Modeling and Verification of Probabilistic Systems

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/teaching/ws-1516/movep15/

November 17, 2015

Overview

Labeled transition system

Transition system

A *(labeled) transition system* TS is a structure $(S, Act, \longrightarrow, I_0, AP, L)$ where

- S is a (possibly infinitely countable) set of states.
- Act is a (possibly infinitely countable) set of actions.
- $\longrightarrow \subseteq S \times Act \times S$ is a transition relation.
- $I_0 \subseteq S$ the set of initial states.
- AP is a set of atomic propositions.
- $L : S \rightarrow 2^{AP}$ is the labeling function.

Labeled transition system

Transition system

A *(labeled) transition system* TS is a structure $(S, Act, \longrightarrow, I_0, AP, L)$ where

- S is a (possibly infinitely countable) set of states.
- Act is a (possibly infinitely countable) set of actions.
- $\longrightarrow \subseteq S \times Act \times S$ is a transition relation.
- $I_0 \subseteq S$ the set of initial states.
- AP is a set of atomic propositions.
- $L : S \rightarrow 2^{AP}$ is the labeling function.

Notation

We write $s \xrightarrow{\alpha} s'$ instead of $(s, \alpha, s') \in \longrightarrow$.

Strong bisimulation

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$.

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$. Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$.
Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$. Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$
2. if $s \xrightarrow{\alpha} s'$ then there exists $t' \in S$ such that $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$.
Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$
2. if $s \xrightarrow{\alpha} s'$ then there exists $t' \in S$ such that $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$
3. if $t \xrightarrow{\alpha} t'$ then there exists $s' \in S$ such that $s \xrightarrow{\alpha} s'$ and $(s', t') \in R$

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$.
Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$
2. if $s \xrightarrow{\alpha} s'$ then there exists $t' \in S$ such that $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$
3. if $t \xrightarrow{\alpha} t'$ then there exists $s' \in S$ such that $s \xrightarrow{\alpha} s'$ and $(s', t') \in R$

Strong bisimilarity

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $s, t \in S$.

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$. Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$
2. if $s \xrightarrow{\alpha} s'$ then there exists $t' \in S$ such that $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$
3. if $t \xrightarrow{\alpha} t'$ then there exists $s' \in S$ such that $s \xrightarrow{\alpha} s'$ and $(s', t') \in R$

Strong bisimilarity

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $s, t \in S$. Then: s is *strongly bisimilar* to t, notation $s \sim t$, if there *exists* a strong bisimulation R such that $(s, t) \in R$.

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$.
Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$
2. if $s \xrightarrow{\alpha} s'$ then there exists $t' \in S$ such that $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$
3. if $t \xrightarrow{\alpha} t'$ then there exists $s' \in S$ such that $s \xrightarrow{\alpha} s'$ and $(s', t') \in R$

Strong bisimilarity

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $s, t \in S$.
Then: s is *strongly bisimilar* to t, notation $s \sim t$, if there *exists* a strong bisimulation R such that $(s, t) \in R$.

Remarks

Not every bisimulation relation is transitive.

Strong bisimulation

Strong bisimulation relation [Milner, 1980 & Park, 1981]

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $R \subseteq S \times S$.
Then R is a *strong bisimulation* on TS whenever for all $(s, t) \in R$:

1. $L(s) = L(t)$
2. if $s \xrightarrow{\alpha} s'$ then there exists $t' \in S$ such that $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$
3. if $t \xrightarrow{\alpha} t'$ then there exists $s' \in S$ such that $s \xrightarrow{\alpha} s'$ and $(s', t') \in R$

Strong bisimilarity

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ be a transition system and $s, t \in S$.
Then: s is *strongly bisimilar* to t, notation $s \sim t$, if there *exists* a strong bisimulation R such that $(s, t) \in R$.

Remarks

Not every bisimulation relation is transitive. But: $\sim$ is an equivalence.

Strong bisimulation

Pictorial representation

$$\begin{array}{ccc} s & \xrightarrow{\alpha} & s' \\ R & & \\ t & & \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} s & \xrightarrow{\alpha} & s' \\ R & & R \\ t & \xrightarrow{\alpha} & t' \end{array}$$

Strong bisimulation

Pictorial representation

s	$\xrightarrow{\alpha}$	s'		s	$\xrightarrow{\alpha}$	s'
R			can be completed to	R		R
t				t	$\xrightarrow{\alpha}$	t'

and

s				s	$\xrightarrow{\alpha}$	s'
R			can be completed to	R		R
t	$\xrightarrow{\alpha}$	t'		t	$\xrightarrow{\alpha}$	t'

Strongly bisimilar transition systems

Strongly bisimilar transition systems

Bisimilar transition systems

Let TS_1, TS_2 be transition systems over the same set of atomic propositions with initial states $I_{0,1}$ and $I_{0,2}$, respectively.

Strongly bisimilar transition systems

Bisimilar transition systems

Let TS_1, TS_2 be transition systems over the same set of atomic propositions with initial states $I_{0,1}$ and $I_{0,2}$, respectively.

Consider the transition system $TS = TS_1 \uplus TS_2$ that results from the disjoint union of TS_1 and TS_2.

Strongly bisimilar transition systems

Bisimilar transition systems

Let TS_1, TS_2 be transition systems over the same set of atomic propositions with initial states $I_{0,1}$ and $I_{0,2}$, respectively.

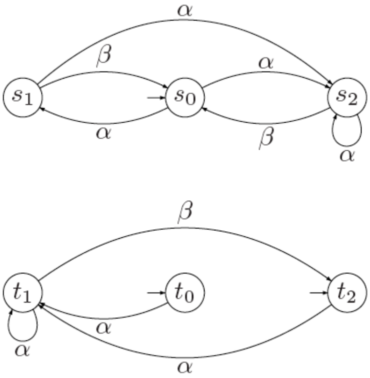
Consider the transition system $TS = TS_1 \uplus TS_2$ that results from the disjoint union of TS_1 and TS_2.

Then: TS_1 and TS_2 are called strongly bisimilar if there exists a strong bisimulation R on $S_1 \uplus S_2$ such that:

1. $\forall s \in I_{0,1}. \exists t \in I_{0,2}. (s, t) \in R$, and
2. $\forall t \in I_{0,2}. \exists s \in I_{0,1}. (s, t) \in R$.

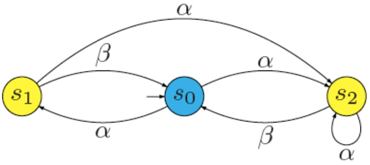
Example (1)

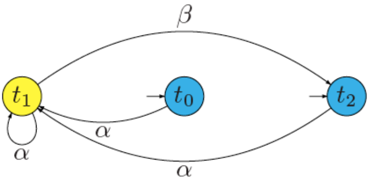
Are these transition systems strongly bisimilar? (No propositions.)



