Ample Set Conditions

Lecture #8 of Advanced Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: `katoen@cs.rwth-aachen.de`

May 15, 2009

Outline of partial-order reduction

- During state space generation obtain $\widehat{TS}$
  - a *reduced version* of transition system TS such that $\widehat{TS} \triangleq TS$
  - $\Rightarrow$ this preserves all stutter sensitive LT properties, such as LTL$_{\setminus\bigcirc}$
  - at state s select a (small) subset of enabled actions in s
  - different approaches on how to select such set: consider Peled's *ample sets*
- *Static* partial-order reduction
  - obtain a high-level description of $\widehat{TS}$ (without generating TS)
  - $\Rightarrow$ POR is preprocessing phase of model checking
- *Dynamic (or: on-the-fly)* partial-order reduction
  - construct $\widehat{TS}$ during LTL$_{\setminus\bigcirc}$ model checking
  - if accept cycle is found, there is no need to generate entire $\widehat{TS}$

Independence of actions

Let *TS* $= (S, \textit{Act}, \rightarrow, I, \textit{AP}, L)$ be action-deterministic and $\alpha \neq \beta \in \textit{Act}$

- α and β are *independent* if for any $s \in S$ with $\alpha, \beta \in \textit{Act}(s)$:

$$\beta \in \textit{Act}(\alpha(s)) \quad \text{and} \quad \alpha \in \textit{Act}(\beta(s)) \quad \text{and} \quad \alpha(\beta(s)) \;=\; \beta(\alpha(s))$$

- α and β are *dependent* if α and β are not independent

- For $A \subseteq \textit{Act}$ and $\beta \in \textit{Act} \setminus A$:
  - β is independent of A if for any $\alpha \in A$, β is independent of α
  - β depends on A in *TS* if $\beta \in \textit{Act} \setminus A$ and α are dependent for some $\alpha \in A$

Stutter actions

- $\alpha \in \mathit{Act}$ is a *stutter action* if for each $s \xrightarrow{\alpha} s'$ in *TS*: $L(s) = L(s')$
  - α is a stutter action in *TS* iff $L(s) = L(\alpha(s))$ for all s in *TS* with $\alpha \in \mathit{Act}(s)$
  - α is a stutter action whenever all transitions $s \xrightarrow{\alpha} s'$ are stutter steps

Permuting independent stutter actions

Let *TS* be action-deterministic, s a state in *TS* and:

- ϱ is a finite execution in s with action sequence $\beta_1 \ldots \beta_n \, \alpha$
- ϱ' is a finite execution in s with action sequence $\alpha \, \beta_1 \ldots \beta_n$

Then:

> if α is a stutter action independent of $\{ \beta_1, \ldots, \beta_n \}$ then $\varrho \triangleq \varrho'$

Permuting independent stutter actions

$$\begin{array}{ccccccccccc} s = s_0 & \xrightarrow{\beta_1} & s_1 & \xrightarrow{\beta_2} & s_2 & \xrightarrow{\beta_3} & \ldots & \xrightarrow{\beta_{n-1}} & s_{n-1} & \xrightarrow{\beta_n} & s_n \\ \big\downarrow{\scriptstyle \alpha} & & & & & & & & & & \\ t_0 & & & & & & & & & & \end{array}$$

can be extended to

$$\begin{array}{ccccccccccc} s = s_0 & \xrightarrow{\beta_1} & s_1 & \xrightarrow{\beta_2} & s_2 & \xrightarrow{\beta_3} & \ldots & \xrightarrow{\beta_{n-1}} & s_{n-1} & \xrightarrow{\beta_n} & s_n \\ \big\downarrow{\scriptstyle \alpha} & & \big\downarrow{\scriptstyle \alpha} & & \big\downarrow{\scriptstyle \alpha} & & & & \big\downarrow{\scriptstyle \alpha} & & \big\downarrow{\scriptstyle \alpha} \\ t_0 & \xrightarrow[\beta_1]{} & t_1 & \xrightarrow[\beta_2]{} & t_2 & \xrightarrow[\beta_3]{} & \ldots & \xrightarrow[\beta_{n-1}]{} & t_{n-1} & \xrightarrow[\beta_n]{} & t_n = t \end{array}$$

Adding an independent stutter action

Let *TS* be action-deterministic, s a state in *TS* and:

- ρ is an infinite execution in s with action sequence $\beta_1 \, \beta_2 \ldots$
- ρ' is an infinite execution in s with action sequence $\alpha \, \beta_1 \, \beta_2 \ldots$

Then:

> if α is a stutter action independent of $\{ \beta_1, \beta_2, \ldots \}$ then $\rho \triangleq \rho'$

The ample-set approach

- Partial-order reduction for LT properties using *ample sets*
  - on state-space generation select $\mathit{ample}(s) \subseteq \mathit{Act}(s)$
  - such that $|\mathit{ample}(s)| \ll |\mathit{Act}(s)|$
- *Reduced* system $\widehat{TS} = (\widehat{S}, \mathit{Act}, \Longrightarrow, I, \mathit{AP}, L')$ where:
  - $\widehat{S}$ contains the states that are reachable (under $\Longrightarrow$) from some $s_0 \in I$
  - $$\frac{s \xrightarrow{\alpha} s' \;\wedge\; \alpha \in \mathit{ample}(s)}{s \overset{\alpha}{\Longrightarrow} s'}$$
  - $L'(s) = L(s)$ for any $s \in \widehat{S}$
- *Constraints*: correctness ($\triangleq$), effectivity and efficiency

Which actions to select in *ample*(s)?

