Why Confluence is More Powerful than Ample Sets in Probabilistic and Non-Probabilistic Branching Time

Mark Timmer
May 23, 2012
Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., an MDP)
Probabilistic model checking:

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

Non-deterministically choose a transition

Probabilistically choose the next state
The context – probabilistic model checking

Probabilistic model checking:

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

Non-deterministically choose a transition

Probabilistically choose the next state
**Probabilistic model checking:**

- Verifying **quantitative properties**,
- Using a **probabilistic model** (e.g., an MDP)

Non-deterministically choose a transition
Probabilistically choose the next state

**Main limitation (as for non-probabilistic model checking):**

- Susceptible to the **state space explosion** problem
Combating the state space explosion

- Probabilistic specification

  Instantiation

  State space (MDP)

  Minimisation (optimisation)

  State space (MDP)
Combating the state space explosion

Probabilistic specification

Partially ordered reduction

Instantiation

Confluence reduction

Minimisation (optimisation)

Optimised instantiation

State space (MDP)
Combating the state space explosion

Probabilistic specification

Instantiation

State space (MDP)

Optimised instantiation
- Partial-order reduction
- Confluence reduction
  (initially for PAs)

Minimisation (optimisation)
Reductions – an overview

Reduction function:
\[ R : S \rightarrow 2^{\Sigma}(R(s) \subseteq \text{enabled}(s)) \]

If \( R(s) \neq \text{enabled}(s) \), then \( R(s) \) consists of remaining transitions.
Reductions – an overview

Reduction function: $R : S \rightarrow 2^\Sigma (R(s) \subseteq \text{enabled}(s))$

If $R(s) \neq \text{enabled}(s)$, then $R(s)$ consists of remaining transitions.
Reductions – an overview

Reduction function: $R : S \rightarrow 2^{\Sigma \left( R(s) \subseteq \text{enabled}(s) \right)}$

If $R(s) \neq \text{enabled}(s)$, then $R(s)$ consists of remaining transitions.
Reduction function:

\[ R: S \rightarrow 2^\Sigma \]
Reductions – an overview

Reduction function:

\[ R: S \to 2^\Sigma \quad (R(s) \subseteq \text{enabled}(s)) \]
Reduction function:

\[ R : S \rightarrow 2^\Sigma \quad (R(s) \subseteq \text{enabled}(s)) \]

If \( R(s) \neq \text{enabled}(s) \), then \( R(s) \) consists of remaining transitions.
Basic concepts

Stuttering transition:
No observable change

Stuttering action:
Yields only stuttering transitions

\[
\{p\} \xrightarrow{a} \{q\} \xrightarrow{b} \{p\} = \text{st} \{p\} \xrightarrow{b} \{q\} \xrightarrow{a} \{p\}
\]
Basic concepts

Stuttering transition:
- No observable change

Stuttering action:
Yields only stuttering transitions

\[ s_1 \xrightarrow{a} s_2, s_1 \xrightarrow{b} s_3, s_2 \xrightarrow{b} s_4, s_3 \xrightarrow{a} s_4 \]
Basic concepts

Stuttering transition:
- No observable change

Stuttering action:
- Yields only stuttering transitions
Basic concepts

Stuttering transition:
- No observable change

Stuttering action:
- Yields only stuttering transitions
Basic concepts

Stuttering transition:
- No observable change

Stuttering action:
- Yields only stuttering transitions

\[
\{p\} \{p\} \{q\} =_{\text{st}} \{p\} \{q\} \{q\}
\]
Basic concepts

Stuttering transition:
- No observable change

Stuttering action:
- Yields only stuttering transitions

\[
\{p\} \{p\} \{q\} =_{\text{st}} \{p\} \{q\} \{q\}
\]
Basic concepts

**Stuttering transition:**
- No observable change

**Stuttering action:**
- Yields *only* stuttering transitions

\[ \{p\}{p}\{q\} \overset{st}{=} \{p\}{q}\{q\} \]
Correctness criteria

Correctness criteria for reductions:

- Preservation of \( \text{LTL}_X \) (linear time)
- Preservation of \( \text{CTL}^*_X \) (branching time)
Correctness criteria

Correctness criteria for reductions:

- Preservation of (quantitative) $\operatorname{LTL}^\chi$ (linear time)
- Preservation of (P)$\operatorname{CTL}^\chi$ (branching time)
Correctness criteria for reductions:

- Preservation of (quantitative) $\text{LTL}_\times$ (linear time)
- Preservation of (P)$\text{CTL}_\times^*$ (branching time)

