UNIVERSITY OF TWENTE.

Formal Methods & Tools.

Confluence Reduction for Probabilistic Systems

Mark Timmer

September 4, 2010

Joint work with
Mariëlle Stoelinga and Jaco van de Pol

YR-CONCUR 2010

Contents

Table of Contents

The context: probabilistic model checking

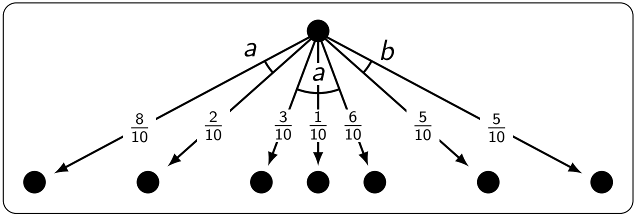
Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., a probabilistic automaton)

The context: probabilistic model checking

Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., a probabilistic automaton)

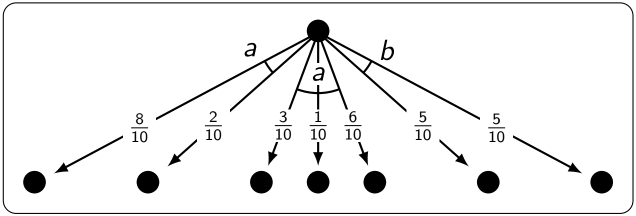


- Non-deterministically choose one of the three transitions
- Probabilistically choose the next state

The context: probabilistic model checking

Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., a probabilistic automaton)

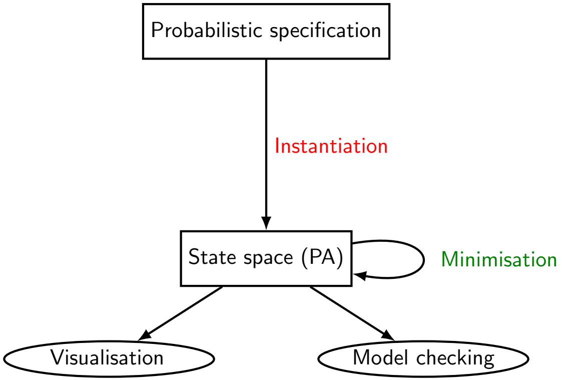


- Non-deterministically choose one of the three transitions
- Probabilistically choose the next state

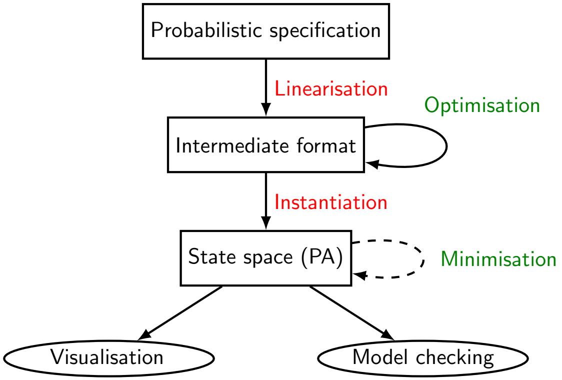
Limitations of previous approaches:

- Susceptible to the state space explosion problem
- Restricted treatment of data

