UNIVERSITY OF TWENTE.

Formal Methods & Tools.

Confluence Reduction for Markov Automata

Mark Timmer

March 23, 2013

Joint work with
Jaco van de Pol and Mariëlle Stoelinga

QAPL 2013

The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism ⟵ LTSs
- Probability ⟵ DTMCs
- Stochastic timing ⟵ CTMCs

The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

Probabilistic Automata (PAs) → Nondeterminism, Probability

The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

Interactive Markov Chains (IMCs)

The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

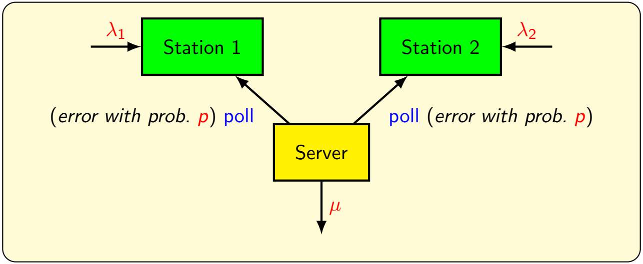
Markov Automata (MAs)

The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

Markov Automata (MAs)

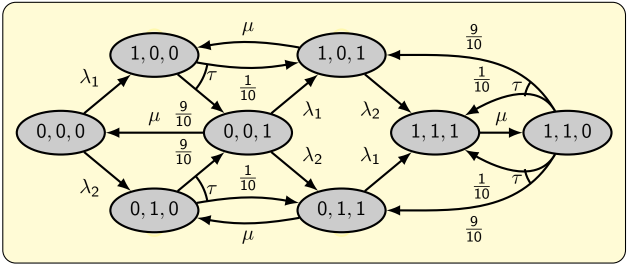


The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

Markov Automata (MAs)

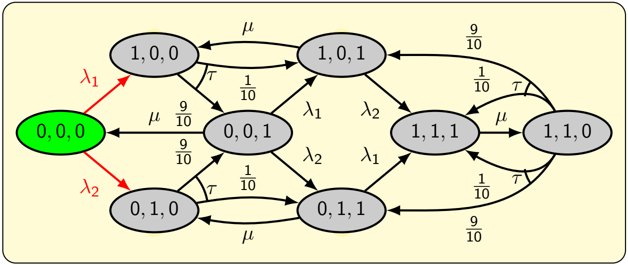


The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

← Markov Automata (MAs)

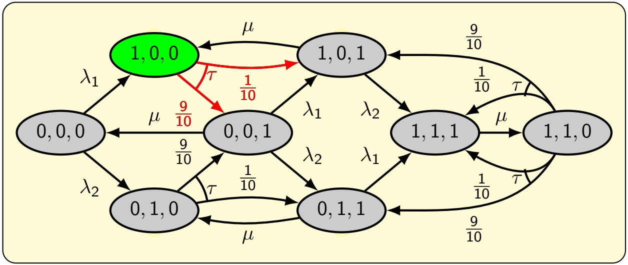


The overall goal: efficient and expressive modelling

Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

← Markov Automata (MAs)

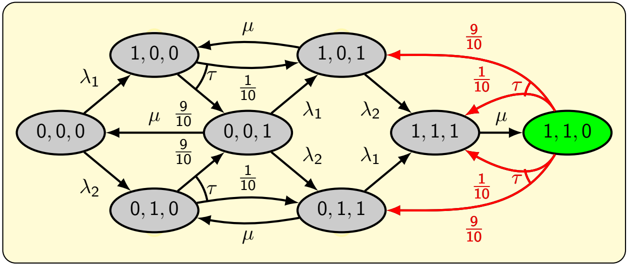


The overall goal: efficient and expressive modelling

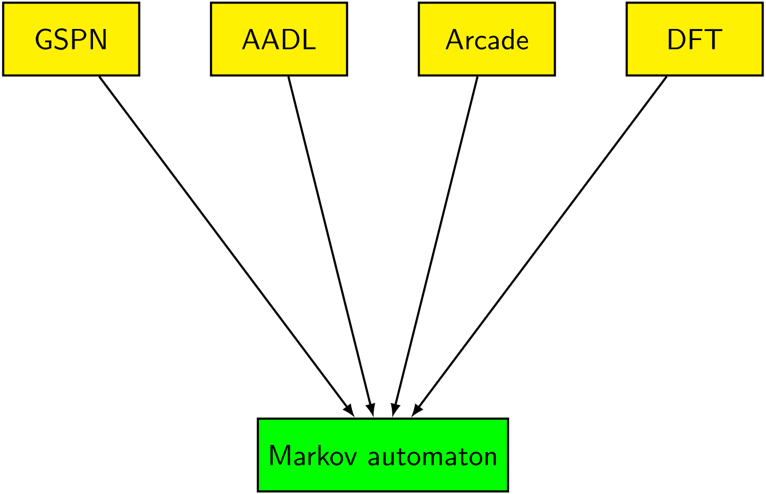
Specifying systems with

- Nondeterminism
- Probability
- Stochastic timing

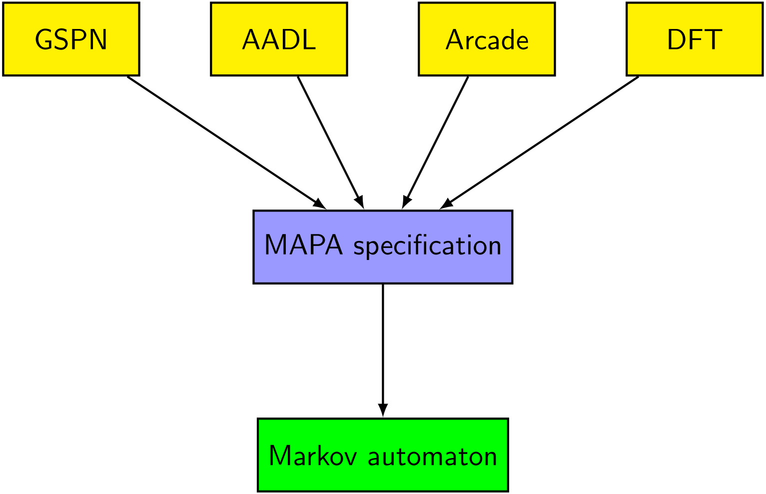
Markov Automata (MAs)



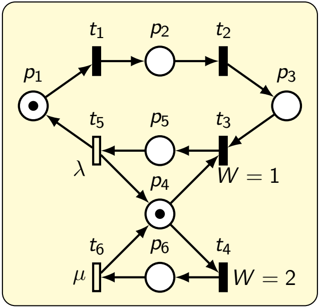
Higher-level formalisms that can be mapped to MAs



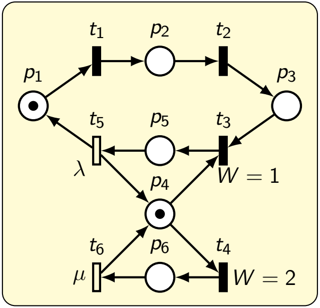
Higher-level formalisms that can be mapped to MAs