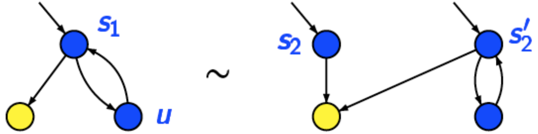
Example (2)

Yes, they are!



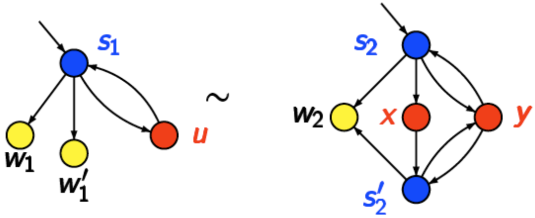


Correct or wrong?



wrong

$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow$ *blue* (thus $s_1 \not\sim s_2$)



correct

bisimulation:

$\{(w_1, w_2), (w_1', w_2), (s_1, s_2), (s_1, s_2'), (u, x), (u, y)\}$

Quotient LTS under $\sim$

Quotient transition system

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \;=\; (S', Act, \longrightarrow', I'_0, AP, L'), \qquad \text{the } \textit{quotient} \text{ of } TS \text{ under } \sim$$

where

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \; = \; (S', Act, \longrightarrow', I_0', AP, L'), \qquad \text{the } \textit{quotient} \text{ of } TS \text{ under } \sim$$

where

- $S' = S/\sim \; = \; \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim \; = \; \{ s' \in S \mid s \sim s' \}$

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \; = \; (S', Act, \longrightarrow', I'_0, AP, L'), \qquad \text{the } quotient \text{ of } TS \text{ under } \sim$$

where

- $S' = S/\sim \; = \; \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim \; = \; \{ s' \in S \mid s \sim s' \}$
- $\longrightarrow'$ is defined by: $$\frac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\alpha}{}' [s']_\sim}$$

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \; = \; (S', Act, \longrightarrow', I_0', AP, L'), \qquad \text{the } \textit{quotient} \text{ of } TS \text{ under } \sim$$

where

- $S' = S/\sim \; = \; \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim \; = \; \{ s' \in S \mid s \sim s' \}$
- $\longrightarrow'$ is defined by: $\dfrac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\alpha}' [s']_\sim}$
- $I_0' = \{ [s_0]_\sim \mid s_0 \in I_0 \}$, the equivalence class of the initial states in TS

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \; = \; (S', Act, \longrightarrow', I'_0, AP, L'), \qquad \text{the } quotient \text{ of } TS \text{ under } \sim$$

where

- $S' = S/\sim \; = \; \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim \; = \; \{ s' \in S \mid s \sim s' \}$
- $\longrightarrow'$ is defined by: $\dfrac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\alpha}{}' [s']_\sim}$
- $I'_0 = \{ [s_0]_\sim \mid s_0 \in I_0 \}$, the equivalence class of the initial states in TS
- $L'([s]_\sim) = L(s)$.

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \; = \; (S', Act, \longrightarrow', I_0', AP, L'), \qquad \text{the } \textit{quotient} \text{ of } TS \text{ under } \sim$$

where

- $S' = S/\sim \; = \; \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim \; = \; \{ s' \in S \mid s \sim s' \}$
- $\longrightarrow'$ is defined by: $\dfrac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\alpha}{}' [s']_\sim}$
- $I_0' = \{ [s_0]_\sim \mid s_0 \in I_0 \}$, the equivalence class of the initial states in TS
- $L'([s]_\sim) = L(s)$.

Remarks

L' is well-defined as all states in $[s]_\sim$ are equally labeled.

Quotient LTS under $\sim$

Quotient transition system

For $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and strong bisimilarity $\sim \subseteq S \times S$ let

$$TS/\sim \; = \; (S', Act, \longrightarrow', I_0', AP, L'), \qquad \text{the } \textit{quotient} \text{ of } TS \text{ under } \sim$$

where

- $S' = S/\sim \; = \; \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim \; = \; \{ s' \in S \mid s \sim s' \}$
- $\longrightarrow'$ is defined by: $\dfrac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\alpha}{}' [s']_\sim}$
- $I_0' = \{ [s_0]_\sim \mid s_0 \in I_0 \}$, the equivalence class of the initial states in TS
- $L'([s]_\sim) = L(s)$.

Remarks

L' is well-defined as all states in $[s]_\sim$ are equally labeled. Note that if $s \xrightarrow{\alpha} s'$, then for all $t \sim s$ we have $t \xrightarrow{\alpha} t'$ with $s' \sim t'$.

Quotient transition system

Quotient transition system

For any transition system TS it holds: $TS \sim TS/\sim$.

Quotient transition system

For any transition system TS it holds: $TS \sim TS/\sim$.

Proof:

The binary relation:

$$R = \{ (s, [s]_\sim) \mid s \in S \}$$

is a strong bisimulation on the disjoint union $TS \uplus TS/\sim$.

Strong bisimulation revisited

Strong bisimulation revisited

Auxiliary predicate

Let $P : S \times Act \times 2^S \to \{ 0, 1 \}$ be a predicate such that for $S' \subseteq S$:

$$P(s, \alpha, S') = \begin{cases} 1 & \text{if } \exists s' \in S'.\ s \xrightarrow{\alpha} s' \\ 0 & \text{otherwise.} \end{cases}$$

Strong bisimulation revisited

Auxiliary predicate

Let $P : S \times Act \times 2^S \rightarrow \{0, 1\}$ be a predicate such that for $S' \subseteq S$:

$$P(s, \alpha, S') = \begin{cases} 1 & \text{if } \exists s' \in S'.\ s \xrightarrow{\alpha} s' \\ 0 & \text{otherwise.} \end{cases}$$

Alternative definition of strong bisimulation

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and R an *equivalence relation* on S.
Then: R is a *strong bisimulation* on S if for $(s, t) \in R$:

1. $L(s) = L(t)$, and

Strong bisimulation revisited

Auxiliary predicate

Let $P : S \times Act \times 2^S \to \{ 0, 1 \}$ be a predicate such that for $S' \subseteq S$:

$$P(s, \alpha, S') = \begin{cases} 1 & \text{if } \exists s' \in S'.\ s \xrightarrow{\alpha} s' \\ 0 & \text{otherwise.} \end{cases}$$

Alternative definition of strong bisimulation

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and R an *equivalence relation* on S.
Then: R is a *strong bisimulation* on S if for $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $P(s, \alpha, C) \; = \; P(t, \alpha, C)$ for all C in S/R and $\alpha \in Act$.

$s \sim' t$, if there *exists* a strong bisimulation R such that $(s, t) \in R$.

Strong bisimulation revisited

Auxiliary predicate

Let $P : S \times Act \times 2^S \rightarrow \{ 0, 1 \}$ be a predicate such that for $S' \subseteq S$:

$$P(s, \alpha, S') = \begin{cases} 1 & \text{if } \exists s' \in S'.\ s \xrightarrow{\alpha} s' \\ 0 & \text{otherwise.} \end{cases}$$

Alternative definition of strong bisimulation

Let $TS = (S, Act, \longrightarrow, I_0, AP, L)$ and R an *equivalence relation* on S.
Then: R is a *strong bisimulation* on S if for $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $P(s, \alpha, C) \ = \ P(t, \alpha, C)$ for all C in S/R and $\alpha \in Act.$

$s \sim' t$, if there *exists* a strong bisimulation R such that $(s, t) \in R$.