<table>
<thead>
<tr>
<th></th>
<th>Partial-order reduction</th>
<th>Confluence reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear time</td>
<td>[BGC’04, AN’04]</td>
<td>–</td>
</tr>
<tr>
<td>Branching time</td>
<td>[BAG’06]</td>
<td>[TSP’11]</td>
</tr>
</tbody>
</table>
Correctness criteria for reductions:

- Preservation of (quantitative) LTL\(\setminus \chi\) (linear time)
- Preservation of (P)CTL\(^*\chi\) (branching time)

<table>
<thead>
<tr>
<th>Linear time</th>
<th>Partial-order reduction</th>
<th>Confluence reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear time</td>
<td>[BGC’04, AN’04]</td>
<td>–</td>
</tr>
<tr>
<td>Branching time</td>
<td>[BAG’06]</td>
<td>(\iff)</td>
</tr>
</tbody>
</table>
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Independence of $a$ and $b$:
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Independence of $a$ and $b$: 

```
\[
\begin{align*}
\text{Por} &= \text{sat} \rightarrow \text{sat} \\
\text{sa} &= \text{sat} \rightarrow \text{sat} \\
\text{as} &= \text{sat} \rightarrow \text{sat} \\
\text{ba} &= \text{sat} \rightarrow \text{sat} \\
\text{ab} &= \text{sat} \rightarrow \text{sat} \\
\text{ba} &= \text{sat} \rightarrow \text{sat} \\
\text{ab} &= \text{sat} \rightarrow \text{sat} \\
\text{ba} &= \text{sat} \rightarrow \text{sat}
\end{align*}
\]```

Why Confluence is More Powerful than Ample Sets
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

**Independence of** \(a\) and \(b\):

![Diagram showing independence of actions](attachment:image.png)
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Independence of $a$ and $b$:
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Independence of \( a \) and \( b \):
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Independence of $a$ and $b$:
Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Independence of $a$ and $b$:

\[ P[s_1 \xrightarrow{ab} s] = P[s_1 \xrightarrow{ba} s], \forall s \]
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
  - Based on independent actions and ample sets

**Ample set conditions:**

Given a reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

**Ample set** conditions:

Given a reduction function \( R: S \rightarrow 2^{\Sigma} \), for every \( s \in S \)

- **A0** \( \emptyset \neq R(s) \)
- **A1**
- **A2**
- **A3**
- **A4**
Partial-order reduction [Baier, D’Argenio, Größer, 2006]
  - Based on independent actions and ample sets

Ample set conditions:

- **A0** $\emptyset \neq R(s)$
- **A1** If $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions
- **A2** For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$
- **A3** Every cycle in the reduced MDP contains a fully-expanded state (if $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s$, then $\exists s_i. R(s_i) = \text{enabled}(s_i)$)
- **A4** If $R(s) \neq \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic and stuttering
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

**Ample set conditions:**

- For every reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$:
  - $A_0$: $\emptyset \neq R(s)$
  - $A_1$: if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions
  - $A_2$: For every original path $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$
  - $A_3$: Every cycle in the reduced MDP contains a fully-expanded state (if $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s_0$, then $\exists s_i. R(s_i) = \text{enabled}(s_i)$)
  - $A_4$: if $R(s) \neq \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic and stuttering
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

- $A_0$: $\emptyset \neq R(s)$
- $A_1$: If $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions
- $A_2$: For every original path $s \rightarrow a_1 \rightarrow s_1 \rightarrow a_2 \rightarrow \ldots \rightarrow a_n \rightarrow s_n \rightarrow b \rightarrow t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$
- $A_3$: Every cycle in the reduced MDP contains a fully-expanded state (if $s \rightarrow a_1 \rightarrow s_1 \rightarrow a_2 \rightarrow \ldots \rightarrow a_n \rightarrow s_n = s$, then $\exists s_i$. $R(s_i) = \text{enabled}(s_i)$)
- $A_4$: If $R(s) \neq \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic and stuttering
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Ample set conditions:

Given a reduction function \( R : S \rightarrow 2^\Sigma \), for every \( s \in S \)

- A0 \( \emptyset \neq R(s) \)
- A1 if \( R(s) \neq \text{enabled}(s) \), then \( R(s) \) contains only stuttering actions
- A2
- A3
- A4
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Ample set conditions:

Original

\[\{p\}\]
\[\{q\}\]

Reduced

\[\{p\}\]
\[\{q\}\]
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

Original

\[
\begin{align*}
\text{Original} & \quad \{ p \} \\
\{ r \} & \quad \{ q \} \\
\{ q \} & \quad \{ q \}
\end{align*}
\]