Higher-level formalisms mapped to MAs



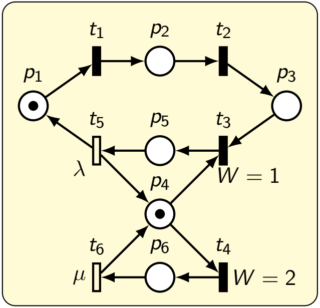
Higher-level formalisms mapped to MAs



MAPA specification

$$
\begin{aligned}
&\mathit{System}(P_1 : \mathbb{N}, P_2 : \mathbb{N}, P_3 : \mathbb{N}, P_4 : \mathbb{N}, P_5 : \mathbb{N}, P_6 : \mathbb{N}) = \\
&\qquad P_1 \geq 1 \implies \tau \cdot \text{System}(P_1 - 1, P_2 + 1, P_3, P_4, P_5, P_6) \\
&\qquad + P_2 \geq 1 \implies \tau \cdot \text{System}(P_1, P_2 - 1, P_3 + 1, P_4, P_5, P_6) \\
&\qquad + P_5 \geq 1 \implies \lambda \cdot \text{System}(P_1 + 1, P_2, P_3, P_4 + 1, P_5 - 1, P_6) \\
&\qquad + P_6 \geq 1 \implies \mu \cdot \text{System}(P_1, P_2, P_3, P_4 + 1, P_5, P_6 - 1) \\
&\qquad + (P_3 \geq 1 \land P_4 \geq 1) \lor (P_4 \geq 1) \implies \tau \cdot \ldots
\end{aligned}
$$

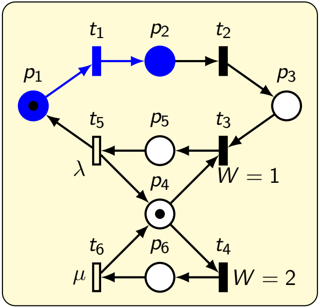
Higher-level formalisms mapped to MAs



MAPA specification

$$\begin{aligned} System(P_1 : \mathbb{N}, P_2 : \mathbb{N}, P_3 : \mathbb{N}, P_4 : \mathbb{N}, P_5 : \mathbb{N}, P_6 : \mathbb{N}) = \\ P_1 \geq 1 \implies \tau \cdot \text{System}(P_1 - 1, P_2 + 1, P_3, P_4, P_5, P_6) \\ + P_2 \geq 1 \implies \tau \cdot \text{System}(P_1, P_2 - 1, P_3 + 1, P_4, P_5, P_6) \\ + P_5 \geq 1 \implies \lambda \cdot \text{System}(P_1 + 1, P_2, P_3, P_4 + 1, P_5 - 1, P_6) \\ + P_6 \geq 1 \implies \mu \cdot \text{System}(P_1, P_2, P_3, P_4 + 1, P_5, P_6 - 1) \\ + (P_3 \geq 1 \wedge P_4 \geq 1) \vee (P_4 \geq 1) \implies \tau \cdot \ldots \end{aligned}$$

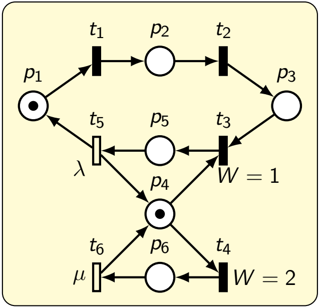
Higher-level formalisms mapped to MAs

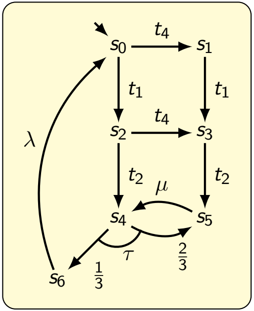


MAPA specification

$$
\begin{aligned}
&\mathit{System}(P_1 : \mathbb{N}, P_2 : \mathbb{N}, P_3 : \mathbb{N}, P_4 : \mathbb{N}, P_5 : \mathbb{N}, P_6 : \mathbb{N}) = \\
&\quad P_1 \geq 1 \implies \tau \cdot \text{System}(P_1 - 1, P_2 + 1, P_3, P_4, P_5, P_6) \\
&\quad + P_2 \geq 1 \implies \tau \cdot \text{System}(P_1, P_2 - 1, P_3 + 1, P_4, P_5, P_6) \\
&\quad + P_5 \geq 1 \implies \lambda \cdot \text{System}(P_1 + 1, P_2, P_3, P_4 + 1, P_5 - 1, P_6) \\
&\quad + P_6 \geq 1 \implies \mu \cdot \text{System}(P_1, P_2, P_3, P_4 + 1, P_5, P_6 - 1) \\
&\quad + (P_3 \geq 1 \land P_4 \geq 1) \lor (P_4 \geq 1) \implies \tau \cdot \ldots
\end{aligned}
$$

Higher-level formalisms mapped to MAs

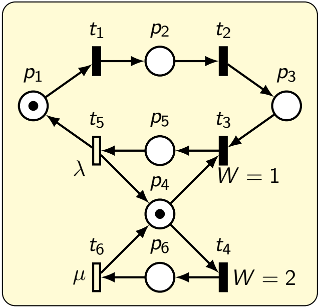


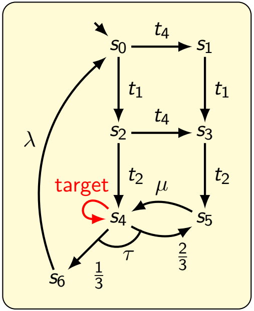


MAPA specification

$$
\begin{aligned}
\mathit{System}(P_1 : \mathbb{N}, P_2 : \mathbb{N}, P_3 : \mathbb{N}, P_4 : \mathbb{N}, P_5 : \mathbb{N}, P_6 : \mathbb{N}) = \\
P_1 \geq 1 &\implies \tau \cdot \text{System}(P_1 - 1, P_2 + 1, P_3, P_4, P_5, P_6) \\
+ P_2 \geq 1 &\implies \tau \cdot \text{System}(P_1, P_2 - 1, P_3 + 1, P_4, P_5, P_6) \\
+ P_5 \geq 1 &\implies \lambda \cdot \text{System}(P_1 + 1, P_2, P_3, P_4 + 1, P_5 - 1, P_6) \\
+ P_6 \geq 1 &\implies \mu \cdot \text{System}(P_1, P_2, P_3, P_4 + 1, P_5, P_6 - 1) \\
+ (P_3 \geq 1 \wedge P_4 \geq 1) \vee (P_4 \geq 1) &\implies \tau \cdot \ldots
\end{aligned}
$$