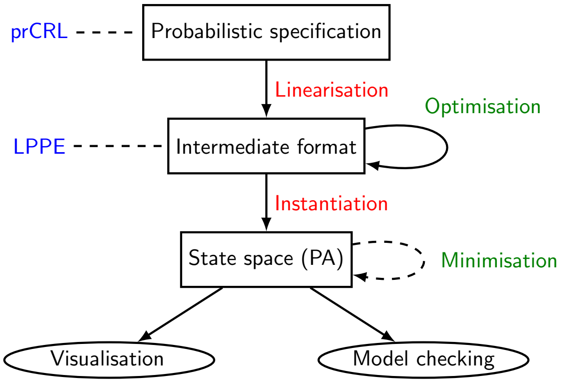
Overview of our approach



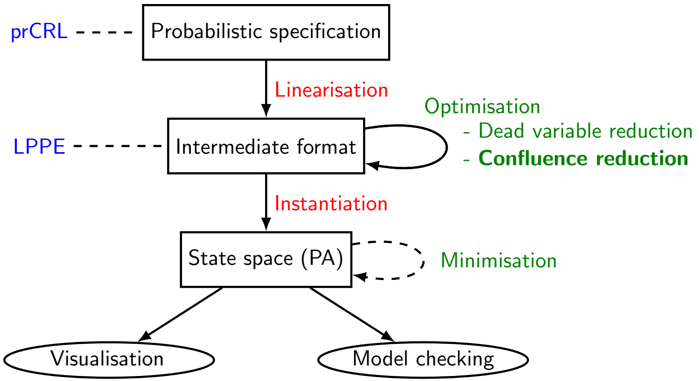
Overview of our approach



Overview of our approach

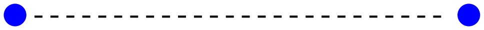


Overview of our approach



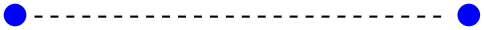
Branching probabilistic bisimulation

Notion of equivalence: branching probabilistic bisimulation



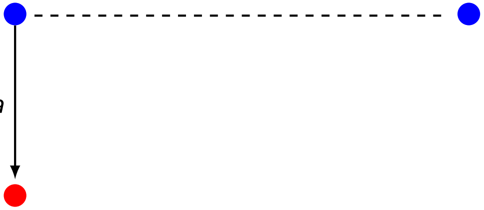
Branching probabilistic bisimulation

Notion of equivalence: strong bisimulation



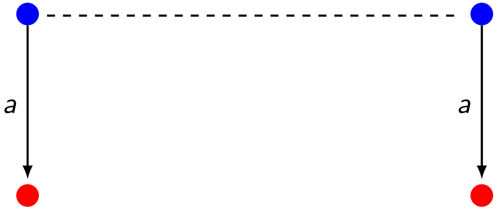
Branching probabilistic bisimulation

Notion of equivalence: strong bisimulation



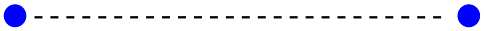
Branching probabilistic bisimulation

Notion of equivalence: strong bisimulation



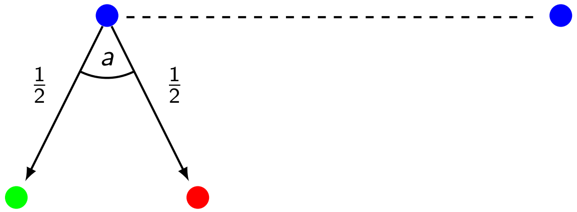
Branching probabilistic bisimulation

Notion of equivalence: strong probabilistic bisimulation



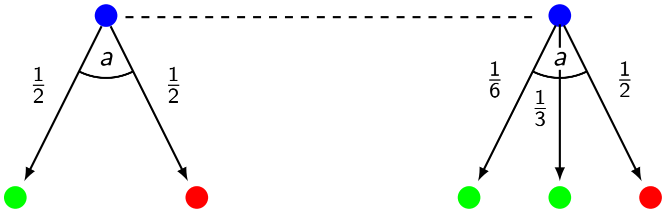
Branching probabilistic bisimulation

Notion of equivalence: strong probabilistic bisimulation



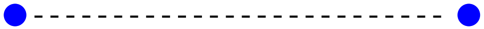
Branching probabilistic bisimulation

Notion of equivalence: strong probabilistic bisimulation



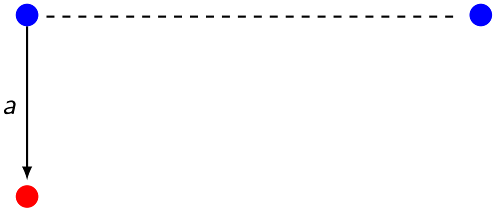
Branching probabilistic bisimulation

Notion of equivalence: branching bisimulation



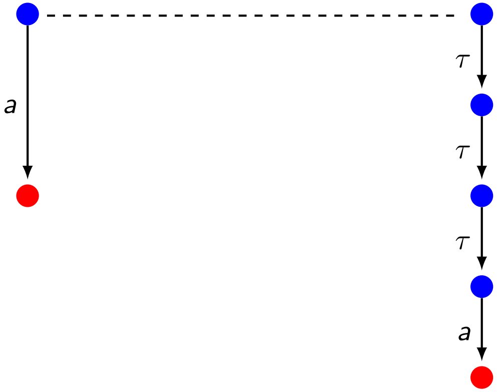
Branching probabilistic bisimulation

Notion of equivalence: branching bisimulation



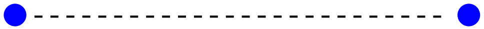
Branching probabilistic bisimulation

Notion of equivalence: branching bisimulation



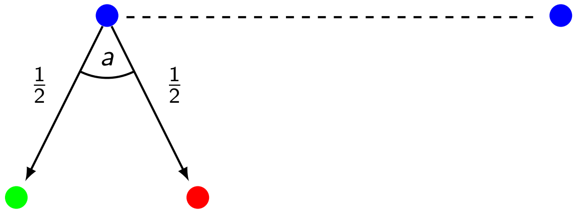
Branching probabilistic bisimulation

Notion of equivalence: branching probabilistic bisimulation



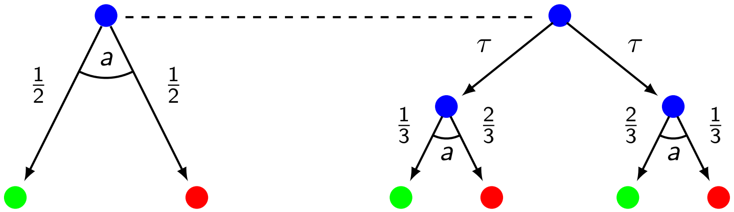
Branching probabilistic bisimulation

Notion of equivalence: branching probabilistic bisimulation



