Probabilistic model checking:

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

**Probabilistic model checking:**

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

\[ s_1 \xrightarrow{0.1} s_2 \]
\[ s_1 \xleftarrow{0.9} s_2 \]

- 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

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)
- Optimised instantiation
  - Partial-order reduction
  - Confluence reduction (initially for PAs)
Combating the state space explosion

Probabilistic specification

Instantiation

Optimised instantiation

State space (MDP)

Minimisation (optimisation)
Combating the state space explosion

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

State space (MDP)

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 reduction 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 reduction transitions.
Reductions – an overview

Reduction function:

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

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

Confluence Reduction versus Partial-Order Reduction
Reductions – an overview

Reduction function:

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

Reduction function:

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

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 reduction transitions.
Basic concepts

Stuttering transition:
No observable change

Stuttering action:
Yields only stuttering transitions

\[
\text{Stuttering transition: } \text{No observable change}
\]

\[
\text{Stuttering action: } \text{Yields only stuttering transitions}
\]

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

Stuttering transition:
- No observable change
Basic concepts

Stuttering transition:
- No observable change

Stuttering action:
- Yields only stuttering transitions

Diagram:
- States: $s_1$, $s_2$, $s_3$, $s_4$
- Edges: $a$, $b$
- Transition rules:
  - $s_1 \to s_2$: $p$ to $p$
  - $s_2 \to s_1$: $a$
  - $s_2 \to s_3$: $b$
  - $s_3 \to s_2$: $b$
  - $s_3 \to s_4$: $a$
  - $s_4 \to s_3$: $q$
  - $s_4 \to s_2$: $q$
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

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

Stuttering transition:
- No observable change

Stuttering action:
- Yields only stuttering transitions

\[
\{p\}{p}\{q\} =_{st} \{p\}{q}\{q\}
\]
Correctness criteria for reductions:

- Preservation of $\text{LTL}_{\times}$ (linear time)
- Preservation of $\text{CTL}^*_{\times}$ (branching time)
Correctness criteria for reductions:

- Preservation of (quantitative) \( \text{LTL}_X \) (linear time)
- Preservation of \( (P)\text{CTL}^*_X \) (branching time)
Correctness criteria for reductions:

- Preservation of (quantitative) $\text{LTL}_X$ (linear time)
- Preservation of (P)$\text{CTL}^*_X$ (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) $\text{LTL} \setminus \chi$ (linear time)
- Preservation of (P)$\text{CTL}^* \setminus \chi$ (branching time)

<table>
<thead>
<tr>
<th>Linear time</th>
<th>Partial-order reduction</th>
<th>Confluence reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BGC’04, AN’04]</td>
<td>$\Leftarrow$</td>
<td>[TSP’11]</td>
</tr>
<tr>
<td>Branching time</td>
<td>[BAG’06]</td>
<td></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** $b$:

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

- Based on independent actions and ample sets

Independence of $a$ and $b$: 

```
\[ P[\{s_1 - ab - \rightarrow s\}] = P[\{s_1 - ba - \rightarrow s\}], \forall s \]
```
Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

**Independence of a and b:**

```
\begin{align*}
S_1 & \xrightarrow{a} S_2 \\
S_2 & \xrightarrow{b} S_3 \\
S_3 & \xrightarrow{a} S_4 \\
S_4 & \xrightarrow{b} S_1 \\
\end{align*}
```
Partial-order reduction [Baier, D’Argenio, Größer, 2006]
- Based on independent actions and ample sets

Independence of $a$ and $b$:

Left diagram:
- $s_1$ to $s_2$ with edge $a$
- $s_1$ to $s_3$ with edge $b$
- $s_2$ to $s_4$ with edge $b$
- $s_3$ to $s_4$ with edge $a$

Right diagram:
- $s_1$ to $s_2$ with edge $a$
- $s_1$ to $s_3$ with edge $b$
- $s_2$ to $s_3$ with edge $0.25$
- $s_3$ to $s_2$ with edge $0.4$
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Independence of $a$ and $b$:

![Diagram](attachment:diagram.png)
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 $b$:

$$
\mathbb{P}[s_1 \xrightarrow{ab} s] = \mathbb{P}[s_1 \xrightarrow{ba} s], \ \forall s
$$
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 [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

**Ample set conditions:**

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

- $A0 \; \emptyset \neq R(s)$
- $A1$
- $A2$
- $A3$
- $A4$
Partial-order reduction: ample sets

