Confluence versus Partial Order Reduction in Statistical Model Checking and Simulation

Mark Timmer
September 13, 2012

Aachen 2012
Problem with the presence of nondeterminism:
No single probability: minimum and maximum

Possible solutions:
Assume a (uniform) distribution
Show the nondeterminism to be spurious
Problem with the presence of nondeterminism:
No single probability: minimum and maximum
Possible solutions:
Assume a (uniform) distribution
Show the nondeterminism to be spurious
Problem with the presence of nondeterminism:

No single probability: minimum and maximum

Possible solutions:

Assume a (uniform) distribution

Show the nondeterminism to be spurious

⇐
Problem with the presence of nondeterminism:

No single probability: minimum and maximum

Possible solutions:

Assume a (uniform) distribution

Show the nondeterminism to be spurious
Problem with the presence of nondeterminism:

No single probability: minimum and maximum

Possible solutions:
- Assume a (uniform) distribution
- Show the nondeterminism to be spurious
Problem with the presence of nondeterminism:

- No single probability: minimum and maximum
Problem with the presence of nondeterminism:
- **No single probability**: minimum and maximum

Possible solutions:
- **Assume** a (uniform) distribution
- Show the nondeterminism to be **spurious**
Problem with the presence of nondeterminism:
- **No single probability**: minimum and maximum

Possible solutions:
- **Assume** a (uniform) distribution
- Show the nondeterminism to be **spurious**
Reduction techniques for nondeterminism

Reduction techniques:

- Partial order reduction (ample sets)
- Confluence reduction
Reduction techniques for nondeterminism

Reduction techniques:

- Partial order reduction (ample sets)
- Confluence reduction

Reduction function $F$:

$$F(s) \subseteq \text{enabled}(s)$$
Reduction techniques:

- Partial order reduction (ample sets)
- Confluence reduction

Reduction function $F$:

$$F(s) \subseteq \text{enabled}(s)$$
Reduction techniques for nondeterminism

Reduction techniques:
- Partial order reduction (ample sets)
- Confluence reduction

Reduction function $F$:

$F(s) \subseteq \text{enabled}(s)$
Reduction techniques:

- **Partial order reduction** (ample sets)
- **Confluence reduction**

Reduction function $F$:

$$F(s) \subseteq \text{enabled}(s)$$
Reduction techniques for nondeterminism

Reduction techniques:
- Partial order reduction (ample sets)
- Confluence reduction

Reduction function $F$:
$$F(s) \subseteq \text{enabled}(s)$$

Important:
reduced system should be equivalent
The model: Probabilistic Automata

Probabilistic Automaton:

- Non-deterministically choose a transition
- Probabilistically choose the next state
The model: Probabilistic Automata

Probabilistic Automaton:

- Non-deterministically choose a transition
- Probabilistically choose the next state
The model: Probabilistic Automata

Probabilistic Automaton:

- Non-deterministically choose a transition
- Probabilistically choose the next state
Behaviour preservation by invisible steps

**Invisible** transitions in confluence reduction:

- Deterministic
- Stuttering
Behaviour preservation by invisible steps

Invisible transitions in confluence reduction:
- Deterministic
- Stuttering

Invisible $\tau$-steps might disable behaviour...
Behaviour preservation by invisible steps

Invisible transitions in confluence reduction:

- Deterministic
- Stuttering

Invisible $\tau$-steps might disable behaviour...
**Behaviour preservation by invisible steps**

**Invisible** transitions in confluence reduction:
- Deterministic
- Stuttering

Invisible $\tau$-steps **might** disable behaviour...

...though often, they **connect equivalent states**
Behaviour preservation by invisible steps

Invisible transitions in confluence reduction:

- Deterministic
- Stuttering

Invisible $\tau$-steps might disable behaviour...

...though often, they connect equivalent states
Behaviour preservation by invisible steps

**Invisible** transitions in confluence reduction:
- Deterministic
- Stuttering

Invisible $\tau$-steps **might** disable behaviour...

...though often, they **connect** equivalent states.
Behaviour preservation by invisible steps

Invisible transitions in confluence reduction:

- Deterministic
- Stuttering

Invisible $\tau$-steps might disable behaviour...

... though often, they connect equivalent states
Confluence reduction:
denoting a subset of the invisible transitions as confluent.
Confluence reduction:

denoting a subset of the invisible transitions as confluence.