It can be easily proven that $\sim$ coincides with $\sim'$. Proof is omitted.

Overview

1. Strong Bisimulation
2. Probabilistic Bisimulation
   - Quotient Markov Chain
   - Examples
3. Logical Preservation
   - The Logics PCTL, PCTL* and PCTL$^-$
   - Preservation Theorem
4. Lumpability
5. Summary

Probabilistic bisimulation: intuition

Probabilistic bisimulation: intuition

Intuition

Probabilistic bisimulation: intuition

Intuition

- Strong bisimulation is used to compare labeled transition systems.

Probabilistic bisimulation: intuition

Intuition

- Strong bisimulation is used to compare labeled transition systems.
- Strongly bisimilar states exhibit the same step-wise behaviour.

Probabilistic bisimulation: intuition

Intuition

- Strong bisimulation is used to compare labeled transition systems.
- Strongly bisimilar states exhibit the same step-wise behaviour.
- Our aim: adapt bisimulation to discrete-time Markov chains.

Probabilistic bisimulation: intuition

Intuition

- Strong bisimulation is used to compare labeled transition systems.
- Strongly bisimilar states exhibit the same step-wise behaviour.
- Our aim: adapt bisimulation to discrete-time Markov chains.
- This yields a probabilistic variant of strong bisimulation.

Probabilistic bisimulation: intuition

Intuition

- Strong bisimulation is used to compare labeled transition systems.
- Strongly bisimilar states exhibit the same step-wise behaviour.
- Our aim: adapt bisimulation to discrete-time Markov chains.
- This yields a probabilistic variant of strong bisimulation.

- When do two DTMC states exhibit the same step-wise behaviour?

Probabilistic bisimulation: intuition

Intuition

- Strong bisimulation is used to compare labeled transition systems.
- Strongly bisimilar states exhibit the same step-wise behaviour.
- Our aim: adapt bisimulation to discrete-time Markov chains.
- This yields a probabilistic variant of strong bisimulation.

- When do two DTMC states exhibit the same step-wise behaviour?
- Key: if their transition probability for each equivalence class coincides.

Probabilistic bisimulation

Probabilistic bisimulation

Probabilistic bisimulation [Larsen & Skou, 1989]

Probabilistic bisimulation

Probabilistic bisimulation **[Larsen & Skou, 1989]**

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.

Probabilistic bisimulation

Probabilistic bisimulation **[Larsen & Skou, 1989]**

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

Probabilistic bisimulation

Probabilistic bisimulation [Larsen & Skou, 1989]

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and

Probabilistic bisimulation

Probabilistic bisimulation **[Larsen & Skou, 1989]**

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$

where $\mathbf{P}(s, C) = \sum_{s' \in C} \mathbf{P}(s, s')$.

Probabilistic bisimulation

Probabilistic bisimulation **[Larsen & Skou, 1989]**

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$

where $\mathbf{P}(s, C) = \sum_{s' \in C} \mathbf{P}(s, s')$.

For states in R, the probability of moving to some equivalence class is equal.

Probabilistic bisimulation

Probabilistic bisimulation [Larsen & Skou, 1989]

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$

where $\mathbf{P}(s, C) = \sum_{s' \in C} \mathbf{P}(s, s')$.

For states in R, the probability of moving to some equivalence class is equal.

Probabilistic bisimilarity

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$.

Probabilistic bisimulation

Probabilistic bisimulation [Larsen & Skou, 1989]

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$

where $\mathbf{P}(s, C) = \sum_{s' \in C} \mathbf{P}(s, s')$.

For states in R, the probability of moving to some equivalence class is equal.

Probabilistic bisimilarity

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then: s is *probabilistic bisimilar* to t, denoted $s \sim_p t$, if there exists a probabilistic bisimulation R with $(s, t) \in R$.

Probabilistic bisimulation

Probabilistic bisimulation

Probabilistic bisimulation

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence. Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$.

Probabilistic bisimulation

Probabilistic bisimulation

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence.
Then: R is a *probabilistic bisimulation* on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$.

Remarks

As opposed to bisimulation on states in transition systems, any probabilistic bisimulation is an equivalence.

Example

Bisimilar DTMCs

Bisimilar DTMCs

Bisimilar DTMCs

Let $\mathcal{D}_1$, $\mathcal{D}_2$ be DTMCs over the same set of atomic propositions with initial distributions ι^1_{init} and ι^2_{init}, respectively.

Bisimilar DTMCs

Bisimilar DTMCs

Let $\mathcal{D}_1$, $\mathcal{D}_2$ be DTMCs over the same set of atomic propositions with initial distributions ι^1_{init} and ι^2_{init}, respectively.

Consider the DTMC $\mathcal{D} = \mathcal{D}_1 \uplus \mathcal{D}_2$ that results from the disjoint union of $\mathcal{D}_1$ and $\mathcal{D}_2$. Consider $\sim_p$ on $\mathcal{D} = \mathcal{D}_1 \uplus \mathcal{D}_2$.

Bisimilar DTMCs

Bisimilar DTMCs

Let $\mathcal{D}_1$, $\mathcal{D}_2$ be DTMCs over the same set of atomic propositions with initial distributions ι^1_{init} and ι^2_{init}, respectively.

Consider the DTMC $\mathcal{D} = \mathcal{D}_1 \uplus \mathcal{D}_2$ that results from the disjoint union of $\mathcal{D}_1$ and $\mathcal{D}_2$. Consider $\sim_p$ on $\mathcal{D} = \mathcal{D}_1 \uplus \mathcal{D}_2$.

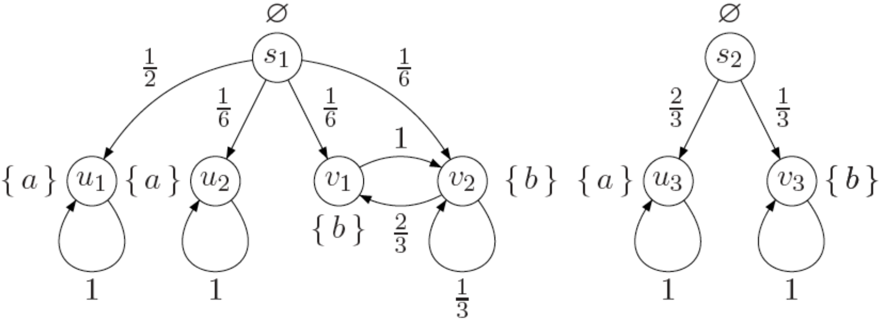
Then $\mathcal{D}_1$ and $\mathcal{D}_2$ are bisimilar, denoted $\mathcal{D}_1 \sim_p \mathcal{D}_2$ whenever

$$\iota^1_{\mathrm{init}}(C) = \iota^2_{\mathrm{init}}(C)$$

for each bisimulation equivalence class C of $\mathcal{D} = \mathcal{D}_1 \uplus \mathcal{D}_2$ under $\sim_p$.