Higher-level formalisms mapped to MAs

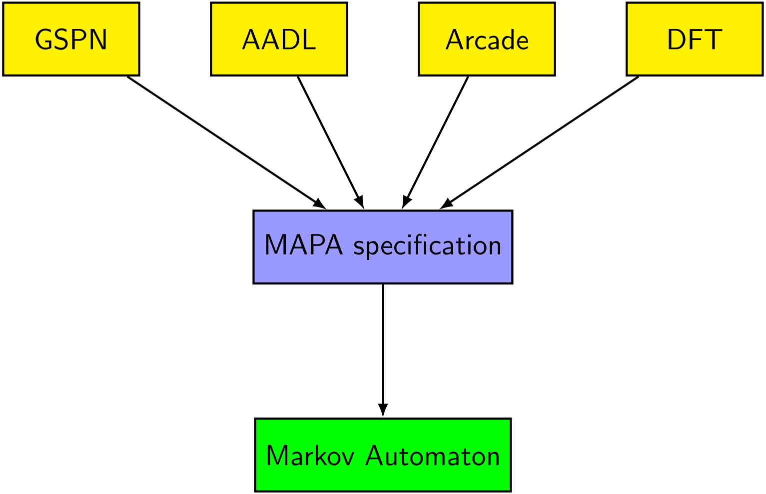




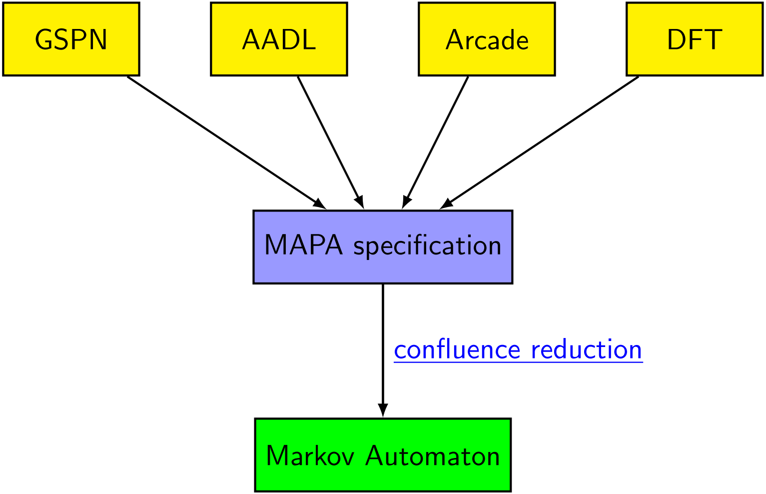
MAPA specification

$$
\begin{aligned}
&\mathit{System}(P_1 : \mathbb{N}, P_2 : \mathbb{N}, P_3 : \mathbb{N}, P_4 : \mathbb{N}, P_5 : \mathbb{N}, P_6 : \mathbb{N}) = \\
&\quad P_1 \geq 1 \implies \tau \cdot \text{System}(P_1 - 1, P_2 + 1, P_3, P_4, P_5, P_6) \\
&\quad + P_2 \geq 1 \implies \tau \cdot \text{System}(P_1, P_2 - 1, P_3 + 1, P_4, P_5, P_6) \\
&\quad + P_5 \geq 1 \implies \lambda \cdot \text{System}(P_1 + 1, P_2, P_3, P_4 + 1, P_5 - 1, P_6) \\
&\quad + P_6 \geq 1 \implies \mu \cdot \text{System}(P_1, P_2, P_3, P_4 + 1, P_5, P_6 - 1) \\
&\quad + (P_3 \geq 1 \wedge P_4 \geq 1) \vee (P_4 \geq 1) \implies \tau \cdot \ldots
\end{aligned}
$$

Higher-level formalisms mapped to MAs



Higher-level formalisms mapped to MAs



Contents

Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

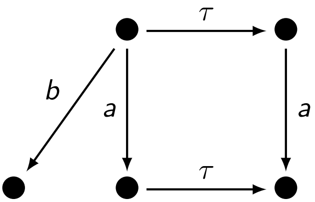
- Labelled by τ
- Deterministic

Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

- Labelled by τ
- Deterministic

Deterministic τ-steps might disable behaviour. . .

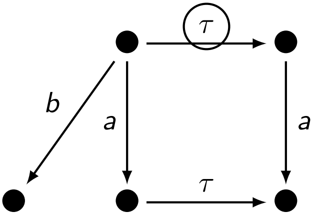


Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

- Labelled by τ
- Deterministic

Deterministic τ-steps might disable behaviour...



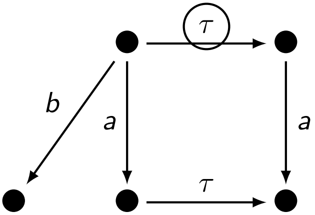
Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

- Labelled by τ
- Deterministic

Deterministic τ-steps might disable behaviour...

...though often, they connect equivalent states



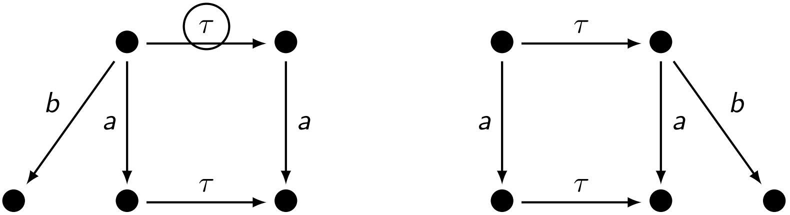
Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

- Labelled by τ
- Deterministic

Deterministic τ-steps might disable behaviour...

...though often, they connect equivalent states



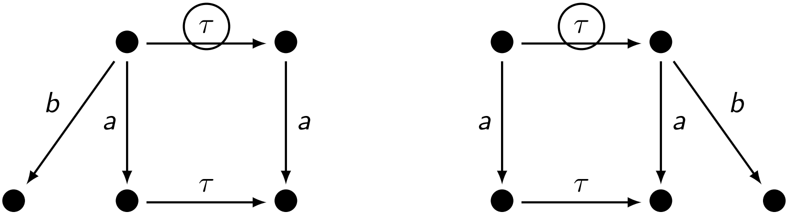
Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

- Labelled by τ
- Deterministic

Deterministic τ-steps might disable behaviour...

...though often, they connect equivalent states



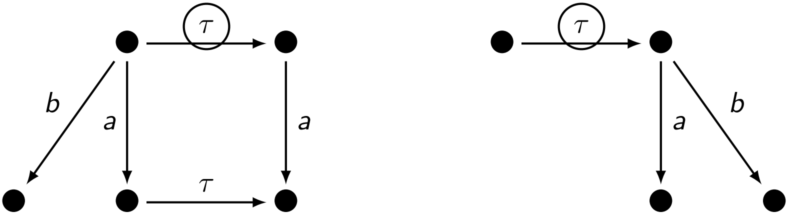
Invisible transitions connecting equivalent states

Invisible transitions in confluence reduction:

- Labelled by τ
- Deterministic

Deterministic τ-steps might disable behaviour...

...though often, they connect equivalent states



Non-probabilistic and probabilistic confluence reduction

Confluence reduction:

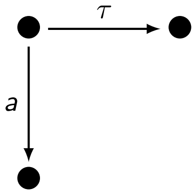
denoting a subset of the invisible transitions as confluent.

Non-probabilistic and probabilistic confluence reduction

Confluence reduction:

denoting a subset of the invisible transitions as confluent.