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:

\[
\begin{align*}
  & a 
  \quad \tau 
  \quad b \\
  & \quad \tau 
  \quad d \\
  & b \\
  & a 
\end{align*}
\]

Probabilistically:

\[
\begin{align*}
  & a \\
  \quad \tau \\
  & a \\
\end{align*}
\]

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

Non-probabilistically:

![Non-probabilistic diagram]

Probabilistically:

![Probabilistic diagram]
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

\[
\begin{align*}
S_0 & \xrightarrow{\frac{1}{3}} S_2 & \\
& \quad \xleftarrow{\frac{1}{3}} S_3 & \\
S_0 & \xrightarrow{\frac{1}{6}} S_4 & \\
S_1 & \xleftarrow{\frac{1}{3}} S_6 & \\
& \quad \xrightarrow{\frac{2}{3}} S_5 & \\
S_5 & \xrightarrow{a} S_1 & \\
& \quad \xleftarrow{a} S_6 & \\
& \quad \xrightarrow{a} S_5 & \\
& \quad \xleftarrow{a} S_6 & \\
\end{align*}
\]
Probabilistic example

\begin{figure}
\centering
\includegraphics[width=\textwidth]{probabilistic_example.png}
\end{figure}
Probabilistic example

\[ \begin{array}{c}
\text{s}_0 \\
\text{s}_1 \\
\text{s}_5 \\
\text{s}_6 \\
\end{array} \]

- **a**
- **b**
- Probabilities: \( \frac{2}{3} \) and \( \frac{1}{3} \)
Alterations to the concept of confluence

- Transitions may be mimicked by differently-labelled transitions
- Transitions only have to be invisible locally
- More liberal notion of equivalence of distributions
Alterations to the concept of confluence

- Transitions may be mimicked by **differently-labelled** transitions
- Transitions only have to be **invisible locally**
- More liberal notion of **equivalence of distributions**

![Diagram](image-url)
Alterations to the concept of confluence

- Transitions may be mimicked by *differently-labelled* transitions
- Transitions only have to be *invisible locally*
- More liberal notion of *equivalence of distributions*

\[
\begin{array}{c}
\text{a} \\
\text{b} \\
\text{c} \\
\text{a}
\end{array}
\]
Alterations to the concept of confluence

- Transitions may be mimicked by differently-labelled transitions
- Transitions only have to be invisible locally
- More liberal notion of equivalence of distributions
Alterations to the concept of confluence

- Transitions may be mimicked by differently-labelled transitions
- Transitions only have to be invisible locally
- More liberal notion of equivalence of distributions
Alterations to the concept of confluence

- More liberal notion of equivalence of distributions
Alterations to the concept of confluence

- More liberal notion of equivalence of distributions

**Definition (Old)**

Distributions $\mu$ and $\nu$ are $T$-equivalent, if there exists a partitioning $\text{spt}(\mu) = \biguplus_{i=1}^{n} S_i$ of the support of $\mu$ and an ordering $\text{spt}(\nu) = \{s_1, \ldots, s_n\}$ of the support of $\nu$, such that $\forall 1 \leq i \leq n$

$$\mu(S_i) = \nu(s_i) \land (S_i = \{s_i\} \lor \forall s \in S_i . \exists a \in \Sigma . s \xrightarrow{a} s_i \in T).$$
Alterations to the concept of confluence

- More liberal notion of equivalence of distributions

**Definition (Old)**

Distributions $\mu$ and $\nu$ are $\mathcal{T}$-equivalent, if there exists a partitioning $\text{spt}(\mu) = \bigcup_{i=1}^{n} S_i$ of the support of $\mu$ and an ordering $\text{spt}(\nu) = \{s_1, \ldots, s_n\}$ of the support of $\nu$, such that $\forall 1 \leq i \leq n$

$$\mu(S_i) = \nu(s_i) \land (S_i = \{s_i\} \lor \forall s \in S_i . \exists a \in \Sigma . s \xrightarrow{a} s_i \in \mathcal{T})$$

**Definition (New)**

Distributions $\mu$ and $\nu$ are $\mathcal{T}$-equivalent, if $\mu \equiv_R \nu$ for the smallest equivalence relation $R$ containing the set

$$\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\}$$
Alterations to the concept of confluence

\[ \mu(S_i) = \nu(s_i) \land (S_i = \{s_i\}) \lor \forall s \in S_i . \exists a \in \Sigma . s \xrightarrow{a} s_i \in T \]
Alterations to the concept of confluence

\[ \mu(S_i) = \nu(s_i) \land (S_i = \{s_i\} \lor \forall s \in S_i . \exists a \in \Sigma . s \xrightarrow{a} s_i \in T) \]
Alterations to the concept of confluence

\[ \mu(S_i) = \nu(s_i) \land (S_i = \{s_i\} \lor \forall s \in S_i . \exists a \in \Sigma . s \xrightarrow{a} s_i \in T) \]
Alterations to the concept of confluence

\[
\mu(S_i) = \nu(s_i) \land (S_i = \{s_i\} \land \forall s \in S_i . \exists a \in \Sigma . s \xrightarrow{a} s_i \in T)
\]

\[
\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in T\}
\]
Correctness of confluence reduction