Here, $\iota_{\mathrm{init}}(C)$ denotes $\sum_{s \in C} \iota_{\mathrm{init}}(s)$.

Example


∅
s_1
$\frac{1}{2}$
$\frac{1}{6}$
$\frac{1}{6}$
$\frac{1}{6}$
1
{ a } u_1
{ a } u_2
v_1
{ b }
$\frac{2}{3}$
v_2 { b }
1
1
$\frac{1}{3}$
∅
s_2
$\frac{2}{3}$
$\frac{1}{3}$
{ a } u_3
v_3 { b }
1
1

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

- $S' = S/\sim_p = \{ [s]_{\sim_p} \mid s \in S \}$ with $[s]_{\sim_p} = \{ s' \in S \mid s \sim_p s' \}$

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

- $S' = S/\sim_p = \{ [s]_{\sim_p} \mid s \in S \}$ with $[s]_{\sim_p} = \{ s' \in S \mid s \sim_p s' \}$
- $\mathbf{P}'([s]_{\sim_p}, [s']_{\sim_p}) = \mathbf{P}(s, [s']_{\sim_p})$

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

- $S' = S/\sim_p = \{ [s]_{\sim_p} \mid s \in S \}$ with $[s]_{\sim_p} = \{ s' \in S \mid s \sim_p s' \}$
- $\mathbf{P}'([s]_{\sim_p}, [s']_{\sim_p}) = \mathbf{P}(s, [s']_{\sim_p})$
- $\iota'_{\text{init}}([s]_{\sim_p}) = \sum_{s' \in [s]_{\sim_p}} \iota_{\text{init}}(s)$

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

- $S' = S/\sim_p = \{ [s]_{\sim_p} \mid s \in S \}$ with $[s]_{\sim_p} = \{ s' \in S \mid s \sim_p s' \}$
- $\mathbf{P}'([s]_{\sim_p}, [s']_{\sim_p}) = \mathbf{P}(s, [s']_{\sim_p})$
- $\iota'_{\text{init}}([s]_{\sim_p}) = \sum_{s' \in [s]_{\sim_p}} \iota_{\text{init}}(s)$
- $L'([s]_{\sim_p}) = L(s)$.

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

- $S' = S/\sim_p = \{ [s]_{\sim_p} \mid s \in S \}$ with $[s]_{\sim_p} = \{ s' \in S \mid s \sim_p s' \}$
- $\mathbf{P}'([s]_{\sim_p}, [s']_{\sim_p}) = \mathbf{P}(s, [s']_{\sim_p})$
- $\iota'_{\text{init}}([s]_{\sim_p}) = \sum_{s' \in [s]_{\sim_p}} \iota_{\text{init}}(s)$
- $L'([s]_{\sim_p}) = L(s)$.

Remarks

The transition probability from $[s]_{\sim_p}$ to $[t]_{\sim_p}$ equals $\mathbf{P}(s, [t]_{\sim_p})$.

Quotient under $\sim_p$

Quotient DTMC under $\sim_p$

For $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ and probabilistic bisimilarity $\sim_p \subseteq S \times S$ let

$$\mathcal{D}/\sim_p = (S', \mathbf{P}', \iota'_{\text{init}}, AP, L'), \quad \text{the } \textit{quotient} \text{ of } \mathcal{D} \text{ under } \sim_p$$

where

- $S' = S/\sim_p = \{ [s]_{\sim_p} \mid s \in S \}$ with $[s]_{\sim_p} = \{ s' \in S \mid s \sim_p s' \}$
- $\mathbf{P}'([s]_{\sim_p}, [s']_{\sim_p}) = \mathbf{P}(s, [s']_{\sim_p})$
- $\iota'_{\text{init}}([s]_{\sim_p}) = \sum_{s' \in [s]_{\sim_p}} \iota_{\text{init}}(s)$
- $L'([s]_{\sim_p}) = L(s)$.

Remarks

The transition probability from $[s]_{\sim_p}$ to $[t]_{\sim_p}$ equals $\mathbf{P}(s, [t]_{\sim_p})$. This is well-defined as $\mathbf{P}(s, C) = \mathbf{P}(s', C)$ for all $s \sim_p s'$ and all bisimulation equivalence classes C.

Example

Craps

- Roll two dice and bet

Craps

- Roll two dice and bet
- Come-out roll (“pass line” wager):
  - outcome 7 or 11: win

Craps

- Roll two dice and bet
- Come-out roll (“pass line” wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose (“craps”)

Craps

- Roll two dice and bet
- Come-out roll ("pass line" wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose ("craps")
  - any other outcome: roll again (outcome is "point")

Craps

- Roll two dice and bet
- Come-out roll (“pass line” wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose (“craps”)
  - any other outcome: roll again (outcome is “point”)
- Repeat until 7 or the “point” is thrown:

Craps

- Roll two dice and bet
- Come-out roll ("pass line" wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose ("craps")
  - any other outcome: roll again (outcome is "point")
- Repeat until 7 or the "point" is thrown:
  - outcome 7: lose ("seven-out")

Craps

- Roll two dice and bet
- Come-out roll ("pass line" wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose ("craps")
  - any other outcome: roll again (outcome is "point")
- Repeat until 7 or the "point" is thrown:
  - outcome 7: lose ("seven-out")
  - outcome the point: win

Craps

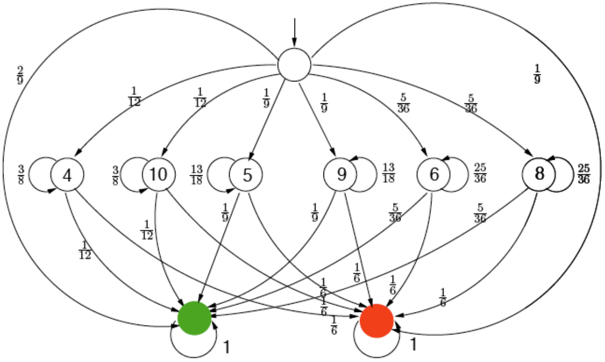
- Roll two dice and bet
- Come-out roll ("pass line" wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose ("craps")
  - any other outcome: roll again (outcome is "point")
- Repeat until 7 or the "point" is thrown:
  - outcome 7: lose ("seven-out")
  - outcome the point: win
  - any other outcome: roll again