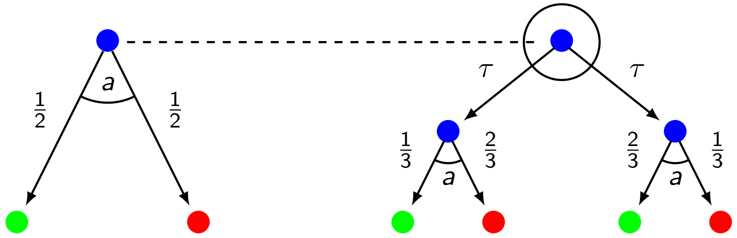
Branching probabilistic bisimulation

Notion of equivalence: branching probabilistic bisimulation



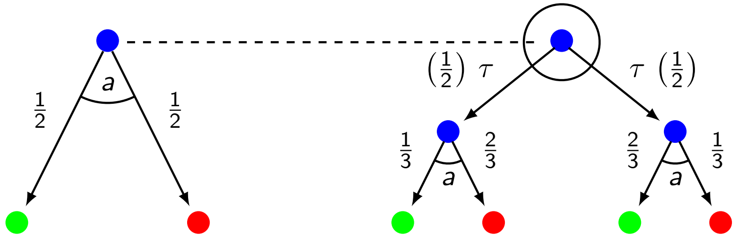
Branching probabilistic bisimulation

Notion of equivalence: branching probabilistic bisimulation



Branching probabilistic bisimulation

Notion of equivalence: branching probabilistic bisimulation

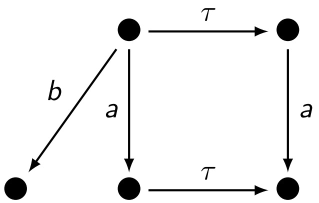


Probability of green: $\frac{1}{2} \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{2}{3} = \frac{1}{2}$

Table of Contents

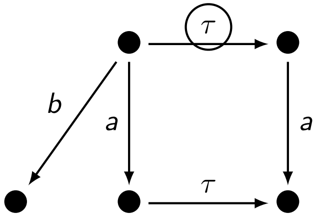
Branching bisimulation preservation by τ-steps

Unobservable τ-steps might disable behaviour...



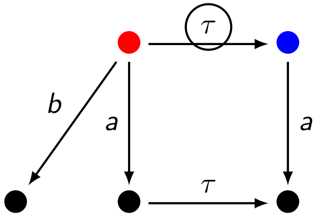
Branching bisimulation preservation by τ-steps

Unobservable τ-steps might disable behaviour…



Branching bisimulation preservation by τ-steps

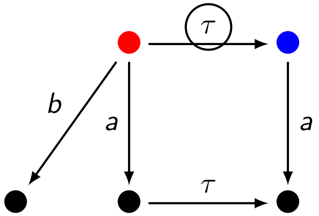
Unobservable τ-steps might disable behaviour...



Branching bisimulation preservation by τ-steps

Unobservable τ-steps might disable behaviour...

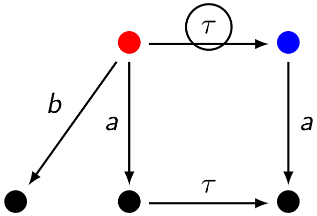
...though often, they connect branching bisimilar states

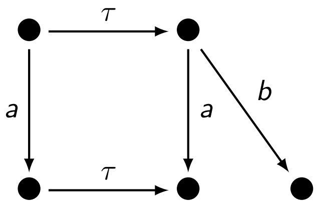


Branching bisimulation preservation by τ-steps

Unobservable τ-steps might disable behaviour. . .

. . . though often, they connect branching bisimilar states

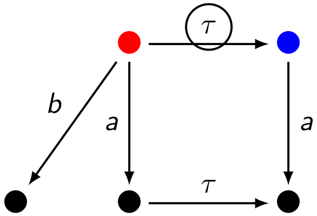


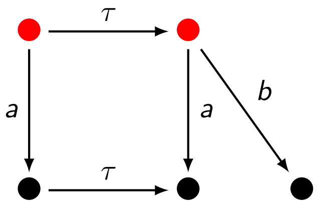


Branching bisimulation preservation by τ-steps

Unobservable τ-steps might disable behaviour...

...though often, they connect branching bisimilar states

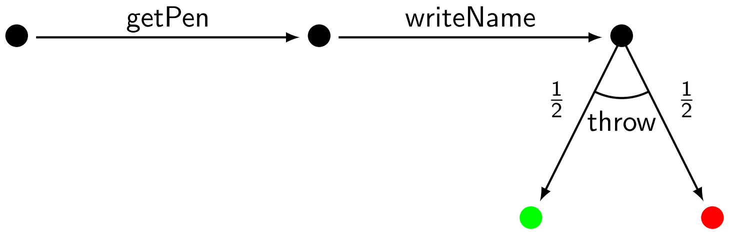




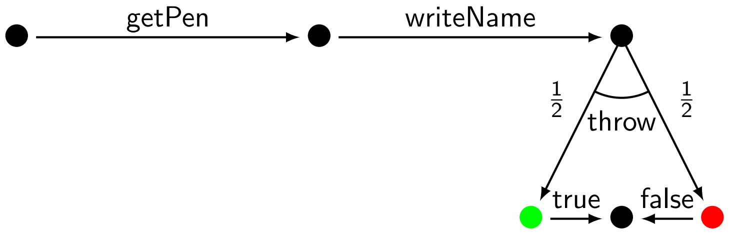
Confluence: an introductory example

$$\begin{tikzcd}\bullet \arrow[r, "\mathsf{getPen}"] & \bullet \arrow[r, "\mathsf{writeName}"] & \bullet\end{tikzcd}$$