Reduced

\[
\begin{align*}
\text{Reduced} & \quad \{ p \} \\
\{ r \} & \quad \{ q \} \\
\{ q \} & \quad \{ q \}
\end{align*}
\]
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

Given a reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$

A0 $\emptyset \neq R(s)$

A1 if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions

A2 For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$

A3

A4
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Ample set conditions:

![Diagram showing original and reduced states and actions]
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Ample set conditions:

Original
- \(a\) to \(b\)
- \(b\) to \(a\)
- \(b\) to \(c\)

Reduced
- \(a\) to \(b\)
- \(b\) to \(\cdot\)

Why Confluence is More Powerful than Ample Sets
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

Given a reduction function $R: S \to 2^\Sigma$, for every $s \in S$

A0 $\emptyset \neq R(s)$

A1 if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions

A2 For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$

A3 Every cycle in the reduced MDP contains a fully-expanded state (if $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s$, then $\exists s_i . R(s_i) = \text{enabled}(s_i)$)

A4
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Ample set conditions:

Original

\[
\begin{align*}
\text{Original} & : a & \rightarrow & b & \rightarrow & a \\
& \rightarrow & a & \rightarrow & b & \rightarrow \\
& \rightarrow & b & \rightarrow & a & \rightarrow \\
& \rightarrow & & & & \\
\end{align*}
\]

Reduced

\[
\begin{align*}
\text{Reduced} & : a & \rightarrow & b & \rightarrow & a \\
& \rightarrow & & & & \\
& \rightarrow & & & & \\
& \rightarrow & & & & \\
\end{align*}
\]
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

- Original
- Reduced
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

![Diagram showing original and reduced MDPs]
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

Given a reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$

- **A0** $\emptyset \neq R(s)$
- **A1** if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions
- **A2** For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$
- **A3** Every cycle in the reduced MDP contains a fully-expanded state (if $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s$, then $\exists s_i . R(s_i) = \text{enabled}(s_i)$)
- **A4** if $R(s) \neq \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Ample set conditions:

Given a reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$

A0 $\emptyset \neq R(s)$

A1 if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions

A2 For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \not\in R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$

A3 Every cycle in the reduced MDP contains a fully-expanded state (if $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s$, then $\exists s_i . R(s_i) = \text{enabled}(s_i)$)

A4 if $R(s) \neq \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic
Partial-order reduction: ample sets

Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

Given a reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$

A0. $\emptyset \neq R(s)$

A1. if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions

A2. For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$

A3. Every cycle in the reduced MDP contains a fully-expanded state (if $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s$, then $\exists s_i . R(s_i) = \text{enabled}(s_i)$)

A4. if $R(s) \neq \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic and stuttering
Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]

- Based on equivalent distributions and confluent transitions
Confluence

Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]
- Based on equivalent distributions and confluent transitions

\( T \)-equivalent distributions

![Graph showing T-equivalent distributions](image-url)
Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]
- Based on equivalent distributions and confluent transitions

$T$-equivalent distributions
Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]

- Based on equivalent distributions and confluent transitions

$T$-equivalent distributions

\[
\begin{align*}
\text{s}_1 & \quad 0.2 & \quad 0.8 & \quad 0.8 & \quad 0.2 \\
\text{s}_2 & \quad & \quad & \quad & \quad \\
\text{t}_1 & \quad & \quad & \quad & \\
\text{t}_2 & \quad & \quad & \quad & \\
\text{t}_4 & \quad & \quad & \quad &
\end{align*}
\]
Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]
- Based on equivalent distributions and confluent transitions

The main idea:
- Choose a set $T$ of transitions
- Make sure all of them are confluent
- $R(s) = \text{enabled}(s)$ or $R(s) = \{a\}$ such that $(s \xrightarrow{a} t) \in T$
Confluence

Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]