A DTMC model of Craps

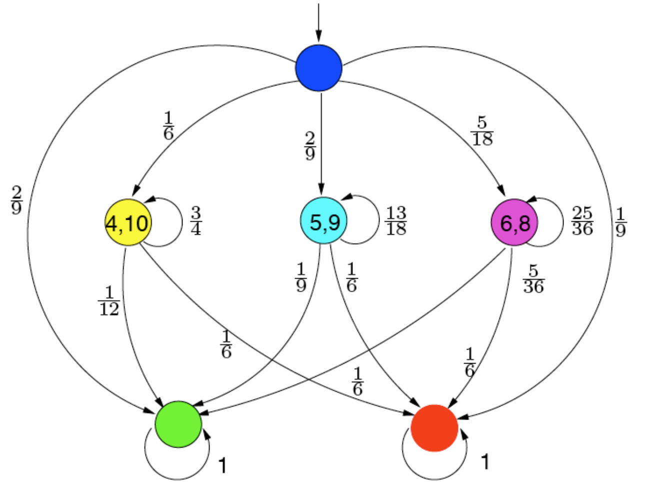
- Come-out roll:
  - 7 or 11: win
  - 2, 3, or 12: lose
  - else: roll again
- Next roll(s):
  - 7: lose
  - point: win
  - else: roll again

A DTMC model of Craps

- Come-out roll:
  - 7 or 11: win
  - 2, 3, or 12: lose
  - else: roll again
- Next roll(s):
  - 7: lose
  - point: win
  - else: roll again



Quotient DTMC of Craps under $\sim_p$



Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd
  - sender selects a crowd member randomly using a uniform distribution

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd
  - sender selects a crowd member randomly using a uniform distribution
  - selected router flips a biased coin:

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd
  - sender selects a crowd member randomly using a uniform distribution
  - selected router flips a biased coin:
    - with probability $1 - p$: direct delivery to final destination
    - otherwise: select a next router randomly (uniformly)

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd
  - sender selects a crowd member randomly using a uniform distribution
  - selected router flips a biased coin:
    - with probability $1 - p$: direct delivery to final destination
    - otherwise: select a next router randomly (uniformly)
  - once a routing path has been established, use it until crowd changes

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

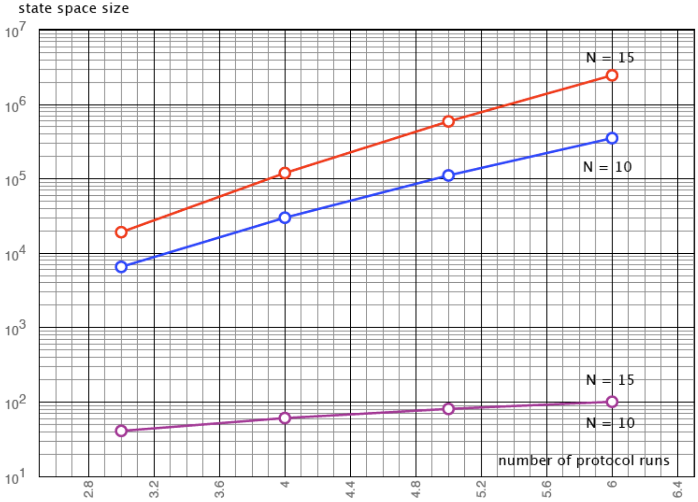
- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd
  - sender selects a crowd member randomly using a uniform distribution
  - selected router flips a biased coin:
    - with probability $1 - p$: direct delivery to final destination
    - otherwise: select a next router randomly (uniformly)
  - once a routing path has been established, use it until crowd changes
- Rebuild routing paths on crowd changes

Example: Crowds protocol

Security: Crowds protocol [Reiter & Rubin, 1998]

- A protocol for anonymous web browsing (variants: mCrowds, BT-Crowds)
- Hide user's communication by random routing within a crowd
  - sender selects a crowd member randomly using a uniform distribution
  - selected router flips a biased coin:
    - with probability $1-p$: direct delivery to final destination
    - otherwise: select a next router randomly (uniformly)
  - once a routing path has been established, use it until crowd changes
- Rebuild routing paths on crowd changes
- Property: Crowds protocol ensures "probable innocence":
  - probability real sender is discovered $< \frac{1}{2}$ if $N \geqslant \frac{p}{p-\frac{1}{2}} \cdot (c+1)$
  - where N is crowd's size and c is number of corrupt crowd members

State space reduction under $\sim_p$



IEEE 802.11 group communication protocol

	original DTMC			quotient DTMC		red. factor	
OD	states	transitions	ver. time	blocks	total time	states	time
4	1125	5369	122	71	13	**15.9**	**9.00**
12	37349	236313	7180	1821	642	**20.5**	**11.2**
20	231525	1590329	50133	10627	5431	**21.8**	**9.2**
28	804837	5750873	195086	35961	24716	**22.4**	**7.9**
36	2076773	15187833	5103900	91391	77694	**22.7**	**6.6**
40	3101445	22871849	7725041	135752	127489	**22.9**	**6.1**

all times in milliseconds

Overview

PCTL syntax

Probabilistic Computation Tree Logic: Syntax

PCTL consists of state- and path-formulas.

PCTL syntax

Probabilistic Computation Tree Logic: Syntax

PCTL consists of state- and path-formulas.

- PCTL *state formulas* over the set AP obey the grammar:

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \mathbb{P}_J(\varphi)$$

where $a \in AP$, φ is a path formula and interval $J \subseteq [0, 1]$.

PCTL syntax

Probabilistic Computation Tree Logic: Syntax

PCTL consists of state- and path-formulas.

- PCTL *state formulas* over the set AP obey the grammar:

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \mathbb{P}_J(\varphi)$$

  where $a \in AP$, φ is a path formula and interval $J \subseteq [0, 1]$.

- PCTL *path formulae* are formed according to the following grammar:

$$\varphi ::= \bigcirc \Phi \;\Big|\; \Phi_1 \, \mathsf{U} \, \Phi_2 \;\Big|\; \Phi_1 \, \mathsf{U}^{\leqslant n} \, \Phi_2$$

  where Φ, Φ_1, and Φ_2 are state formulae and $n \in \mathbb{N}$.

Preservation of PCTL-formulas

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Remarks

$s \sim_p t$ implies that

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$s \sim_p t$ if and only if s and t are PCTL-equivalent.

Remarks

$s \sim_p t$ implies that

1. transient probabilities, reachability probabilities,

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Remarks

$s \sim_p t$ implies that

1. transient probabilities, reachability probabilities,
2. repeated reachability, persistence probabilities

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Remarks

$s \sim_p t$ implies that

1. transient probabilities, reachability probabilities,
2. repeated reachability, persistence probabilities
3. all qualitative PCTL formulas

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Remarks

$s \sim_p t$ implies that

1. transient probabilities, reachability probabilities,
2. repeated reachability, persistence probabilities
3. all qualitative PCTL formulas

for s and t are equal.

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Remarks

$s \sim_p t$ implies that

1. transient probabilities, reachability probabilities,
2. repeated reachability, persistence probabilities
3. all qualitative PCTL formulas

for s and t are equal.

If for PCTL-formula Φ we have $s \models \Phi$ but $t \not\models \Phi$, then it follows $s \not\sim_p t$.