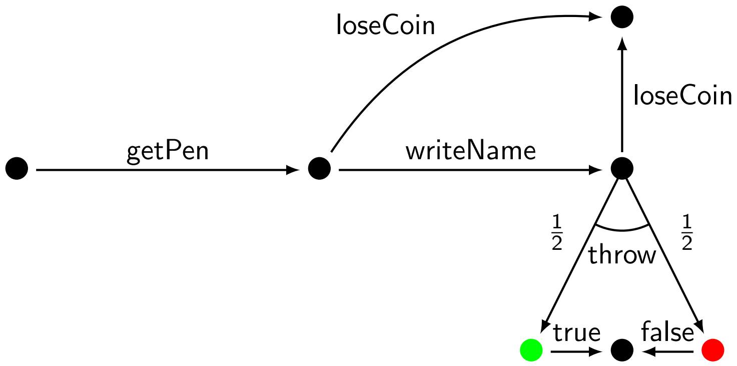
Confluence: an introductory example



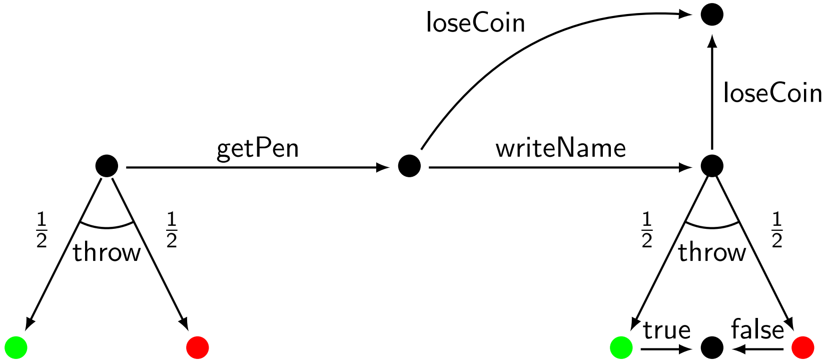
Confluence: an introductory example



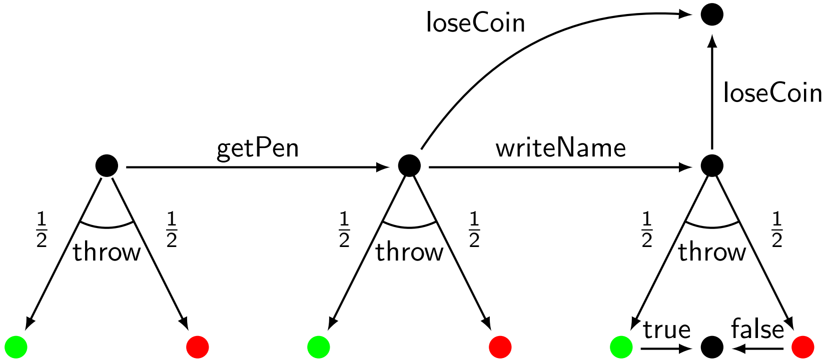
Confluence: an introductory example



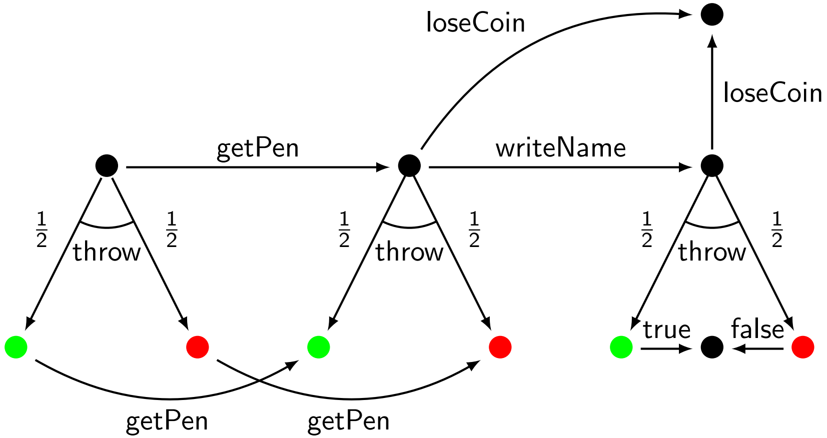
Confluence: an introductory example



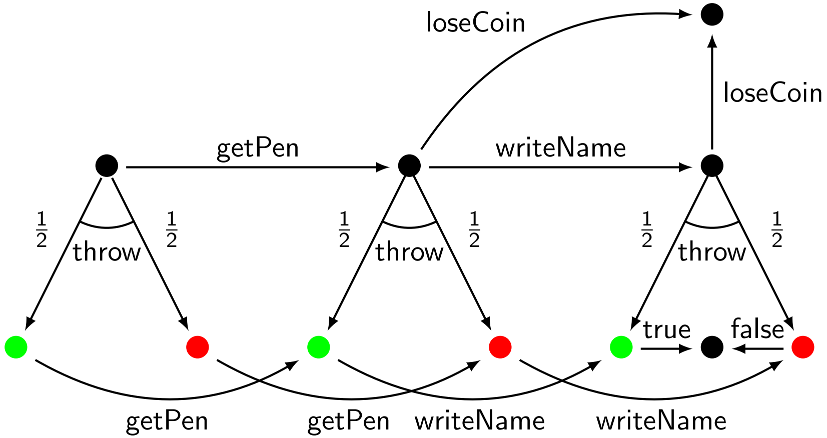
Confluence: an introductory example



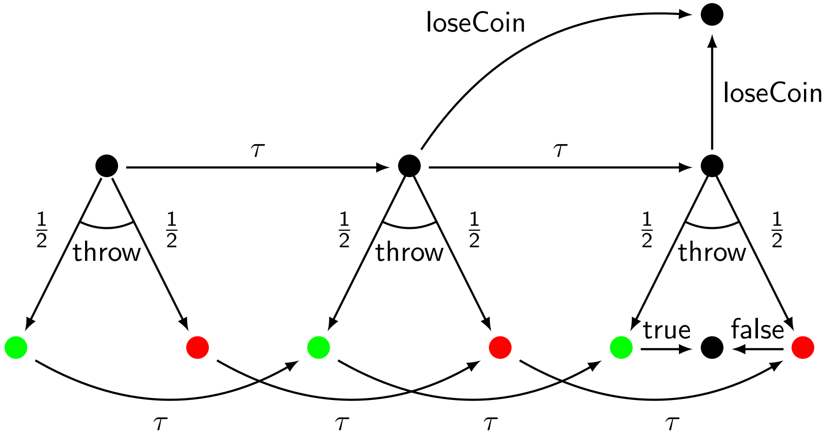
Confluence: an introductory example



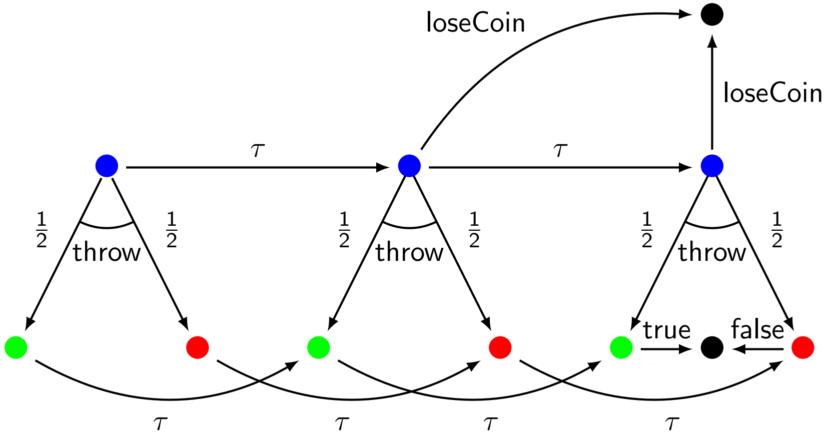
Confluence: an introductory example



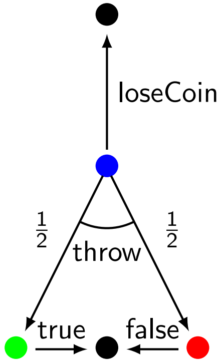
Confluence: an introductory example



Confluence: an introductory example



Confluence: an introductory example



Confluence: non-probabilistic versus probabilistic

Three notions of confluence:

- weak confluence
- confluence
- strong confluence

Confluence: non-probabilistic versus probabilistic

Three notions of confluence:

• weak confluence		• weak probabilistic confluence
• confluence	$\Rightarrow$	• probabilistic confluence
• strong confluence		• strong probabilistic confluence

Confluence: non-probabilistic versus probabilistic