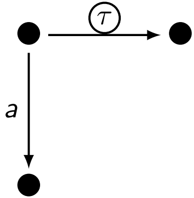
Non-probabilistically:



Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

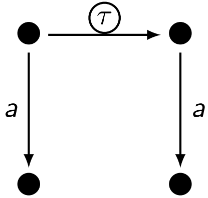
Non-probabilistically:



Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

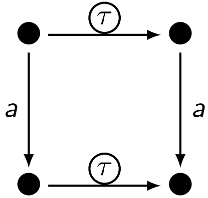
Non-probabilistically:



Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

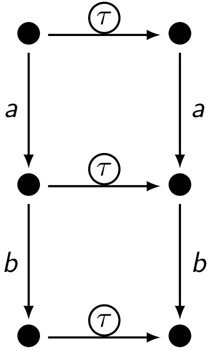
Non-probabilistically:



Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

Non-probabilistically:

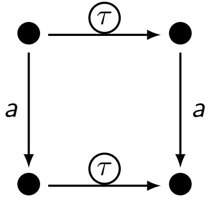


Non-probabilistic and probabilistic confluence reduction

Confluence reduction:

denoting a subset of the invisible transitions as confluent.

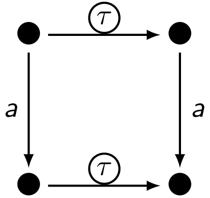
Non-probabilistically:



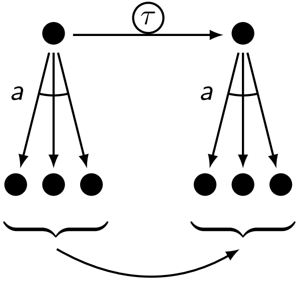
Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

Non-probabilistically:



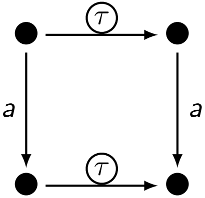
Probabilistically:



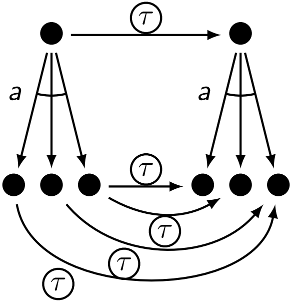
Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

Non-probabilistically:



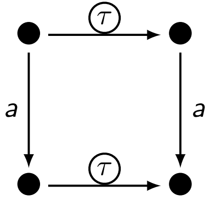
Probabilistically:



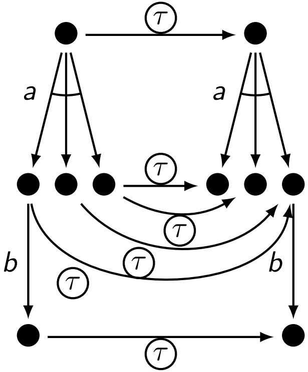
Non-probabilistic and probabilistic confluence reduction

Confluence reduction:
denoting a subset of the invisible transitions as confluent.

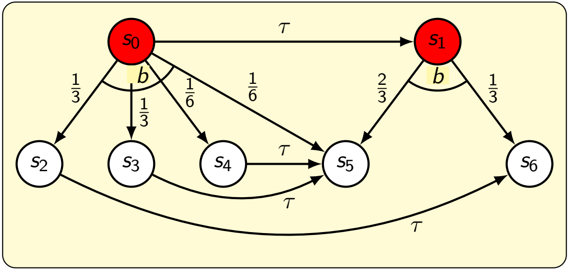
Non-probabilistically:



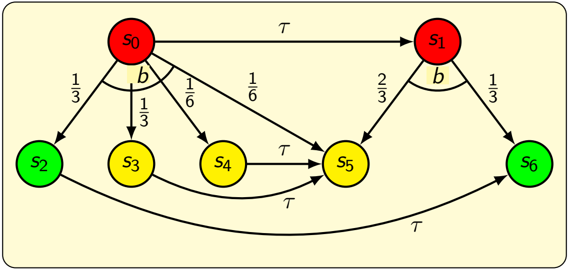
Probabilistically:



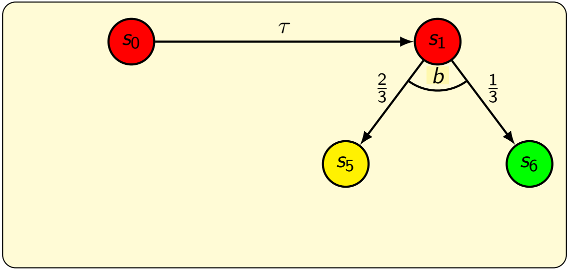
Probabilistic example



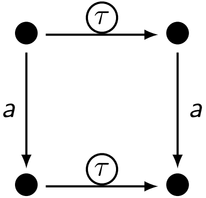
Probabilistic example



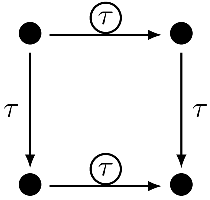
Probabilistic example



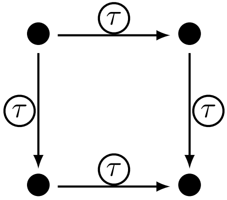
Problem with earlier definitions: no closure under union



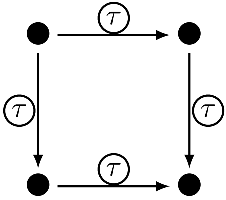
Problem with earlier definitions: no closure under union

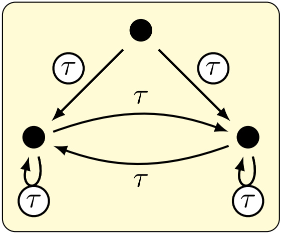


Problem with earlier definitions: no closure under union

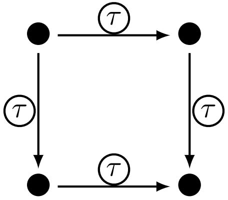


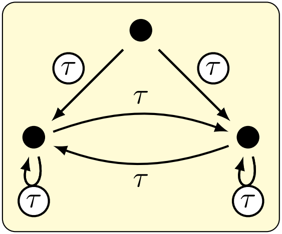
Problem with earlier definitions: no closure under union