- Based on equivalent distributions and confluent transitions

The main idea:

- Choose a set $T$ of transitions
- Make sure all of them are confluent
- $R(s) = \text{enabled}(s)$ or $R(s) = \{a\}$ such that $(s \xrightarrow{a} t) \in T$
- Make sure $T$ is acyclic to prevent infinite postponing
A set of transitions $T$ is confluent if

- Every transition in $T$ is labelled by a deterministic stuttering action.
- If $s \xrightarrow{} s' \in T$ and $s \xrightarrow{b} \mu$, then
  1. either $s' \xrightarrow{b} \nu$ and $\mu$ is $T$-equivalent to $\nu$
  2. or $\mu(s') = 1$ ($b$ deterministically goes to $s'$)
A set of transitions $T$ is confluent if

- Every transition in $T$ is labelled by a deterministic stuttering action
- If $s \xrightarrow{\tau} s' \in T$ and $s \xrightarrow{b} \mu$, then
  1. either $s' \xrightarrow{b} \nu$ and $\mu$ is $T$-equivalent to $\nu$
  2. or $\mu(s') = 1$ ($b$ deterministically goes to $s'$)
A set of transitions $T$ is confluent if

- Every transition in $T$ is labelled by a deterministic stuttering action.
- If $s \xrightarrow{\tau} s' \in T$ and $s \xrightarrow{b} \mu$, then
  1. either $s' \xrightarrow{b} \nu$ and $\mu$ is $T$-equivalent to $\nu$
  2. or $\mu(s') = 1$ ($b$ deterministically goes to $s'$)

Why Confluence is More Powerful than Ample Sets
A set of transitions $T$ is confluent if

- Every transition in $T$ is labelled by a **deterministic stuttering** action
- If $s \xrightarrow{\tau} s' \in T$ and $s \xrightarrow{b} \mu$, then
  
  1. either $s' \xrightarrow{b} \nu$ and $\mu$ is $T$-equivalent to $\nu$
  2. or $\mu(s') = 1$ ($b$ deterministically goes to $s'$)
Comparison

**Similarities** among ample sets and confluence:

- **Requirement**
  
  
  - Size of $R(s)$
  
  
  
  - $R(s) = \text{enabled}(s)$ or $|R(s)| = 1$

- **Remaining transitions**
  
  - Deterministic and stuttering

- **Acyclicity**
  
  - No cycle of remaining transitions allowed

- **Preservation**
  
  - Branching time properties

**Differences between ample sets and confluence:**

- **POR**
  
  - For every original path $s - a_1 - \rightarrow s_1 - a_2 - \rightarrow ... - a_n - \rightarrow s_n - b - \rightarrow t$ such that $b \not\in R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$

- **Conf**
  
  - If $s - \tau - \rightarrow t$ and $s - b - \rightarrow \mu$, then $\mu = \text{dirac}(t)$ or $t - b - \rightarrow \nu$ and $\mu$ is equivalent to $\nu$. 
Similarities among ample sets and confluence:

<table>
<thead>
<tr>
<th>Size of $R(s)$</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R(s) = \text{enabled}(s)$ or $</td>
<td>R(s)</td>
</tr>
</tbody>
</table>
## Comparison

**Similarities** among ample sets and confluence:

<table>
<thead>
<tr>
<th></th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of $R(s)$</td>
<td>$R(s) = \text{enabled}(s)$ or $</td>
</tr>
<tr>
<td>Remaining transitions</td>
<td>Deterministic and stuttering</td>
</tr>
</tbody>
</table>
### Similarities among ample sets and confluence:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Size of $R(s)$</th>
<th>Remaining transitions</th>
<th>Acyclicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R(s) = \text{enabled}(s)$ or $</td>
<td>R(s)</td>
<td>= 1$</td>
<td>$R(s)$</td>
</tr>
</tbody>
</table>
**Similarities** among ample sets and confluence:

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of $R(s)$</td>
</tr>
<tr>
<td>Remaining transitions</td>
</tr>
<tr>
<td>Acyclicity</td>
</tr>
<tr>
<td>Preservation</td>
</tr>
</tbody>
</table>
## Comparison

### Similarities among ample sets and confluence:

| Requirement | Size of $R(s)$ | $R(s) = \text{enabled}(s)$ or $|R(s)| = 1$ |
|--------------|---------------|------------------------------------------|
| Deterministic and stuttering | Remaining transitions | No cycle of remaining transitions allowed |
| Branching time properties | Acyclicity | Branching time properties |
| Preservation | |

### Differences between ample sets and confluence:

**POR** For every original path $s \overset{a_1}{\rightarrow} s_1 \overset{a_2}{\rightarrow} \ldots \overset{a_n}{\rightarrow} s_n \overset{b}{\rightarrow} t$ such that $b \notin R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$
Comparison

Similarities among ample sets and confluence:

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of $R(s)$</td>
</tr>
<tr>
<td>Remaining transitions</td>
</tr>
<tr>
<td>Acyclicity</td>
</tr>
<tr>
<td>Preservation</td>
</tr>
</tbody>
</table>

Differences between ample sets and confluence:

**POR** For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \not\in R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i \in R(s)$

**Conf** If $s \xrightarrow{\tau} t$ and $s \xrightarrow{b} \mu$, then $\mu = \text{dirac}(t)$ or $t \xrightarrow{b} \nu$ and $\mu$ is equivalent to $\nu$. 
Theorem