Three notions of confluence:

• weak confluence		• weak probabilistic confluence
• confluence	$\Rightarrow$	• probabilistic confluence
• strong confluence		• strong probabilistic confluence

Confluence: non-probabilistic versus probabilistic

Three notions of confluence:

- weak confluence
- confluence
- strong confluence

$\Rightarrow$

- weak probabilistic confluence
- probabilistic confluence
- strong probabilistic confluence

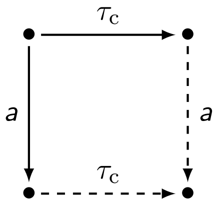
$$\begin{tikzcd}
\bullet \arrow[r, "\tau_{\mathrm{c}}"] \arrow[d, "a"'] & \bullet \arrow[d, dashed, "a"] \\
\bullet \arrow[r, dashed, "\tau_{\mathrm{c}}"] & \bullet
\end{tikzcd}$$

Strong confluence

Confluence: non-probabilistic versus probabilistic

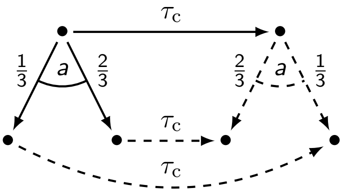
Three notions of confluence:

weak confluence		weak probabilistic confluence
confluence	$\Rightarrow$	probabilistic confluence
strong confluence		strong probabilistic confluence



$$\begin{tikzcd}
\bullet \arrow[r, "\tau_c"] \arrow[d, "a"'] & \bullet \arrow[d, dashed, "a"] \\
\bullet \arrow[r, dashed, "\tau_c"] & \bullet
\end{tikzcd}$$