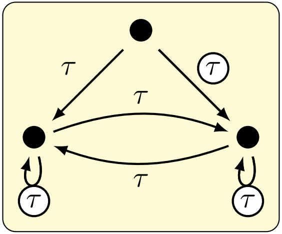




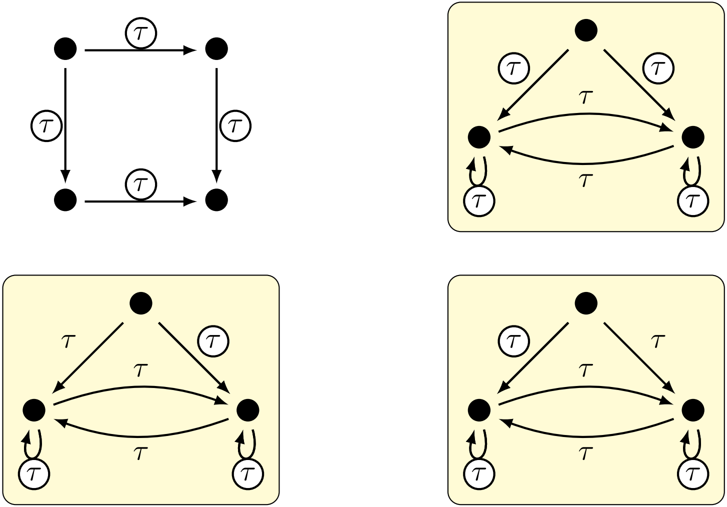
Problem with earlier definitions: no closure under union



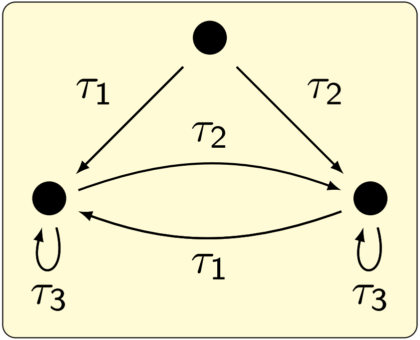




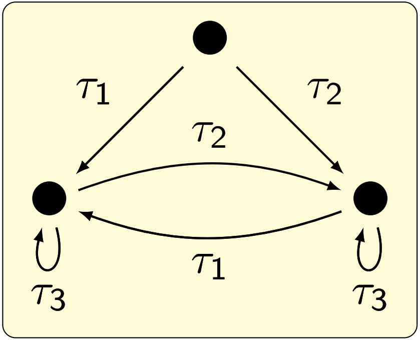
Problem with earlier definitions: no closure under union



Our solution: confluence classification

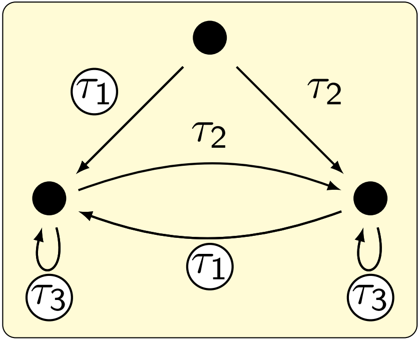


Our solution: confluence classification



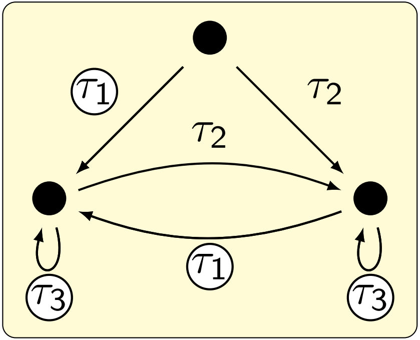
- Mimicking always by a transition from the same group
- For each group, either all transitions or no transitions are confluent

Our solution: confluence classification



- Mimicking always by a transition from the same group
- For each group, either all transitions or no transitions are confluent

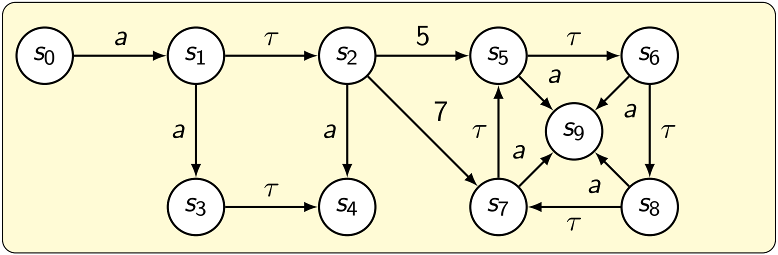
Our solution: confluence classification



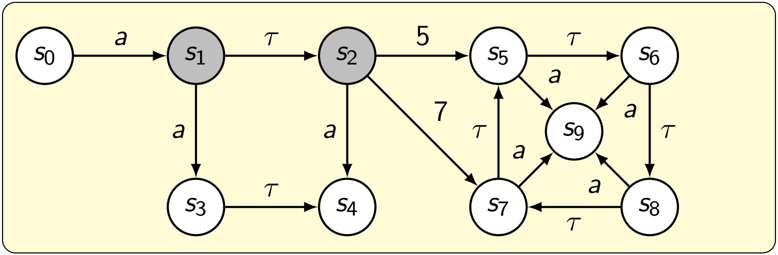
- Mimicking always by a transition from the same group
- For each group, either all transitions or no transitions are confluent

Closure under unions is now really ensured.