Even though

- Transitions may be mimicked by \textit{differently-labelled} transitions
- Transitions only have to be \textit{invisible locally}
- We have a more liberal notion of \textit{equivalence of distributions}
Correctness of confluence reduction

Even though

- Transitions may be mimicked by \textit{differently-labelled} transitions
- Transitions only have to be \textit{invisible locally}
- We have a more liberal notion of \textit{equivalence of distributions}

Still we find:

\textbf{Theorem}

\textit{Confluent transitions can be given priority, preserving }PCTL^*_X\textit{.}
On-the-fly detection of confluence

Simulation using on-the-fly confluence detection:

1. Simulate until reaching a nondeterministic choice
Simulation using on-the-fly confluence detection:

1. Simulate until reaching a nondeterministic choice
2. Check for each outgoing transition if it is confluent
   - If one choice is confluent, take it and continue
   - If no choice is confluent, abort
On-the-fly detection of confluence

Simulation using on-the-fly confluence detection:

1. Simulate until reaching a nondeterministic choice
2. Check for each outgoing transition if it is confluent
   - If one choice is confluent, take it and continue
   - If no choice is confluent, abort

(Possibly, if confluence fails, attempt the same using POR)
On-the-fly detection of confluence

Simulation using on-the-fly confluence detection:

1. Simulate until reaching a nondeterministic choice
2. Check for each outgoing transition if it is confluent
   - If one choice is confluent, take it and continue
   - If no choice is confluent, abort

(Possibly, if confluence fails, attempt the same using POR)

To check if a transition is confluent:

- Check if it is invisible
On-the-fly detection of confluence

Simulation using on-the-fly confluence detection:

1. Simulate until reaching a nondeterministic choice
2. Check for each outgoing transition if it is confluent
   - If one choice is confluent, take it and continue
   - If no choice is confluent, abort

(Possibly, if confluence fails, attempt the same using POR)

To check if a transition is confluent:

- Check if it is invisible
- Check if all its neighbouring transitions are mimicked
  - For this, additional transitions might need to be confluent
Checking a transition for confluence

- No; it is not invisible
- It is invisible
- Yes, possibly by the transition d.
- But then f has to be confluent: check this.
Checking a transition for confluence

Check if c is confluent
Checking a transition for confluence

Check if \( c \) is confluent

- No; it is not invisible

Check if \( a \) is confluent

It is invisible

Is the \( c \)-transition mimicked? Possibly by the \( d \)-transition

But then \( f \) has to be confluent: check this.
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible
- Check if $a$ is confluent

![Diagram](attachment:image.png)
Checking a transition for confluence

- Check if \( c \) is confluent
  - No; it is not invisible
- Check if \( a \) is confluent
  - It is invisible
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible
- Check if $a$ is confluent
  - It is invisible
  - Is the $c$-transition mimicked?
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible
- Check if $a$ is confluent
  - It is invisible
  - Is the $c$-transition mimicked?
    - Possibly by the $d$-transition
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible
- Check if $a$ is confluent
  - It is invisible
  - Is the $c$-transition mimicked?
    - Possibly by the $d$-transition
    - But then $f$ has to be confluent: check this
    - ...
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible
- Check if $a$ is confluent
  - It is invisible
  - Is the $c$-transition mimicked?
    - Possibly by the $d$-transition
    - But then $f$ has to be confluent: check this
    - ...
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible

- Check if $a$ is confluent
  - It is invisible
  - Is the $c$-transition mimicked?
    - Possibly by the $d$-transition
    - But then $f$ has to be confluent: check this
    - ...
Checking a transition for confluence

- Check if $c$ is confluent
  - No; it is not invisible
- Check if $a$ is confluent
  - It is invisible
  - Is the $c$-transition mimicked?
    - Possibly by the $d$-transition
    - But then $f$ has to be confluent: check this
    - ...
Checking a transition for confluence

Check if $d$ is confluent
Checking a transition for confluence

- Check if $d$ is confluent
  - No; it is not invisible
Checking a transition for confluence

- Check if $d$ is confluent
  - No; it is not invisible
- Check if $b$ is confluent
Checking a transition for confluence

- Check if $d$ is confluent
  - No; it is not invisible
- Check if $b$ is confluent
  - It is invisible
Checking a transition for confluence

- Check if \( d \) is confluent
  - No; it is not invisible
- Check if \( b \) is confluent
  - It is invisible
  - Is the \( d \)-transition mimicked?
Checking a transition for confluence

- Check if $d$ is confluent
  - No; it is not invisible
- Check if $b$ is confluent
  - It is invisible
  - Is the $d$-transition mimicked?
    - Possibly by the $e$-transition
Checking a transition for confluence