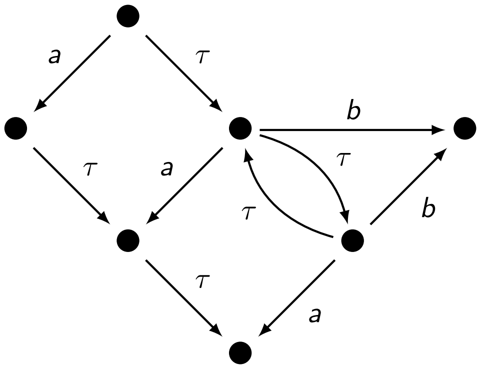
Strong confluence



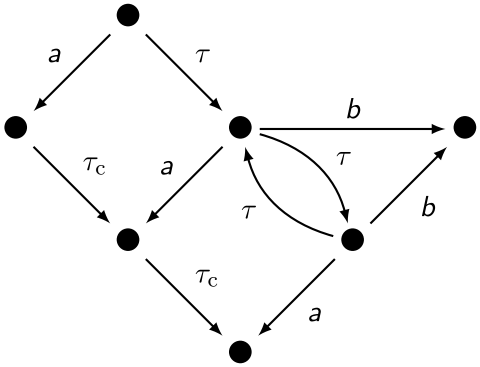
Strong probabilistic confluence

Table of Contents

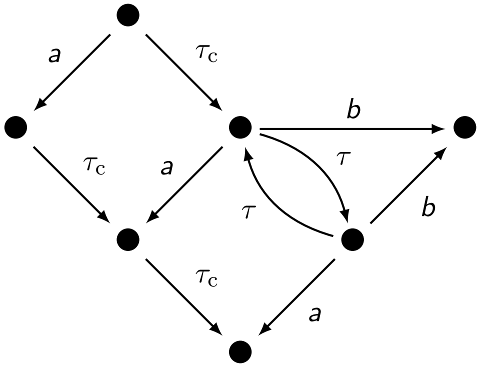
State space reduction using confluence



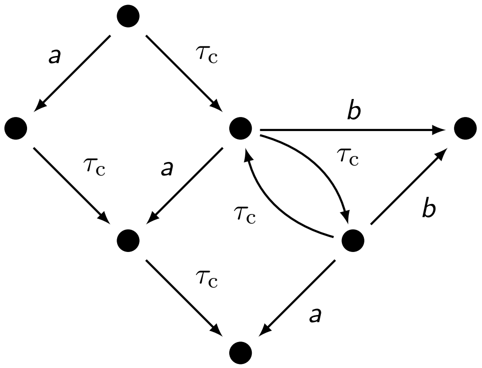
State space reduction using confluence



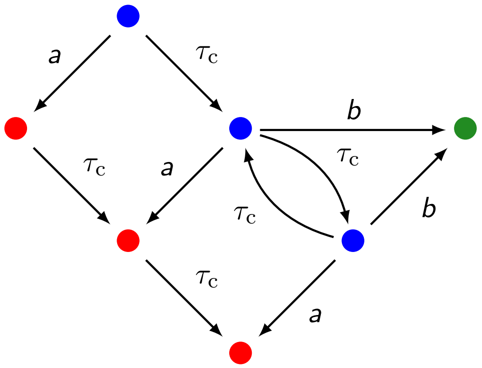
State space reduction using confluence



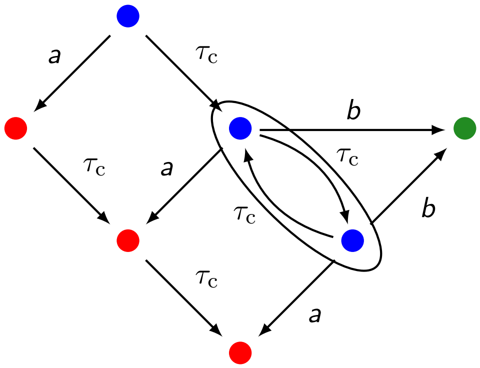
State space reduction using confluence



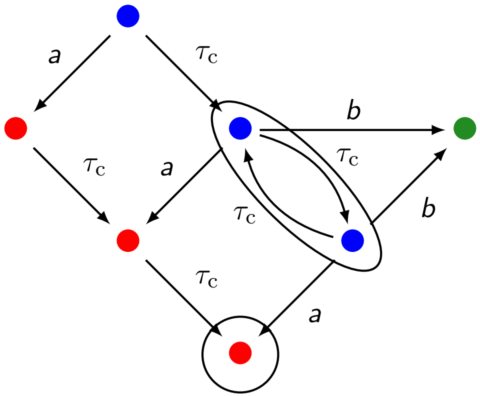
State space reduction using confluence



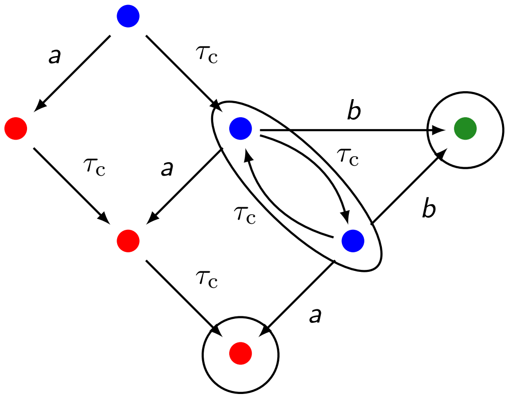
State space reduction using confluence



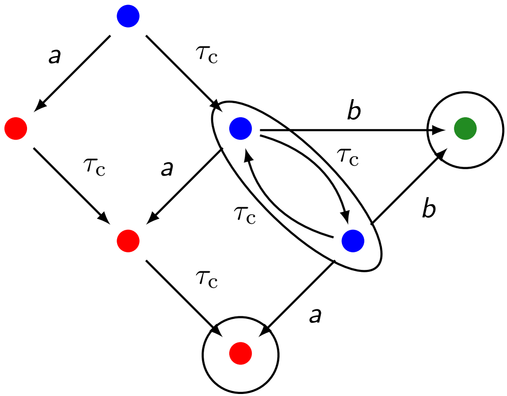
State space reduction using confluence



State space reduction using confluence



State space reduction using confluence



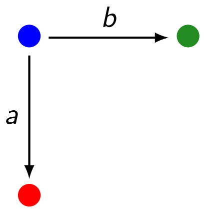


Table of Contents

Detecting confluence: LPPEs

We detect confluence symbolically using LPPEs:

$$
\begin{aligned}
X(\vec{g} : \vec{G}) = & \sum_{\vec{d_1}:\vec{D_1}} c_1 \Rightarrow a_1 \sum\!\!\!\!\bullet_{\vec{e_1}:\vec{E_1}} f_1 : X(\vec{n_1}) \\
& \dots \\
+ & \sum_{\vec{d_k}:\vec{D_k}} c_k \Rightarrow a_k \sum\!\!\!\!\bullet_{\vec{e_k}:\vec{E_k}} f_k : X(\vec{n_k})
\end{aligned}
$$

Detecting confluence: LPPEs

We detect confluence symbolically using LPPEs:

$$\begin{aligned} X(\vec{g} : \vec{G}) = &\sum_{\vec{d_1}:\vec{D_1}} c_1 \Rightarrow a_1 \sum\!\!\!\bullet_{\vec{e_1}:\vec{E_1}} f_1 : X(\vec{n_1}) \\ &\dots \\ + &\sum_{\vec{d_k}:\vec{D_k}} c_k \Rightarrow a_k \sum\!\!\!\bullet_{\vec{e_k}:\vec{E_k}} f_k : X(\vec{n_k}) \end{aligned}$$