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

Ample set conditions:

![Diagram showing ample sets](image-url)
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 \( s \in S \):
  - \( A_0 \): \( \emptyset \neq R(s) \)
  - \( A_1 \): 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: ample sets

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

Ample set conditions:

Original

Reduced

\[ \text{Ample set conditions:} \]

- For every original path \( s - a_1 - \rightarrow s_1 - a_2 - \rightarrow \ldots - a_n - \rightarrow s_n - b \rightarrow t \) such that \( b \notin R(s) \) and \( b \) depends on \( R(s) \), there exists an \( i \) such that \( a_i \in R(s) \).
- Every cycle in the reduced MDP contains a fully-expanded state (if \( s - a_1 - \rightarrow s_1 - a_2 - \rightarrow \ldots - a_n - \rightarrow s_n = s \), then \( \exists s_i. R(s_i) = \text{enabled}(s_i) \)).
- 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:**

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 [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

**Ample set conditions:**

![Ample set conditions diagram](image-url)
Partial-order reduction: ample sets

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

- Based on independent actions and ample sets

Ample set conditions:

Original

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

Reduced

\[
\begin{align*}
\text{Reduced} & \quad \{p\} \\
\{p\} & \quad a \\
\{q\} & \quad \{p\} \\
\{q\} & \quad \{q\}
\end{align*}
\]
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 $s \in S$:
  - $A_0$: $\emptyset \neq R(s)$
  - $A_1$: if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions
  - $A_2$: For every original path $s \xrightarrow{a} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{b} t$ 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 $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n = s$, then $\exists s_i. R(s_i) = \text{enabled}(s_i)$)
  - $A_4$: 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:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>$\emptyset \neq R(s)$</td>
</tr>
<tr>
<td>A1</td>
<td>If $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions</td>
</tr>
<tr>
<td>A2</td>
<td>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 $a_i \in R(s)$</td>
</tr>
<tr>
<td>A3</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td></td>
</tr>
</tbody>
</table>

Given a reduction function $R: S \rightarrow 2^\Sigma$, for every $s \in S$
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

Original

\[
\begin{align*}
& a \
& b \
& c \
& a \
& b \
& c \
& a
\end{align*}
\]

Reduced

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

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

Ample set conditions:

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

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

Ample set conditions:

- Original
  - a
  - b
  - a
  - b

- Reduced
  - a
  - b

UNIVERSITY OF TWENTE.
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
Partial-order reduction: ample sets

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

Ample set conditions:

Original

\[ a \rightarrow b \rightarrow a \rightarrow b \rightarrow a \]

Reduced

\[ a \rightarrow b \rightarrow a \]

\[ b \rightarrow a \]

\[ a \rightarrow b \]

\[ a \rightarrow b \]

\[ b \rightarrow a \]

\[ a \rightarrow b \]

\[ b \rightarrow a \]
Partial-order reduction [Baier, D’Argenio, Größer, 2006]

- Based on independent actions and ample sets

Ample set conditions:

---

**Original**

```
Original
    a
   / \
  a   b
 /   \
/     \
a     a

Reduced ✓

```

---
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

- $a \rightarrow b \rightarrow a$
- $b \rightarrow a \rightarrow b$
- $a \rightarrow a$
- $b \rightarrow b$

<table>
<thead>
<tr>
<th>Original</th>
<th>Reduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bullet$</td>
<td>$\bullet$</td>
</tr>
<tr>
<td>$a$</td>
<td>$a$</td>
</tr>
<tr>
<td>$b$</td>
<td>$a$</td>
</tr>
<tr>
<td>$a$</td>
<td>$a$</td>
</tr>
<tr>
<td>$b$</td>
<td>$b$</td>
</tr>
</tbody>
</table>
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 $R: S → 2^Σ$, for every $s ∈ S$

A0  $∅ ≠ R(s)$

A1  if $R(s) ≠ \text{enabled}(s)$, 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 $b ∉ R(s)$ and $b$ depends on $R(s)$, there exists an $i$ such that $a_i ∈ R(s)$

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 $∃ s_i . R(s_i) = \text{enabled}(s_i)$)

