}_c \nu' \stackrel{\tau}{\Longrightarrow}_c \mu$$

Things that we need to prove the results

1. if $s \stackrel{a}{\Longrightarrow}_c \mu$ then there is ν and ν' such that:

$$s \stackrel{\tau}{\Longrightarrow}_c \nu \stackrel{a}{\longrightarrow}_c \nu' \stackrel{\tau}{\Longrightarrow}_c \mu$$

2. if $\mu \stackrel{\tau}{\Longrightarrow}_c \mu'$ then there are $\mu = \mu_0, \mu_1, \ldots$ such that

$$\mu_0 \stackrel{\tau}{\longrightarrow}_c \mu_1 \stackrel{\tau}{\longrightarrow}_c \mu_2 \ldots \text{ and } \lim_{n\to\infty} \mu_n = \mu'$$

Things that we need to prove the results

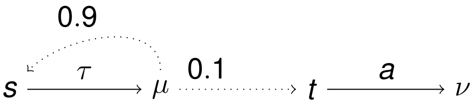
1. if $s \stackrel{a}{\Longrightarrow}_c \mu$ then there is ν and ν' such that:

$$s \stackrel{\tau}{\Longrightarrow}_c \nu \stackrel{a}{\longrightarrow}_c \nu' \stackrel{\tau}{\Longrightarrow}_c \mu$$

2. if $\mu \stackrel{\tau}{\Longrightarrow}_c \mu'$ then there are $\mu = \mu_0, \mu_1, \ldots$ such that

$$\mu_0 \stackrel{\tau}{\longrightarrow}_c \mu_1 \stackrel{\tau}{\longrightarrow}_c \mu_2 \ldots \text{ and } \lim_{n \to \infty} \mu_n = \mu'$$

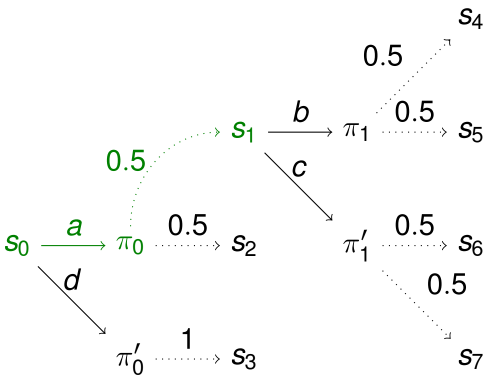
Example: $s \stackrel{\tau}{\Longrightarrow}_c \delta_t$ in



Definitions that we have introduced

- Conditional probabilities
- Convex combination of schedulers

Conditional Probabilities



What is the probability of executing $s_1 \xrightarrow{b} \pi_1$ and reach s_4 with the schedule σ given that s_1 was reached?

Conditional Probabilities

Definition

Let σ be a scheduler for a PA $\mathcal{A}$ and $\tilde{\alpha}$ be a trace.The *conditional scheduler* σ *based on* $\tilde{\alpha}$, notation $\sigma_{\tilde{\alpha}}$, is defined by

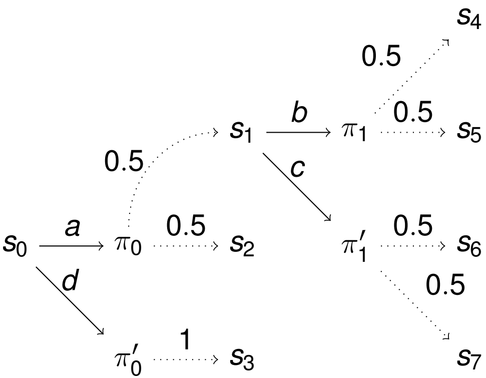
$$\sigma_{\tilde{\alpha}}(\alpha) = \begin{cases} \sigma(\tilde{\alpha}\alpha) & \text{if } \mathit{lst}(\tilde{\alpha}) = \mathit{fst}(\alpha) \\ \delta_{\perp} & \text{otherwise} \end{cases}$$

Definition

Let σ be a scheduler for a PA $\mathcal{A}$ and s, s' be two states s.t. there is $\tilde{\alpha}$ with $\mathit{lst}(\tilde{\alpha}) = s'$ and $\mu_{\sigma,s}(C_{\tilde{\alpha}}) > 0$. The *probability of the execution fragment* α *based on* σ *starting at* s *given that* s' *is reached* is defined by:

$$\mu_{\sigma,s,s'}(\alpha) = \sum_{\mathit{lst}(\tilde{\alpha})=s'} \frac{\mu_{\sigma,s}(C_{\tilde{\alpha}})}{\sum_{\mathit{lst}(\alpha')=s'} \mu_{\sigma,s}(C_{\alpha'})} \cdot \mu_{\sigma_{\tilde{\alpha}},s'}(\alpha)$$