Example of an LPPE

$$\begin{aligned} &X(\mathrm{pc} : \{1..2\}, \mathrm{active} : \mathrm{Bool}) = \\ &\quad \sum_{n:\{1,2,3\}} \mathrm{pc} = 1 \Rightarrow \mathrm{output}(n) \sum\!\!\!\bullet_{\mathrm{b:Bool}} \tfrac{1}{2} : X(2, \mathrm{b}) \\ &+ \quad \mathrm{pc} = 2 \wedge \mathrm{active} \Rightarrow \mathrm{beep} \cdot X(1, \mathrm{active}) \end{aligned}$$

Detecting confluence: LPPEs

We detect confluence symbolically using LPPEs:

$$X(\vec{g} : \vec{G}) = \sum_{\vec{d_1}:\vec{D_1}} c_1 \Rightarrow a_1 \sum\!\!\!\bullet_{\vec{e_1}:\vec{E_1}} f_1 : X(\vec{n_1})$$
$$\cdots$$
$$+ \sum_{\vec{d_k}:\vec{D_k}} c_k \Rightarrow a_k \sum\!\!\!\bullet_{\vec{e_k}:\vec{E_k}} f_k : X(\vec{n_k})$$

Example of an LPPE

$$X(\mathrm{pc} : \{1..2\}, \mathrm{active} : \mathrm{Bool}) =$$
$$\sum_{n:\{1,2,3\}} \mathrm{pc} = 1 \Rightarrow \mathrm{output}(n) \sum\!\!\!\bullet_{\mathrm{b:Bool}} \tfrac{1}{2} : X(2, \mathrm{b})$$
$$+ \quad \mathrm{pc} = 2 \wedge \mathrm{active} \Rightarrow \tau \cdot X(1, \mathrm{active})$$

Detecting confluence: LPPEs

Example of an LPPE

$$
\begin{aligned}
& X(\mathsf{pc} : \{1..2\}, \mathsf{active} : \mathsf{Bool}) = \\
& \qquad \sum_{n:\{1,2,3\}} \mathsf{pc} = 1 \qquad \Rightarrow \mathsf{output}(n) \sum_{\mathsf{b}:\mathsf{Bool}}\!\!\bullet\ \tfrac{1}{2} : X(2, \mathsf{b}) \\
& \quad + \qquad \mathsf{pc} = 2 \wedge \mathsf{active} \Rightarrow \tau \cdot X(1, \mathsf{active})
\end{aligned}
$$

How to know whether a summand is confluent?