A4  if $R(s) ≠ \text{enabled}(s)$, then $|R(s)| = 1$ and the chosen action is deterministic
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:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>$\emptyset \neq R(s)$</td>
</tr>
<tr>
<td>A1</td>
<td>if $R(s) \neq \text{enabled}(s)$, then $R(s)$ contains only stuttering actions</td>
</tr>
<tr>
<td>A2</td>
<td>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 $a_i \in R(s)$</td>
</tr>
<tr>
<td>A3</td>
<td>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 $\exists s_i : R(s_i) = \text{enabled}(s_i)$)</td>
</tr>
<tr>
<td>A4</td>
<td>if $R(s) \neq \text{enabled}(s)$, then $</td>
</tr>
</tbody>
</table>
Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]

- Based on equivalent distributions and confluent transitions
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

![Diagram showing $T$-equivalent distributions with nodes $s_1, s_2, s_3, s_4, t_1, t_2, t_4$ and edges labeled with probabilities and transitions labeled 'a'.]
Confluence reduction [Timmer, Stoelinga, van de Pol, 2011]
- Based on equivalent distributions and confluent transitions

$T$-equivalent distributions
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 $(s \xrightarrow{a} t) \in T$
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 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 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

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

- Every transition 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:
**Comparison**

*Similarities* among ample sets and confluence:

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

#### Similarities among ample sets and confluence:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Size of $R(s)$</th>
<th>Reduction transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R(s) = \text{enabled}(s)$ or $</td>
<td>R(s)</td>
<td>= 1$</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>Reduction transitions</td>
<td>Deterministic and stuttering</td>
</tr>
<tr>
<td>Acyclicity</td>
<td>No cycle of reduction transitions allowed</td>
</tr>
</tbody>
</table>

Deterministic and stuttering reduction transitions and no cycle of reduction transitions allowed.
**Similarities** among ample sets and confluence:

<table>
<thead>
<tr>
<th>Similarity</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>Reduction transitions</td>
<td>Deterministic and stuttering</td>
</tr>
<tr>
<td>Acyclicity</td>
<td>No cycle of reduction transitions allowed</td>
</tr>
<tr>
<td>Preservation</td>
<td>Branching time properties</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>Reduction transitions</td>
<td>Deterministic and stuttering</td>
</tr>
<tr>
<td>Acyclicity</td>
<td>No cycle of reduction transitions allowed</td>
</tr>
<tr>
<td>Preservation</td>
<td>Branching time properties</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 $a_i \in R(s)$
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>Reduction 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 $\nu$. 
Comparison – POR implies Confluence

Theorem

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

Theorem

Let $R$ be a reduction function satisfying the ample set conditions. Then, all reduction 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 reduction 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 reduction transitions of the partial-order reduction.
2. Recursively add transitions needed to complete the confluence diamonds.
3. Proof that the resulting set is indeed confluent.
Comparison – Confluence does not imply POR
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.

Confluence Reduction versus Partial-Order Reduction

October 6, 2011
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 confluent actions to be globally independent of other actions enabled in the same states (or weaken ample sets)
Strengthening of confluence

We can change confluence in the following way:

- Do not allow **shortcuts**
- Do not allow **overlapping distributions** to be equivalent
- Require confluent actions to be **globally independent** of other actions enabled in the same states (or **weaken ample sets**)
- 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 confluent actions to be globally independent of other actions enabled in the same states (or weaken ample sets)
- Require action-separability
Theorem

*Under the strengthened notion of confluence, every acyclic confluence reduction is an ample set reduction.*
Theorem

*Under the strengthened notion of confluence, every acyclic confluence reduction is an ample set reduction.*

Corollary

*Under the above circumstances, confluence reduction and ample set reduction coincide.*
Theorem

Under the strengthened notion of confluence, every acyclic confluence reduction is an ample set reduction.

Corollary

Under the above circumstances, confluence reduction and ample set reduction coincide.

Corollary

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

State space generation using representatives:

Original

- State space generation using representatives:
- Reduction
- No need for the cycle condition anymore!
State space generation using representatives:

Original:
- States: a, b, c, d
- Transitions: a → b, b → a, b → c

Reduction:
- States: b, d
- Transitions: b → d, d → a

Reduction:
- States: b, d
- Transitions: b → d, d → a

No need for the cycle condition anymore!
Implications

State space generation using representatives:

Original

Reduction

Reduced state space compared to the original state space.
Implications

State space generation using representatives:

- Representative in **bottom strongly connected component**
- **Additional reduction** of states and transitions
- **No need for the cycle condition anymore!**
Conclusions

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