Preservation of PCTL-formulas

Bisimulation preserves PCTL

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL-equivalent.}$$

Remarks

$s \sim_p t$ implies that

1. transient probabilities, reachability probabilities,
2. repeated reachability, persistence probabilities
3. all qualitative PCTL formulas

for s and t are equal.

If for PCTL-formula Φ we have $s \models \Phi$ but $t \not\models \Phi$, then it follows $s \not\sim_p t$.
A single PCTL-formula suffices!

PCTL* syntax

Probabilistic Computation Tree Logic: Syntax

PCTL* consists of state- and path-formulas.

PCTL* syntax

Probabilistic Computation Tree Logic: Syntax

PCTL* consists of state- and path-formulas.

- PCTL* *state formulas* over the set AP obey the grammar:

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \mathbb{P}_J(\varphi)$$

  where $a \in AP$, φ is a path formula and $J \subseteq [0, 1]$, $J \neq \varnothing$ is a non-empty interval.

PCTL* syntax

Probabilistic Computation Tree Logic: Syntax

PCTL* consists of state- and path-formulas.

- PCTL* *state formulas* over the set AP obey the grammar:

$$\Phi \ ::= \ \text{true} \ \Big| \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \neg\Phi \ \Big| \ \mathbb{P}_J(\varphi)$$

  where $a \in AP$, φ is a path formula and $J \subseteq [0,1]$, $J \neq \varnothing$ is a non-empty interval.

- PCTL* *path formulae* are formed according to the following grammar:

$$\varphi \ ::= \Phi \ \Big| \ \neg\varphi \ \Big| \ \varphi_1 \wedge \varphi_2 \ \Big| \ \bigcirc \varphi \ \Big| \ \varphi_1 \, \mathsf{U} \, \varphi_2$$

  where Φ is a state formula and φ, φ_1, and φ_2 are path formulae.

PCTL* semantics (1)

PCTL* semantics (1)

Notation

$\mathcal{D}, s \models \Phi$ if and only if state-formula Φ holds in state s of (possibly infinite) DTMC $\mathcal{D}$. As $\mathcal{D}$ is known from the context we simply write $s \models \Phi$.

PCTL* semantics (1)

Notation

$\mathcal{D}, s \models \Phi$ if and only if state-formula Φ holds in state s of (possibly infinite) DTMC $\mathcal{D}$. As $\mathcal{D}$ is known from the context we simply write $s \models \Phi$.

Satisfaction relation for state formulas

The satisfaction relation $\models$ is defined for PCTL* state formulas by:

$$\begin{array}{lll} s \models a & \text{iff} & a \in L(s) \\ s \models \neg\Phi & \text{iff} & \text{not } (s \models \Phi) \\ s \models \Phi \wedge \Psi & \text{iff} & (s \models \Phi) \text{ and } (s \models \Psi) \end{array}$$

PCTL* semantics (1)

Notation

$\mathcal{D}, s \models \Phi$ if and only if state-formula Φ holds in state s of (possibly infinite) DTMC $\mathcal{D}$. As $\mathcal{D}$ is known from the context we simply write $s \models \Phi$.

Satisfaction relation for state formulas

The satisfaction relation $\models$ is defined for PCTL* state formulas by:

$$
\begin{array}{lll}
s \models a & \text{iff} & a \in L(s) \\
s \models \neg\,\Phi & \text{iff} & \text{not } (s \models \Phi) \\
s \models \Phi \wedge \Psi & \text{iff} & (s \models \Phi) \text{ and } (s \models \Psi) \\
s \models \mathbb{P}_J(\varphi) & \text{iff} & Pr(s \models \varphi) \in J
\end{array}
$$

where $Pr(s \models \varphi) = Pr_s\{\, \pi \in Paths(s) \mid \pi \models \varphi \,\}$

PCTL* semantics (2)

PCTL* semantics (2)

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2 \ldots$ be an infinite path in (possibly infinite) DTMC $\mathcal{D}$.

PCTL* semantics (2)

Satisfaction relation for path formulas

Let $\pi = s_0 \, s_1 \, s_2 \ldots$ be an infinite path in (possibly infinite) DTMC $\mathcal{D}$. Let $\pi^i = s_i \, s_{i+1} \, s_{i+2} \ldots$ denotes the i-th suffix of π.

The satisfaction relation $\models$ is defined for state formulas by:

$$\pi \models \Phi \qquad \text{iff} \quad \pi[0] \models \Phi$$

PCTL* semantics (2)

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2 \ldots$ be an infinite path in (possibly infinite) DTMC $\mathcal{D}$. Let $\pi^i = s_i\, s_{i+1}\, s_{i+2} \ldots$ denotes the i-th suffix of π.

The satisfaction relation $\models$ is defined for state formulas by:

$$
\begin{array}{lll}
\pi \models \Phi & \text{iff} & \pi[0] \models \Phi \\
\pi \models \neg\varphi & \text{iff} & \text{not } \pi \models \varphi \\
\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \pi \models \varphi_1 \text{ and } \pi \models \varphi_2 \\
\pi \models \bigcirc\varphi & \text{iff} & \pi^1 \models \varphi
\end{array}
$$

PCTL* semantics (2)

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2 \ldots$ be an infinite path in (possibly infinite) DTMC $\mathcal{D}$. Let $\pi^i = s_i\, s_{i+1}\, s_{i+2} \ldots$ denotes the i-th suffix of π.

The satisfaction relation $\models$ is defined for state formulas by:

$$
\begin{array}{lll}
\pi \models \Phi & \text{iff} & \pi[0] \models \Phi \\
\pi \models \neg\varphi & \text{iff} & \text{not } \pi \models \varphi \\
\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \pi \models \varphi_1 \text{ and } \pi \models \varphi_2 \\
\pi \models \bigcirc\varphi & \text{iff} & \pi^1 \models \varphi \\
\pi \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \exists k \geqslant 0. (\pi^k \models \varphi_2 \wedge \forall 0 \leqslant i < k. \pi^i \models \varphi_1)
\end{array}
$$

Measurability

Measurability

PCTL* measurability

For any PCTL* path formula φ and state s of DTMC $\mathcal{D}$,
the set $\{ \pi \in \mathit{Paths}(s) \mid \pi \models \varphi \}$ is measurable.

Proof:

Left as an exercise, using the result for PCTL measurability and the measurability of ω-regular properties.