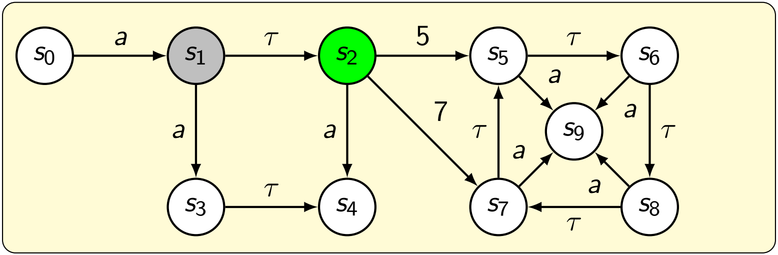
Representative map approach



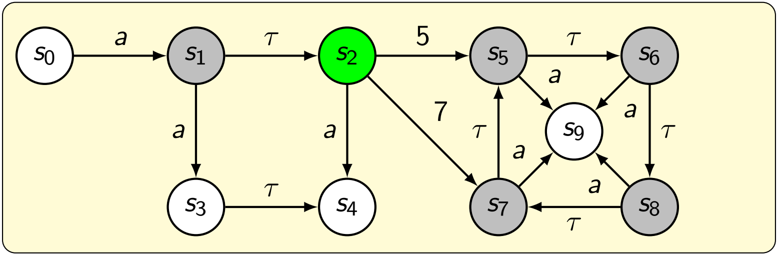
Representative map approach



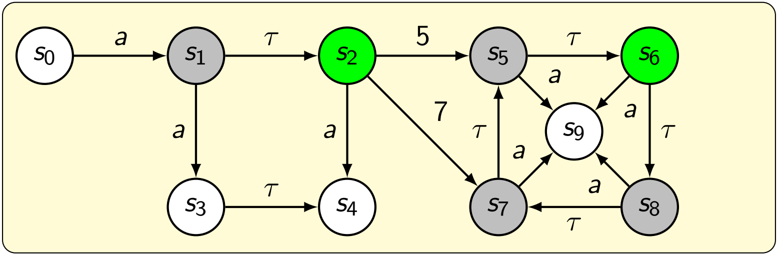
Representative map approach



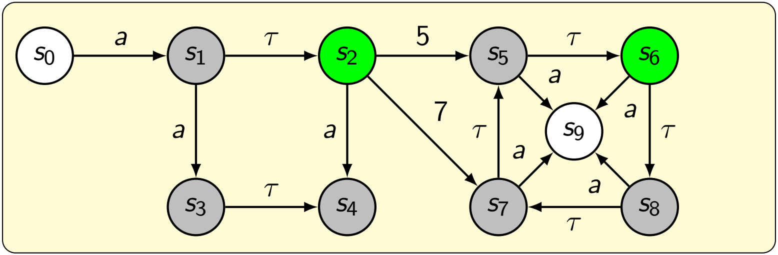
Representative map approach



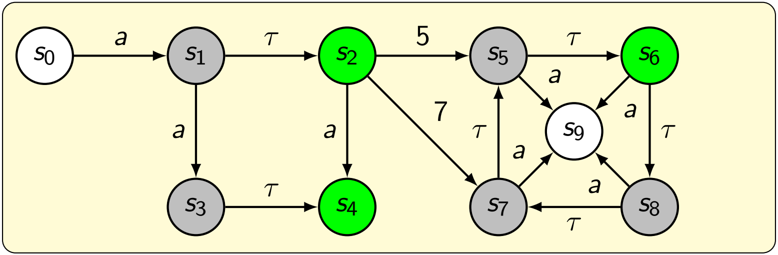
Representative map approach



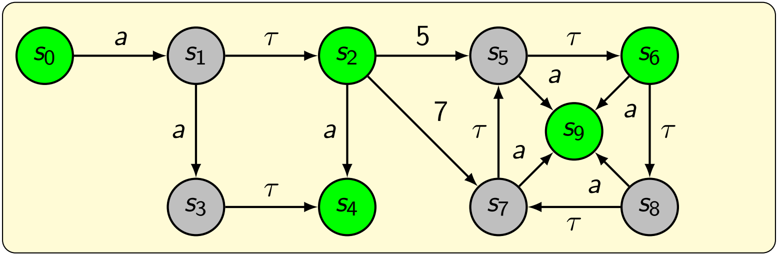
Representative map approach



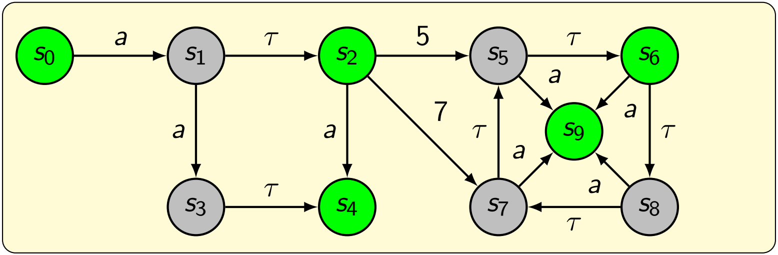
Representative map approach

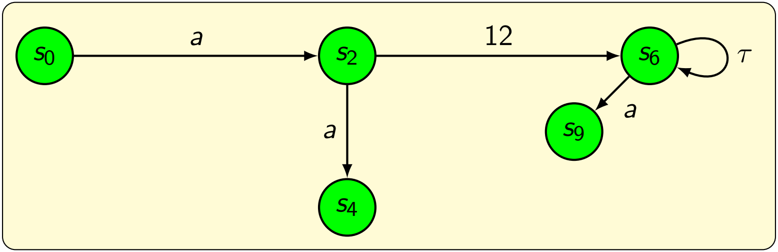


Representative map approach

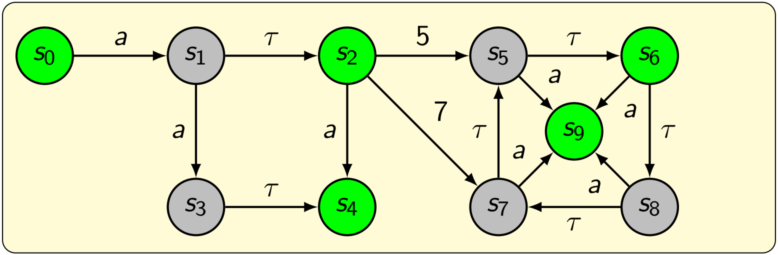


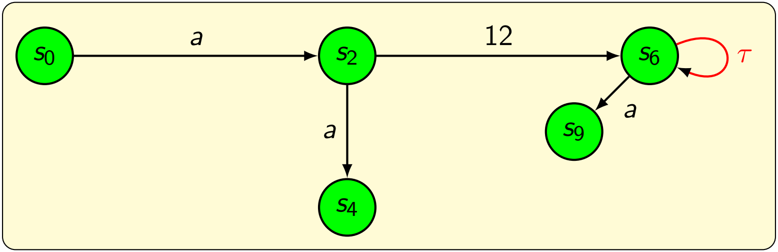
Representative map approach





Representative map approach





A process algebra for Markov automata: MAPA

Specification language MAPA:

- Based on μCRL (so data), with additional probabilistic choice and Markovian rates
- Semantics defined in terms of Markov automata
- Minimal set of operators to facilitate formal manipulation
- Syntactic sugar easily definable

A process algebra for Markov automata: MAPA

Specification language MAPA:

- Based on μCRL (so data), with additional probabilistic choice and Markovian rates
- Semantics defined in terms of Markov automata
- Minimal set of operators to facilitate formal manipulation
- Syntactic sugar easily definable

Operators

$$p ::= Y(t) \mid c \Rightarrow p \mid p + p \mid \sum_{x:D} p \mid a(t) \sum\!\!\!\bullet_{x:D} f : p \mid (\lambda) \cdot p$$

A process algebra for Markov automata: MAPA

Specification language MAPA:

- Based on μCRL (so data), with additional probabilistic choice and Markovian rates
- Semantics defined in terms of Markov automata
- Minimal set of operators to facilitate formal manipulation
- Syntactic sugar easily definable

Operators

$$p ::= Y(t) \;\mid\; c \Rightarrow p \;\mid\; p + p \;\mid\; \sum_{x:D} p \;\mid\; a(t) \sum_{x:D}\!\!\bullet\; f : p \;\mid\; (\lambda) \cdot p$$

- Composibility via parallel composition, encapsulation, hiding and renaming

MLPPEs

We defined a special format for MAPA, the MLPPE:

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{i \in I} \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow a_i(\boldsymbol{b_i}) \sum\!\!\!\!\!\bullet_{\boldsymbol{e_i} : \boldsymbol{E_i}} f_i : X(\boldsymbol{n_i})$$

$$+ \sum_{j \in J} \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow (\lambda_j) \cdot X(\boldsymbol{n_j})$$