Detecting confluence: LPPEs

Example of an LPPE

$$
\begin{aligned}
& X(\mathsf{pc} : \{1..2\}, \mathsf{active} : \mathsf{Bool}) = \\
& \qquad \sum_{n:\{1,2,3\}} \mathsf{pc} = 1 \qquad \Rightarrow \mathsf{output}(n) \textstyle\sum\!\!\!\bullet_{\mathsf{b:Bool}} \frac{1}{2} : X(2, \mathsf{b}) \\
& \quad + \qquad \mathsf{pc} = 2 \wedge \mathsf{active} \Rightarrow \tau \cdot X(1, \mathsf{active})
\end{aligned}
$$

How to know whether a summand is confluent?

- Its action should be τ

Detecting confluence: LPPEs

Example of an LPPE

$$
\begin{aligned}
&X(\mathsf{pc} : \{1..2\}, \mathsf{active} : \mathsf{Bool}) = \\
&\qquad \sum_{n:\{1,2,3\}} \mathsf{pc} = 1 \quad \Rightarrow \mathsf{output}(n) \sum_{\mathsf{b:Bool}}\!\!\bullet\ \tfrac{1}{2} : X(2, \mathsf{b}) \\
&\quad + \qquad \mathsf{pc} = 2 \wedge \mathsf{active} \Rightarrow \tau \cdot X(1, \mathsf{active})
\end{aligned}
$$

How to know whether a summand is confluent?

- Its action should be τ
- Its next state should be chosen nonprobabilistically

Detecting confluence: LPPEs

Example of an LPPE

$$
\begin{aligned}
& X(\mathsf{pc} : \{1..2\}, \mathsf{active} : \mathsf{Bool}) = \\
& \qquad \sum_{n:\{1,2,3\}} \mathsf{pc} = 1 \qquad \Rightarrow \mathsf{output}(n) \sum\!\!\bullet_{\mathsf{b:Bool}} \tfrac{1}{2} : X(2, \mathsf{b}) \\
& + \qquad\qquad \mathsf{pc} = 2 \wedge \mathsf{active} \Rightarrow \tau \cdot X(1, \mathsf{active})
\end{aligned}
$$

How to know whether a summand is confluent?

- Its action should be τ
- Its next state should be chosen nonprobabilistically
- It should commute with all the other summands

Detecting confluence: LPPEs

Example of an LPPE

$$
\begin{aligned}
&X(\mathrm{pc} : \{1..2\}, \mathrm{active} : \mathrm{Bool}) = \\
&\quad \sum_{n:\{1,2,3\}} \mathrm{pc} = 1 \Rightarrow \mathrm{output}(n) \sum_{\mathrm{b:Bool}}\!\!\bullet\; \tfrac{1}{2} : X(2, \mathrm{b}) \\
&+ \quad \mathrm{pc} = 2 \wedge \mathrm{active} \Rightarrow \tau \cdot X(1, \mathrm{active})
\end{aligned}
$$

How to know whether a summand is confluent?

- Its action should be τ
- Its next state should be chosen nonprobabilistically
- It should commute with all the other summands
    - Never enabled at the same time
    - Not touching the same variables

Table of Contents

Case study: leader election protocol

	Original		Reduced		Running time	
Specification	States	Trans.	States	Trans.	Before	After
`leader`	3763	6158	1399	1922	1.86 sec	0.72 sec
`leaderReduced`	1693	2438	589	722	0.90 sec	0.44 sec
`leader-2-2`	67	94	27	32	0.04 sec	0.65 sec
`leader-2-6`	535	710	199	212	0.36 sec	0.81 sec
`leader-2-36`	18325	23690	6589	6662	516.23 sec	43.11 sec
`leader-3-2`	1018	1815	376	561	1.61 sec	3.81 sec
`leader-3-6`	21664	36519	7936	10233	221.22 sec	44.92 sec

−60% −70%

Case study: leader election protocol

	Original		Reduced		Running time	
Specification	States	Trans.	States	Trans.	Before	After
`leader`	3763	6158	1399	1922	1.86 sec	0.72 sec
`leaderReduced`	1693	2438	589	722	0.90 sec	0.44 sec
`leader-2-2`	67	94	27	32	0.04 sec	0.65 sec
`leader-2-6`	535	710	199	212	0.36 sec	0.81 sec
`leader-2-36`	18325	23690	6589	6662	516.23 sec	43.11 sec
`leader-3-2`	1018	1815	376	561	1.61 sec	3.81 sec
`leader-3-6`	21664	36519	7936	10233	221.22 sec	44.92 sec

-60% -70%

Table of Contents

Conclusions

Related work

- Confluence reduction for non-probabilistic systems
- Partial-order reduction for probabilistic systems

Conclusions

Related work

- Confluence reduction for non-probabilistic systems
- Partial-order reduction for probabilistic systems

Conclusions / results

- We developed three new notions of confluence for probabilistic automata that preserve branching probabilistic bisimulation;
- We showed how confluence can be used for state space reduction;
- We discussed how to detect confluence based on a probabilistic process algebra;
- We illustrated the power of our method using a case study.

Questions

Questions?