Bounded until in PCTL*

Bounded until in PCTL*

Bounded until

Bounded until can be defined using the other operators:

Bounded until in PCTL*

Bounded until

Bounded until can be defined using the other operators:

$$\varphi_1 \, \mathsf{U}^{\leqslant n} \varphi_2 \;=\; \bigvee_{0 \leqslant i \leqslant n} \psi_i \quad \text{where } \psi_0 = \varphi_2 \text{ and } \psi_{i+1} = \varphi_1 \wedge \bigcirc \psi_i \text{ for } i \geqslant 0.$$

Bounded until in PCTL*

Bounded until

Bounded until can be defined using the other operators:

$$\varphi_1 \, \mathsf{U}^{\leqslant n} \varphi_2 \;=\; \bigvee_{0 \leqslant i \leqslant n} \psi_i \quad \text{where } \psi_0 = \varphi_2 \text{ and } \psi_{i+1} = \varphi_1 \wedge \bigcirc \psi_i \text{ for } i \geqslant 0.$$

Examples in PCTL* but not in PCTL

$\mathbb{P}_{>\frac{1}{4}}(\bigcirc a \, \mathsf{U} \bigcirc b)$

Bounded until in PCTL*

Bounded until

Bounded until can be defined using the other operators:

$$\varphi_1 \, \mathsf{U}^{\leqslant n} \varphi_2 \;=\; \bigvee_{0 \leqslant i \leqslant n} \psi_i \quad \text{where } \psi_0 = \varphi_2 \text{ and } \psi_{i+1} = \varphi_1 \wedge \bigcirc \psi_i \text{ for } i \geqslant 0.$$

Examples in PCTL* but not in PCTL

$\mathbb{P}_{>\frac{1}{4}}(\bigcirc a \, \mathsf{U} \bigcirc b)$ and $\mathbb{P}_{=1}(\mathbb{P}_{>\frac{1}{2}}(\Box\Diamond a \vee \Diamond\Box b))$.

Preservation of PCTL*-formulas

Preservation of PCTL*-formulas

Bisimulation preserves PCTL*

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL}^*\text{-equivalent.}$$

Preservation of PCTL*-formulas

Bisimulation preserves PCTL*

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL}^*\text{-equivalent.}$$

Remarks

Preservation of PCTL*-formulas

Bisimulation preserves PCTL*

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL}^*\text{-equivalent.}$$

Remarks

1. Bisimulation thus preserves not only all PCTL but also all PCTL* formulas.

Preservation of PCTL*-formulas

Bisimulation preserves PCTL*

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL}^*\text{-equivalent.}$$

Remarks

1. Bisimulation thus preserves not only all PCTL but also all PCTL* formulas.
2. By the last two results it follows that PCTL- and PCTL*-equivalence coincide.

Preservation of PCTL*-formulas

Bisimulation preserves PCTL*

Let $\mathcal{D}$ be a DTMC and s, t states in $\mathcal{D}$. Then:

$$s \sim_p t \quad \text{if and only if} \quad s \text{ and } t \text{ are PCTL}^*\text{-equivalent.}$$

Remarks

1. Bisimulation thus preserves not only all PCTL but also all PCTL* formulas.
2. By the last two results it follows that PCTL- and PCTL*-equivalence coincide. Thus any two states that satisfy the same PCTL formulas, satisfy the same PCTL* formulas.

PCTL$^-$ syntax

PCTL^{-} syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL^{-} only consists of state-formulas. These formulas over the set AP obey the grammar:

$$\Phi \ ::= \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \Phi_1 \vee \Phi_2 \ \Big| \ \mathbb{P}_{\leqslant p}(\bigcirc \Phi)$$

where $a \in AP$ and p is a probability in $[0, 1]$.

PCTL$^-$ syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL$^-$ only consists of state-formulas. These formulas over the set AP obey the grammar:

$$\Phi \ ::= \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \Phi_1 \vee \Phi_2 \ \Big| \ \mathbb{P}_{\leqslant p}(\bigcirc \Phi)$$

where $a \in AP$ and p is a probability in $[0, 1]$.

Remarks

This is a truly simple logic.

PCTL$^-$ syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL$^-$ only consists of state-formulas. These formulas over the set AP obey the grammar:

$$\Phi \ ::= \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \Phi_1 \vee \Phi_2 \ \Big| \ \mathbb{P}_{\leqslant p}(\bigcirc \Phi)$$

where $a \in AP$ and p is a probability in $[0, 1]$.

Remarks

This is a truly simple logic. It does not contain the until-operator.

PCTL$^-$ syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL$^-$ only consists of state-formulas. These formulas over the set AP obey the grammar:

$$\Phi ::= a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \Phi_1 \vee \Phi_2 \;\Big|\; \mathbb{P}_{\leqslant p}(\bigcirc \Phi)$$

where $a \in AP$ and p is a probability in $[0, 1]$.

Remarks

This is a truly simple logic. It does not contain the until-operator. Negation is not present and cannot be expressed.

PCTL$^-$ syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL$^-$ only consists of state-formulas. These formulas over the set AP obey the grammar:

$$\Phi \ ::= \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \Phi_1 \vee \Phi_2 \ \Big| \ \mathbb{P}_{\leqslant p}(\bigcirc \Phi)$$

where $a \in AP$ and p is a probability in $[0, 1]$.

Remarks

This is a truly simple logic. It does not contain the until-operator. Negation is not present and cannot be expressed. Only upper bounds on probabilities.

PCTL$^-$ syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL$^-$ only consists of state-formulas. These formulas over the set AP obey the grammar:

$$\Phi \ ::= \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \Phi_1 \vee \Phi_2 \ \Big| \ \mathbb{P}_{\leqslant p}(\bigcirc \Phi)$$

where $a \in AP$ and p is a probability in $[0, 1]$.

Remarks

This is a truly simple logic. It does not contain the until-operator. Negation is not present and cannot be expressed. Only upper bounds on probabilities.

The next theorem shows that PCTL-, PCTL*- and PCTL$^-$-equivalence coincide.

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

(d) s_1 and s_2 are PCTL$^-$-equivalent, i.e., fulfill the same PCTL$^-$ formulas

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

(d) s_1 and s_2 are PCTL$^-$-equivalent, i.e., fulfill the same PCTL$^-$ formulas

Proof:

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

(d) s_1 and s_2 are PCTL$^-$-equivalent, i.e., fulfill the same PCTL$^-$ formulas

Proof:

1. (a) $\Longrightarrow$ (b): by structural induction on PCTL* formulas.

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

(d) s_1 and s_2 are PCTL$^-$-equivalent, i.e., fulfill the same PCTL$^-$ formulas

Proof:

1. (a) $\Longrightarrow$ (b): by structural induction on PCTL* formulas.
2. (b) $\Longrightarrow$ (c): trivial as PCTL is a sublogic of PCTL*.

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

(d) s_1 and s_2 are PCTL$^-$-equivalent, i.e., fulfill the same PCTL$^-$ formulas

Proof:

1. (a) $\Longrightarrow$ (b): by structural induction on PCTL* formulas.
2. (b) $\Longrightarrow$ (c): trivial as PCTL is a sublogic of PCTL*.
3. (c) $\Longrightarrow$ (d): trivial as PCTL$-$ is a sublogic of PCTL.

Preservation of PCTL

PCTL/PCTL* and Bisimulation Equivalence

Let $\mathcal{D}$ be a DTMC and s_1, s_2 states in $\mathcal{D}$. Then, the following statements are equivalent:

(a) $s_1 \sim_p s_2$.