Example of an MLPPE

GSPN-generated MAPA specification

$$
\begin{aligned}
&\mathit{System}(P_1 : \mathbb{N}, P_2 : \mathbb{N}, P_3 : \mathbb{N}, P_4 : \mathbb{N}, P_5 : \mathbb{N}, P_6 : \mathbb{N}) = \\
&\qquad P_1 \geq 1 \implies \tau \cdot \text{System}(P_1 - 1, P_2 + 1, P_3, P_4, P_5, P_6) \\
&\quad + P_2 \geq 1 \implies \tau \cdot \text{System}(P_1, P_2 - 1, P_3 + 1, P_4, P_5, P_6) \\
&\quad + P_5 \geq 1 \implies \lambda \cdot \text{System}(P_1 + 1, P_2, P_3, P_4 + 1, P_5 - 1, P_6) \\
&\quad + P_6 \geq 1 \implies \mu \cdot \text{System}(P_1, P_2, P_3, P_4 + 1, P_5, P_6 - 1) \\
&\quad + (P_3 \geq 1 \wedge P_4 \geq 1) \vee (P_4 \geq 1) \implies \tau \sum_{i:\{4,5\}} f : \\
&\qquad \text{System}(P_1, P_2, \text{if } i = 4 \text{ then } P_3 - 1 \text{ else } P_3, P_4 - 1, \\
&\qquad\qquad \text{if } i = 4 \text{ then } P_5 + 1 \text{ else } P_5, \\
&\qquad\qquad \text{if } i = 4 \text{ then } P_6 \text{ else } P_6 + 1)
\end{aligned}
$$

MLPPEs

We defined a special format for MAPA, the MLPPE:

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{i \in I} \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow a_i(\boldsymbol{b_i}) \sum_{\boldsymbol{e_i} : \boldsymbol{E_i}}\!\!\!\bullet\; f_i : X(\boldsymbol{n_i})$$
$$+ \sum_{j \in J} \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow (\lambda_j) \cdot X(\boldsymbol{n_j})$$

Advantages of using MLPPEs instead of MAPA specifications:

- Easy state space generation
- Straight-forward parallel composition
- **Symbolic optimisations enabled at the language level**

MLPPEs

We defined a special format for MAPA, the MLPPE:

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{i \in I} \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow a_i(\boldsymbol{b_i}) \sum\!\!\!\!\!\bullet_{\boldsymbol{e_i} : \boldsymbol{E_i}} f_i : X(\boldsymbol{n_i})$$
$$+ \sum_{j \in J} \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow (\lambda_j) \cdot X(\boldsymbol{n_j})$$

Advantages of using MLPPEs instead of MAPA specifications:

- Easy state space generation
- Straight-forward parallel composition
- **Symbolic optimisations enabled at the language level**

Theorem

Every specification (without unguarded recursion) can be linearised to an MLPPE, preserving strong bisimulation.

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

How to know whether a summand is confluent?

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

How to know whether a summand is confluent?

- Its action should be τ

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

How to know whether a summand is confluent?

- Its action should be τ
- Its next state should be chosen nonprobabilistically (*heuristic:* there is no probabilistic choice)

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

How to know whether a summand is confluent?

- Its action should be τ
- Its next state should be chosen nonprobabilistically (*heuristic:* there is no probabilistic choice)
- It should commute with all the other summands

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

How to know whether a summand is confluent?

- Its action should be τ
- Its next state should be chosen nonprobabilistically (*heuristic:* there is no probabilistic choice)
- It should commute with all the other interactive summands

Detecting confluence symbolically on MLPPEs

Symbolic detection of confluence: denote entire summands to be confluent (i.e., all their concrete transitions are confluent)

- Each summand is a group in the confluence classification
- Underapproximation of the actual confluent transitions

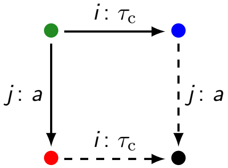
How to know whether a summand is confluent?

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

  (*heuristic:* there is no probabilistic choice)
- It should commute with all the other interactive summands
  - They does not disable each other
  - They should not influence each other's action
  - They should not influence each other's probability expression
  - Their order should not influence the next state

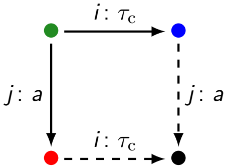
Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i})$$
$$\cdots$$
$$+ \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\boldsymbol{e_j} : \boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



Heuristics for detecting confluence on MAPA

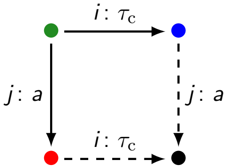
$$X(g : G) = \sum_{d_i : D_i} c_i \Rightarrow \tau \cdot X(n_i)$$
$$\ldots$$
$$+ \sum_{d_j : D_j} c_j \Rightarrow a_j \sum_{e_j : E_j} f_j : X(n_j)$$



Heuristics for verifying commutativity of summands i, j:

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i}) \\ \dots \\ + \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\boldsymbol{e_j} : \boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$

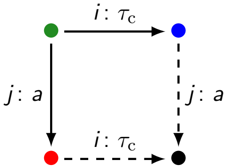


Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i}) \\ \dots \\ + \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\boldsymbol{e_j} : \boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



Heuristics for verifying commutativity of summands i, j:

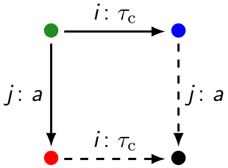
- The conditions of i and j are disjoint

  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$

  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i}:\boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i})$$
$$\dots$$
$$+ \sum_{\boldsymbol{d_j}:\boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\bullet\, \boldsymbol{e_j}:\boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint

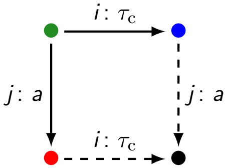
  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$

  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$

- Neither summand uses variables that are changed by the other

Heuristics for detecting confluence on MAPA

$$X(g : G) = \sum_{d_i : D_i} c_i \Rightarrow \tau \cdot X(n_i)$$
$$\dots$$
$$+ \sum_{d_j : D_j} c_j \Rightarrow a_j \sum_{e_j : E_j} f_j : X(n_j)$$



Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint

  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$

  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$

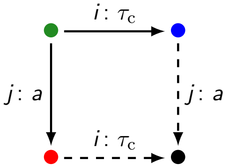
- Neither summand uses variables that are changed by the other

  i: $pc1 = 2 \wedge x > 5 \wedge y > 2 \Rightarrow \tau \cdot X(pc1 := 3, x := 0)$

  j: $pc2 = 1 \wedge y > 2 \Rightarrow send(y) \cdot X(pc2 := 2)$

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i}:\boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i})$$
$$\dots$$
$$+ \sum_{\boldsymbol{d_j}:\boldsymbol{D_j}} c_j \Rightarrow a_j \sum\!\!\!\!\bullet_{\boldsymbol{e_j}:\boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint

  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$

  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$

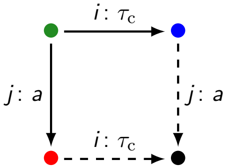
- Neither summand uses variables that are changed by the other

  i: $pc1 = 2 \wedge x > 5 \wedge y > 2 \Rightarrow \tau \cdot X(pc1 := 3, x := 0)$

  j: $pc2 = 1 \wedge y > 2 \Rightarrow send(y) \cdot X(pc2 := 2)$

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i})$$
$$\cdots$$
$$+ \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\boldsymbol{e_j} : \boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