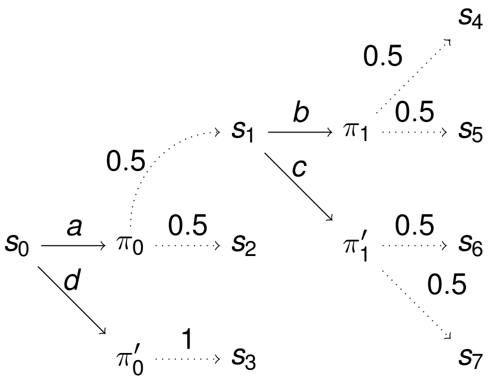
Convex combination of schedulers



$$\sigma_1(s_0) = [1]s_0 \xrightarrow{a} \pi_0, \quad \sigma_1(s_0 a s_1) = [0.5]s_1 \xrightarrow{b} \pi_1 + [0.5]s_1 \xrightarrow{c} \pi_1'$$

$$\pi_{\sigma_1,s_0}(t) = \mu_{\sigma_1,s_0}(\{\alpha \mid lst(\alpha) = t\})$$

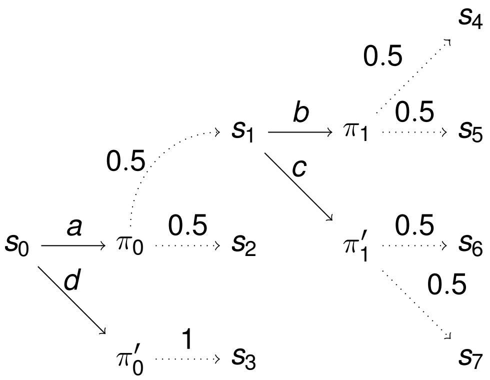
Convex combination of schedulers



$$\sigma_2(s_0) = [0.5]s_0 \xrightarrow{a} \pi_0 + [0.5]s_0 \xrightarrow{d} \pi_0', \quad \sigma_2(s_0 a s_1) = [1]s_1 \xrightarrow{b} \pi_1$$

$$\pi_{\sigma_2,s_0}(t) = \mu_{\sigma_2,s_0}(\{\alpha \mid lst(\alpha) = t\})$$

Convex combination of schedulers



$$\mu_{\{\sigma_1,\sigma_2\},\{\sigma_1\to p,\sigma_2\to(1-p)\},s_0} = p\cdot\mu_{\sigma_1,s_0} + (1-p)\cdot\mu_{\sigma_2,s_0}$$

Convex combination of schedulers

Definition

Let $\mathcal{S}$ be a set of scheduler for a PA $\mathcal{A}$ with weight function w. The *convex combination of the probabilities of an execution fragment* α is defined by

$$\mu_{\mathcal{S},w,s}(\alpha) = \sum_{\sigma \in \mathcal{S}} w(\sigma) \cdot \mu_{\sigma,s}(\alpha)$$

$\square$

Lemma

Let $\mathcal{S}$ be a set of schedulers with weight function w and s_0 be a state. Let $\tilde{\sigma}$ be a scheduler defined by

$$\tilde{\sigma}(\alpha)(s \xrightarrow{a} \mu) = \begin{cases} \sum_{\sigma \in \mathcal{S}} \frac{w(\sigma)\mu_{\sigma,s_0}(C_\alpha)}{\sum_{\hat{\sigma} \in \mathcal{S}} w(\hat{\sigma})\mu_{\hat{\sigma},s_0}(C_\alpha)} \sigma(\alpha)(s \xrightarrow{a} \mu) & \textit{if den.} \neq 0 \\ 0 & \textit{otherwise} \end{cases}$$

then $\mu_{\mathcal{S},w,s_0}(\alpha) = \mu_{\tilde{\sigma},s_0}(\alpha)$.

Auxiliary results and the main one

Lemma

Let $\mathcal{A}$ be a PA, s_0 be a state of $\mathcal{A}$, μ be a distribution and a be an action s.t. $a \neq \tau$. If $a \overset{a}{\Longrightarrow}_c \mu$ then there are ν, ν' s.t. $s_0 \overset{\tau}{\Longrightarrow}_c \nu \overset{a}{\longrightarrow}_c \nu' \overset{\tau}{\Longrightarrow}_c \mu$.

Lemma

Let $\mathcal{A}$ be a PA, s_0 be a state of $\mathcal{A}$ and σ be a scheduler. If $s_0 \overset{\tau}{\Longrightarrow}_{\sigma_n} \mu_n$ and $s_0 \overset{\tau}{\Longrightarrow}_{\sigma_{n+1}} \mu_{n+1}$ then $\mu_n \overset{\tau}{\longrightarrow}_c \mu_{n+1}$.

Theorem

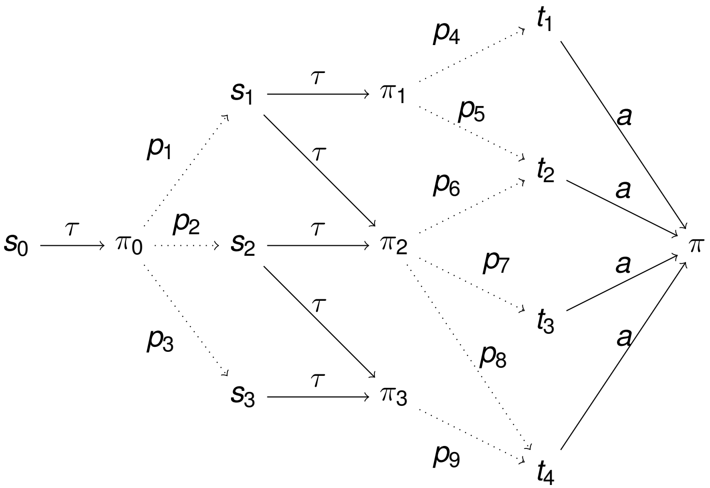
$\approx_b$ is stronger than $\approx_w$.