(b) s_1 and s_2 are PCTL*-equivalent, i.e., fulfill the same PCTL* formulas

(c) s_1 and s_2 are PCTL-equivalent, i.e., fulfill the same PCTL formulas

(d) s_1 and s_2 are PCTL$^-$-equivalent, i.e., fulfill the same PCTL$^-$ formulas

Proof:

1. (a) $\Longrightarrow$ (b): by structural induction on PCTL* formulas.
2. (b) $\Longrightarrow$ (c): trivial as PCTL is a sublogic of PCTL*.
3. (c) $\Longrightarrow$ (d): trivial as PCTL$-$ is a sublogic of PCTL.
4. (d) $\Longrightarrow$ (a): involved. First finite DTMCs, then for arbitrary DTMCs.

Proof

Overview

1960: Laurie Snell and John Kemeny

Lumpability

Ignore the initial distribution and state-labelling of a Markov chain.

Lumpability **[Kemeny & Snell, 1960]**

Let $\mathcal{D}$ be a (possibly countably infinite) DTMC with state space S and $\mathcal{B} = \{ B_1, \ldots, B_n \}$ be a partitioning of S (where B_j may be countably infinite). $\mathcal{D}$ is lumpable with respect to $\mathcal{B}$ iff for any B_i and B_j in $\mathcal{B}$ and any $s, s' \in B_i$:

$$\sum_{u \in B_j} \mathbf{P}(s, u) \; = \; \sum_{u \in B_j} \mathbf{P}(s', u) \quad \text{that is} \quad \mathbf{P}(s, B_j) = \mathbf{P}(s', B_j).$$

Lumpability

Ignore the initial distribution and state-labelling of a Markov chain.

Lumpability [Kemeny & Snell, 1960]

Let $\mathcal{D}$ be a (possibly countably infinite) DTMC with state space S and $\mathcal{B} = \{ B_1, \ldots, B_n \}$ be a partitioning of S (where B_j may be countably infinite). $\mathcal{D}$ is lumpable with respect to $\mathcal{B}$ iff for any B_i and B_j in $\mathcal{B}$ and any $s, s' \in B_i$:

$$\sum_{u \in B_j} \mathbf{P}(s, u) \;=\; \sum_{u \in B_j} \mathbf{P}(s', u) \quad \text{that is} \quad \mathbf{P}(s, B_j) = \mathbf{P}(s', B_j).$$

If $\mathcal{D}$ is lumpable with respect to $\mathcal{B}$, $\mathcal{B}$ is called a lumpable partition

Lumpability

Ignore the initial distribution and state-labelling of a Markov chain.

Lumpability **[Kemeny & Snell, 1960]**

Let $\mathcal{D}$ be a (possibly countably infinite) DTMC with state space S and $\mathcal{B} = \{ B_1, \ldots, B_n \}$ be a partitioning of S (where B_j may be countably infinite). $\mathcal{D}$ is lumpable with respect to $\mathcal{B}$ iff for any B_i and B_j in $\mathcal{B}$ and any $s, s' \in B_i$:

$$\sum_{u \in B_j} \mathbf{P}(s, u) \;=\; \sum_{u \in B_j} \mathbf{P}(s', u) \quad \text{that is} \quad \mathbf{P}(s, B_j) = \mathbf{P}(s', B_j).$$

If $\mathcal{D}$ is lumpable with respect to $\mathcal{B}$, $\mathcal{B}$ is called a lumpable partition

It is easy to show that $S/\sim_p$ is a lumpable partition of the state space S. In fact, it is the coarsest possible lumpable partition.

Lumping equivalence

Lumping equivalence **[Kemeny & Snell, 1960]**

The DTMCs $\mathcal{D}$ and $\mathcal{D}'$ are lumping equivalent if there are lumpable partitions $\mathcal{B}$ of $\mathcal{D}$ and $\mathcal{B}'$ of $\mathcal{D}'$ such that there is an injective function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that:

$$\mathbf{P}(B_i, B_j) \ = \ \mathbf{P}'(B'_{f(i)}, B'_{f(j)}).$$

Lumping equivalence

Lumping equivalence **[Kemeny & Snell, 1960]**

The DTMCs $\mathcal{D}$ and $\mathcal{D}'$ are lumping equivalent if there are lumpable partitions $\mathcal{B}$ of $\mathcal{D}$ and $\mathcal{B}'$ of $\mathcal{D}'$ such that there is an injective function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that:

$$\mathbf{P}(B_i, B_j) \;=\; \mathbf{P}'(B'_{f(i)}, B'_{f(j)}).$$

Corollary

$D \sim_p D'$ if and only if $\mathcal{D}$ and $\mathcal{D}'$ are lumping equivalent (with respect to the coarsest possible lumpable partition on their union).

Lumping equivalence

Remark

For finite Markov chains, the correspondence between lumping equivalence and $\sim_p$ allows to obtain the coarsest possible lumpable partition in an algorithmic, i.e., automated manner.

Lumping equivalence

Remark

For finite Markov chains, the correspondence between lumping equivalence and $\sim_p$ allows to obtain the coarsest possible lumpable partition in an algorithmic, i.e., automated manner.

This can be considered as a breakthrough in Markov chain theory.

Overview

Summary

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- All states in a quotient DTMC are equivalence classes under $\sim_p$.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- All states in a quotient DTMC are equivalence classes under $\sim_p$.
- $\sim_p$ and PCTL-equivalence coincide.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- All states in a quotient DTMC are equivalence classes under $\sim_p$.
- $\sim_p$ and PCTL-equivalence coincide.
- PCTL, PCTL*, and PCTL$^-$-equivalence coincide.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- All states in a quotient DTMC are equivalence classes under $\sim_p$.
- $\sim_p$ and PCTL-equivalence coincide.
- PCTL, PCTL*, and PCTL$^-$-equivalence coincide.
- To show $s \not\sim_p t$, show $s \models \Phi$ and $t \not\models \Phi$ for $\Phi \in$ PCTL$^-$.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- All states in a quotient DTMC are equivalence classes under $\sim_p$.
- $\sim_p$ and PCTL-equivalence coincide.
- PCTL, PCTL*, and PCTL$^-$-equivalence coincide.
- To show $s \not\sim_p t$, show $s \models \Phi$ and $t \not\models \Phi$ for $\Phi \in$ PCTL$^-$.
- Bisimulation may yield up to exponential savings in state space.

Summary

- Bisimilar states have equal transition probabilities for every equivalence class.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- All states in a quotient DTMC are equivalence classes under $\sim_p$.
- $\sim_p$ and PCTL-equivalence coincide.
- PCTL, PCTL*, and PCTL$^-$-equivalence coincide.
- To show $s \not\sim_p t$, show $s \models \Phi$ and $t \not\models \Phi$ for $\Phi \in \text{PCTL}^-$.
- Bisimulation may yield up to exponential savings in state space.

Take-home message

Probabilistic bisimulation coincides with a notion from the sixties, named (ordinary) lumpability.