Let $R$ be a reduction function satisfying the ample set conditions. Then, all remaining transitions are confluent.
Comparison – POR implies Confluence

Theorem

Let \( R \) be a reduction function satisfying the ample set conditions. Then, all remaining transitions are confluent.

Or:

Any reduction allowed by partial-order reduction is also allowed by confluence reduction.
Comparison – POR implies Confluence

Theorem

Let $R$ be a reduction function satisfying the ample set conditions. Then, all remaining transitions are confluent.

Or:

Any reduction allowed by partial-order reduction is also allowed by confluence reduction.

Proof (sketch).

1. Take the set of all remaining transitions of the partial-order reduction.
2. Recursively add transitions needed to complete the confluence diamonds.
3. Prove that the resulting set is indeed confluent.
Comparison – Confluence does not imply POR

POR's notion of independence is stronger than necessary.
Comparison – Confluence does not imply POR

POR's notion of independence is stronger than necessary.
Comparison – Confluence does not imply POR

POR's notion of independence is stronger than necessary.
Comparison – Confluence does not imply POR

POR’s notion of independence is stronger than necessary.
Comparison – Confluence does not imply POR

POR’s notion of independence is stronger than necessary.
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
- Do not allow overlapping distributions to be equivalent
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
- Do not allow overlapping distributions to be equivalent
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
- Do not allow overlapping distributions to be equivalent
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
- Do not allow overlapping distributions to be equivalent
- Require action-separability
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
- Do not allow overlapping distributions to be equivalent
- Require action-separability
Strengthening of confluence

We can change confluence in the following way:

- Do not allow shortcuts
- Do not allow overlapping distributions to be equivalent
- Require action-separability
Relaxing of partial-order reduction

We can change partial-order reduction in the following way:

- Relax the **dependency condition**
Relaxing of partial-order reduction

We can change partial-order reduction in the following way:

- Relax the **dependency condition**

  For every original path \( s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t \) such that \( b \neq R(s) \) and \( R(s) \) depends on \( b \) at \( s \), there exists an \( i \) such that \( a_i \in R(s) \)
We can change partial-order reduction in the following way:

- Relax the dependency condition

For every original path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ such that $b \neq R(s)$ and $R(s)$ depends on $b$ at $s$, there exists an $i$ such that $a_i \in R(s)$.
Strengthening of confluence

Theorem

Every acyclic action-separable strengthened confluence reduction is a relaxed ample set reduction and vice versa.
Theorem

Every acyclic action-separable strengthened confluence reduction is a relaxed ample set reduction and vice versa.

Corollary

In the non-probabilistic setting, the same statements hold: confluence is stronger than partial-order reduction, and the notions are equivalent for the adjusted definitions.
State space generation using representatives:
Implications

State space generation using representatives:

Original

1. Original
2. Reduction
3. Representative in bottom strongly connected component
4. Additional reduction of states and transitions
5. No need for an explicit cycle condition anymore!
Implications

State space generation using representatives:

Original

Reduction

Why Confluence is More Powerful than Ample Sets
Implications

State space generation using representatives:

- Original
- Reduction
- Representative in bottom strongly connected component
- Additional reduction of states and transitions
- No need for an explicit cycle condition anymore!
State space generation using representatives:

- Representative in **bottom strongly connected component**
- **Additional reduction** of states and transitions
- No need for an explicit cycle condition anymore!
What to take home from this . . .

- We adapted the existing notion of confluence reduction to work in a state-based setting with MDPs.
- We proved that every ample set can be mimicked by a confluent set, but the converse doesn’t always hold.
- We showed how to make ample set reduction and confluence reduction equivalent.
- We demonstrated one implication of our results, applying a technique from confluence reduction to POR.
- The results are independent of specific heuristics, and also hold non-probabilistically.
Questions

A paper, containing all details and proofs, can be found at
http://wwwhome.cs.utwente.nl/~timmer/research.php