- Check if \( d \) is confluent
  - No; it is not invisible
- Check if \( b \) is confluent
  - It is invisible
  - Is the \( d \)-transition mimicked?
    - Possibly by the \( e \)-transition
    - But then \( g \) has to be confluent: check this
    - ...
Checking a transition for confluence

- Check if $d$ is confluent
  - No; it is not invisible
- Check if $b$ is confluent
  - It is invisible
  - Is the $d$-transition mimicked?
    - Possibly by the $e$-transition
    - But then $g$ has to be confluent: check this
    - ...
Checking a transition for confluence

- Check if $d$ is confluent
  - No; it is not invisible
- Check if $b$ is confluent
  - It is invisible
  - Is the $d$-transition mimicked?
    - Possibly by the $e$-transition
    - But then $g$ has to be confluent: check this
    - ...
Detecting equivalence of transitions

\[
\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\}
\]
Detecting equivalence of transitions

\[ \{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\} \]
Detecting equivalence of transitions

\[
\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\}
\]
Detecting equivalence of transitions

\( \{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\} \)
Detecting equivalence of transitions

\[(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in T\]
Detecting equivalence of transitions

\[ \{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\} \]
Detecting equivalence of transitions

\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\}
Detecting equivalence of transitions

\begin{equation}
\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T}\}
\end{equation}
Detecting equivalence of transitions

\{(s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in T\}
Detecting equivalence of transitions

\{ (s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \xrightarrow{a} t \in \mathcal{T} \}
Detecting equivalence of transitions

\{ (s, t) \mid s \in \text{spt}(\mu), t \in \text{spt}(\nu), s \overset{a}{\rightarrow} t \in \mathcal{T} \}
**Implementation**

*modes*: a discrete-event simulator for the ModEST language
- Statistical model checking of deterministic systems
  - Partial order reduction
  - Confluence reduction
Implementation

modes: a discrete-event simulator for the MODEST language
- Statistical model checking of deterministic systems
  - Partial order reduction
  - Confluence reduction

Three case studies:
- Dining Cryptographers
- IEEE 802.3 CSMA/CD
- Binary Exponential Backoff
Case study: Dining Cryptographers

**Table:** Confluence simulation runtime compared

<table>
<thead>
<tr>
<th>model \ (N)</th>
<th>simulation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>uniform</td>
<td>confluence</td>
</tr>
<tr>
<td>3</td>
<td>3 s</td>
<td>13 s</td>
</tr>
<tr>
<td>4</td>
<td>4 s</td>
<td>66 s</td>
</tr>
<tr>
<td>5</td>
<td>5 s</td>
<td>338 s</td>
</tr>
</tbody>
</table>

Partial order reduction was **not able** to resolve the nondeterminism.
Case study: CSMA/CD

Table: Confluence simulation runtime compared

<table>
<thead>
<tr>
<th>model \ (RED, BC_{MAX})</th>
<th>simulation</th>
<th>model checking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>uniform</td>
<td>confluence</td>
</tr>
<tr>
<td>(2, 1)</td>
<td>6 s</td>
<td>18 s</td>
</tr>
<tr>
<td>(1, 1)</td>
<td>6 s</td>
<td>18 s</td>
</tr>
<tr>
<td>(1, 2)</td>
<td>11 s</td>
<td>48 s</td>
</tr>
</tbody>
</table>

Partial order reduction was not able to resolve the nondeterminism. (for confluence, probabilistic transitions needed to be synchronised)
Case study: Binary Exponential Backoff

<table>
<thead>
<tr>
<th>model ((K, N, H))</th>
<th>uniform simulation</th>
<th>partial order simulation</th>
<th>confluence simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>((4, 3, 3))</td>
<td>1 s</td>
<td>2 s</td>
<td>2 s</td>
</tr>
<tr>
<td>((8, 7, 4))</td>
<td>14 s</td>
<td>18 s</td>
<td>16 s</td>
</tr>
</tbody>
</table>
Conclusions

- We improved on the notion of confluence reduction:
  - Independent of action labels
  - Independent of global behaviour
  - More liberal equivalence of distributions
Conclusions

- We improved on the notion of confluence reduction:
  - Independent of action labels
  - Independent of global behaviour
  - More liberal equivalence of distributions
- We provided an on-the-fly detection algorithm for SMC
- We implemented the new technique in MODEST
Conclusions

- We improved on the notion of confluence reduction:
  - Independent of action labels
  - Independent of global behaviour
  - More liberal equivalence of distributions

- We provided an on-the-fly detection algorithm for SMC

- We implemented the new technique in Modest

- Case studies show that confluence reduction reduces more and slightly faster than partial order reduction

- More models can now statistically be checked
Questions