Proof Sketch

- $s \approx_b t$ and $s \stackrel{a}{\Longrightarrow}_\sigma \mu$.
- Lemmas implies $s \stackrel{\tau}{\longrightarrow}_c \dots \stackrel{\tau}{\longrightarrow}_c \nu \stackrel{a}{\longrightarrow}_c \nu' \stackrel{\tau}{\longrightarrow}_c \dots \stackrel{\tau}{\longrightarrow} \mu$
- Defintion of $\approx_b$ allows to show $t \stackrel{a \downharpoonright P}{\Longrightarrow}_c \nu$ with P a set of (branching) preserving transitions and $\mu \approx_b \nu$, then
- $s \approx_w t$

Outline

The intuition



For all scheduler σ, $s_0 \overset{a}{\Longrightarrow}_\sigma \pi$

Concrete excution

A sequence $s_0 a_0 \mu_0 s_1 a_2 \mu_2 s_3 \ldots \mu_{n-1} s_n a_n \mu_n$ is a *concrete execution* of a PA $\mathcal{A}$ if $s_i \xrightarrow{a_i} \mu_i$ for $0 \leq i \leq n$ and $s_{j+1} \in Supp[\mu_j]$ for $0 < j < n$.

Branching bisimulation without schedulers

Definition

An equivalence relation $\mathcal{B} \subseteq S \times S$ is a branching bisimulation without schedulers for a PA $\mathcal{A} = (S, s_0, \Sigma \cup \{\tau\}, D)$ iff for every $s \, \mathcal{L}(\mathcal{B}) \, s'$, if $s \xrightarrow{a_\tau} \mu$ then

1. $a_\tau = \tau$ and $\delta_s \, \mathcal{L}(B) \, \mu$ or
2. $a_\tau \in \Sigma \cup \{\tau\}$ and there is a concrete execution $s_0 \tau \mu_1 s_1 \tau \mu_3 s_3 \ldots \mu_n s_n a_\tau \mu'$ s.t. $s' = s_0$, $s \, \mathcal{B} \, s_i$ for $0 \leq i \leq n$, $\mu_j \, \mathcal{L}(\mathcal{B}) \, \delta_s$ for $1 \leq j \leq n$ and $\mu \, \mathcal{L}(\mathcal{B}) \, \mu'$.

We write $s \sim_b s'$ if there is a branching bisimulation without schedulers that relates s and s'.

Outline

Syntax

Definition

The syntax of pL_{U} is defined as follows:

$$\psi := \top \mid \psi \wedge \psi \mid \neg\psi \mid \psi\langle a\rangle\Delta \mid \psi\langle\tau\rangle\Delta$$
$$\Delta := [\psi]_{\geq p} \mid \Delta \wedge \Delta \mid \neg\Delta$$

where ψ-formula and Δ-formula are called *state formula* and *distribution formula*, respectively.

Semantic

Definition

Let $\mathcal{A} = (S, s_0, \Sigma \cup \{\tau\}, D)$ be a PA and let $s \xrightarrow{a} \mu \in D$. The satisfaction relation $\models$ for a pL_{U} formula is defined by:

$s \models$	$\top$	always
$s \models$	$\psi \wedge \psi'$	if $s \models \psi$ and $s \models \psi'$
$s \models$	$\neg\psi$	if $s \not\models \psi$
$s \models$	$\psi\langle a\rangle\Delta$	if there is concrete execution $s\tau\mu_1 s_1\tau\mu_2 \ldots \tau\mu_n s_n a\mu$ s.t. $a \in A \cup \{\tau\}$, $s \models \psi$, $s_i \models \psi$, $\mu_i \models [\psi]_{\geq 1}$ for $1 \leq i \leq n$, $n \geq 0$ and $\mu \models \Delta$
$s \models$	$\psi\langle\tau\rangle\Delta$	$s \models \psi$ and $\delta_s \models \Delta$
$\mu \models$	$[\psi]_{\geq p}$	if $\mu(\{s \mid s \models \psi\}) \geq p$
$\mu \models$	$\Delta \wedge \Delta'$	if $\mu \models \Delta$ and $\mu \models \Delta'$
$\mu \models$	$\neg\Delta$	if $\mu \not\models \Delta$

The logic characterize branching bisimulation

Theorem

Let $\mathcal{A}$ be a PA with bounded nondeterminism, then

- $s \approx_b s'$ *if and only if* $s \approx_{pL_U} s'$.
- $\mu\, \mathcal{L}(\approx_b)\, \mu'$ *if and only if* $\mu \approx_{pL_U} \mu'$

Questions?