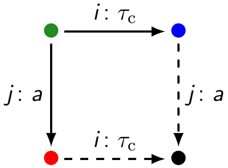
Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint
  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$
  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$
- Neither summand uses variables that are changed by the other
  i: $pc1 = 2 \wedge x > 5 \wedge y > 2 \Rightarrow \tau \cdot X(pc1 := 3, x := 0)$
  j: $pc2 = 1 \wedge y > 2 \Rightarrow send(y) \cdot X(pc2 := 2)$

(Exception allowed: change such as $x := x + 1$, usage such as $x \geq 2$)

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i})$$
$$\cdots$$
$$+ \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\boldsymbol{e_j} : \boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint

  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$
  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$

- Neither summand uses variables that are changed by the other

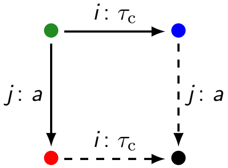
  i: $pc1 = 2 \wedge x > 5 \wedge y > 2 \Rightarrow \tau \cdot X(pc1 := 3, x := 0)$
  j: $pc2 = 1 \wedge y > 2 \Rightarrow send(y) \cdot X(pc2 := 2)$

  (Exception allowed: change such as $x := x + 1$, usage such as $x \geq 2$)

- $i = j$ and this summand only produces one transition per state

Heuristics for detecting confluence on MAPA

$$X(\boldsymbol{g} : \boldsymbol{G}) = \sum_{\boldsymbol{d_i} : \boldsymbol{D_i}} c_i \Rightarrow \tau \cdot X(\boldsymbol{n_i})$$
$$\cdots$$
$$+ \sum_{\boldsymbol{d_j} : \boldsymbol{D_j}} c_j \Rightarrow a_j \sum_{\boldsymbol{e_j} : \boldsymbol{E_j}} f_j : X(\boldsymbol{n_j})$$



Heuristics for verifying commutativity of summands i, j:

- The conditions of i and j are disjoint
  i: $pc = 3 \Rightarrow \tau \cdot X(pc := 4)$
  j: $pc = 5 \Rightarrow send(y) \cdot X(pc := 1)$
- Neither summand uses variables that are changed by the other
  i: $pc1 = 2 \wedge x > 5 \wedge y > 2 \Rightarrow \tau \cdot X(pc1 := 3, x := 0)$
  j: $pc2 = 1 \wedge y > 2 \Rightarrow send(y) \cdot X(pc2 := 2)$

  (Exception allowed: change such as $x := x + 1$, usage such as $x \geq 2$)
- $i = j$ and this summand only produces one transition per state
  i: $pc = 1 \Rightarrow \tau \cdot X(pc := 2)$

Implementation

We implemented:

- GEMMA
    - Transform GSPNs to MAPA specifications
- SCOOP
    - Generate Markov automata from MAPA specifications
    - Optimise specifications, apply confluence reduction

Implementation

We implemented:

- GEMMA
  - Transform GSPNs to MAPA specifications
- SCOOP
  - Generate Markov automata from MAPA specifications
  - Optimise specifications, apply confluence reduction

We use:

- IMCA: Quantitative analysis on Markov automata
  (expected time, time-bounded reachability, long-run average)

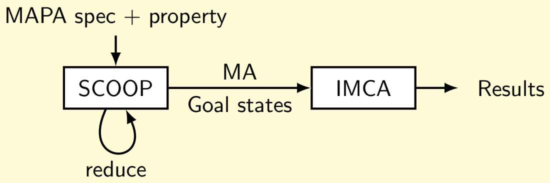
Implementation

We implemented:

- GEMMA
  - Transform GSPNs to MAPA specifications
- SCOOP
  - Generate Markov automata from MAPA specifications
  - Optimise specifications, apply confluence reduction

We use:

- IMCA: Quantitative analysis on Markov automata
  (expected time, time-bounded reachability, long-run average)



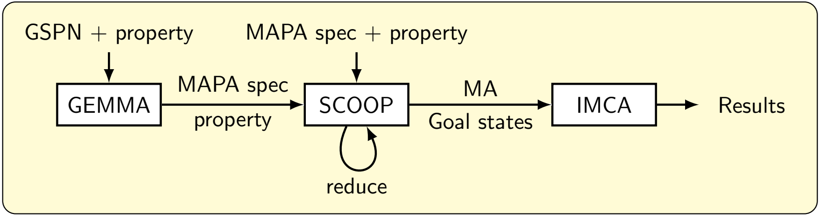
Implementation

We implemented:

- GEMMA
  - Transform GSPNs to MAPA specifications
- SCOOP
  - Generate Markov automata from MAPA specifications
  - Optimise specifications, apply confluence reduction

We use:

- IMCA: Quantitative analysis on Markov automata
  (expected time, time-bounded reachability, long-run average)



Case studies

Specification	Original state space			Reduced state space			Reduction	
	States	Trans.	IMCA	States	Trans.	IMCA	States	Time
leader-3-7	25,505	34,257	103.8	4,652	5,235	5.2	82%	90%
leader-3-9	52,465	71,034	214.3	9,058	10,149	9.9	83%	92%
leader-3-11	93,801	127,683	431.7	15,624	17,463	16.7	83%	93%
leader-4-2	8,467	11,600	74.9	2,071	2,650	5.2	76%	90%
leader-4-3	35,468	50,612	369.3	7,014	8,874	22.4	80%	92%
leader-4-4	101,261	148,024	1,325.3	17,885	22,724	62.2	82%	94%
poll-2-2-4	4,811	8,578	3.7	3,047	6,814	2.3	37%	32%
poll-2-2-6	27,651	51,098	90.9	16,557	40,004	49.1	40%	47%
poll-2-4-2	6,667	11,290	39.9	4,745	9,368	26.2	29%	32%
poll-2-5-2	27,659	47,130	1,573.8	19,721	39,192	1,053.5	29%	33%
poll-3-2-2	2,600	4,909	7.1	1,914	4,223	4.8	26%	29%
poll-4-6-1	15,439	29,506	330.0	4,802	18,869	109.3	69%	66%
poll-5-4-1	21,880	43,760	815.0	6,250	28,130	317.5	71%	61%
grid-2	2,508	4,608	2.8	1,393	2,922	1.1	44%	49%
grid-3	10,852	20,872	66.3	6,011	13,240	19.8	45%	67%
grid-4	31,832	62,356	922.5	17,565	39,558	316.5	45%	65%

Conclusions and future work

Conclusions

- We introduced the first reduction technique for MAs abstracting from internal behaviour: confluence reduction
- It preserves divergences and is closed under unions
- We showed how to detect confluence on MAPA specifications and use the representation map approach to reduce on-the-fly
- Case studies show that significant reductions can be obtained

Conclusions and future work

Conclusions

- We introduced the first reduction technique for MAs abstracting from internal behaviour: confluence reduction
- It preserves divergences and is closed under unions
- We showed how to detect confluence on MAPA specifications and use the representation map approach to reduce on-the-fly
- Case studies show that significant reductions can be obtained

Future work

- Develop even more powerful reduction techniques
- Define partial-order reduction as a restriction of confluence