(A1) **Nonemptiness condition**

Select in any state in $\widehat{TS}$ at least one action.

(A2) **Dependency condition**

For any finite execution in *TS*: an action depending on *ample*(s) can only occur after some action in *ample*(s) has occurred.

(A3) **Stutter condition**

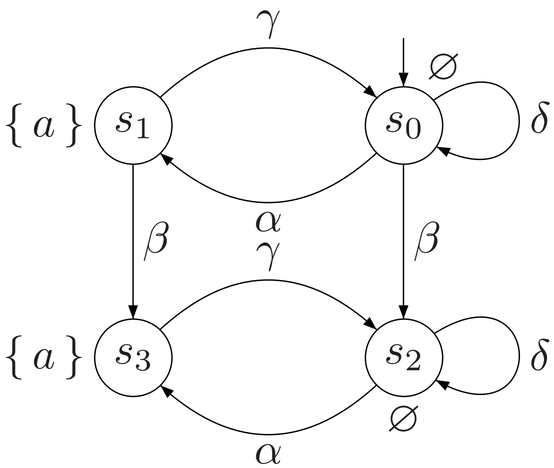
If not all actions in s are selected, then only select stutter actions in s.

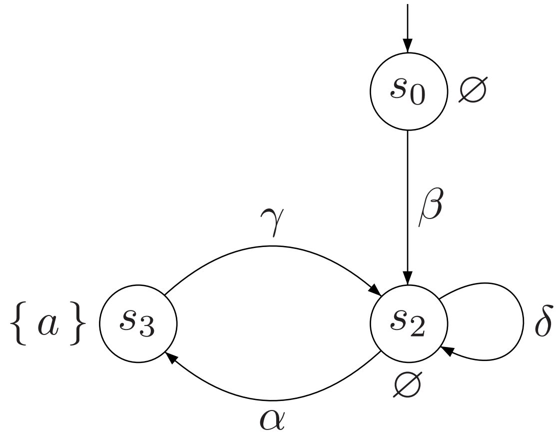
(A4) **Cycle condition**

Any action in *Act*(s_i) with s_i on a cycle in $\widehat{TS}$ must be selected in some s_j on that cycle.

(A1) through (A3) apply to states in $\widehat{S}$; (A4) to cycles in $\widehat{TS}$

Example





Nonemptiness condition

(A1) Nonemptiness condition

$\varnothing \neq \mathit{ample}(s) \subseteq \mathit{Act}(s)$

- If a state has at least one direct successor in *TS*, then it has least at one direct successor in $\widehat{TS}$

$\Rightarrow$ As *TS* has no terminal states, $\widehat{TS}$ has no terminal states

Dependency condition

(A2) Dependency condition

Let $s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_n} s_n \xrightarrow{\alpha} t$ be a finite execution in *TS* such that α depends on *ample*(s).

Then: $\beta_i \in$ *ample*(s) for some $0 < i \leqslant n$.

- In every (!) finite execution fragment of *TS*, an action depending on *ample*(s) cannot occur before some action from *ample*(s) occurs first
- (A2) ensures that for any state s with *ample*$(s) \subset$ *Act*(s), any $\alpha \in$ *ample*(s) is independent of *Act*$(s) \setminus$ *ample*(s)

Properties

- (A2) guarantees that any finite execution in *TS* is of the form:

$$\varrho \;=\; s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_n} s_n \xrightarrow{\alpha} t \quad \text{with} \quad \alpha \in \textit{ample}(s)$$

  and β_i independent of *ample*(s) for $0 < i \leqslant n$.

  – if α is a stutter action: shifting α to the beginning yields an equivalent execution

  $\Rightarrow$ if ϱ is pruned in *TS*, then an execution is obtained by first taking α in s

- (A2) guarantees that any infinite execution in *TS* is of the form:

$$s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} s_2 \ldots \quad \text{with } \beta_i \text{ independent of } \textit{ample}(s) \text{ for } 0 < i \leqslant n.$$

  – performing stutter action $\alpha \in$ *ample*(s) in s yields an equivalent execution

Properties

For any $\alpha \in ample(s)$ and $s \in Reach(\widehat{TS})$:

if $ample(s)$ satisfies (A2) then α is independent of $Act(s) \setminus ample(s)$

For finite execution $s = s_0 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_n} s_n$ in TS and $s \in Reach(\widehat{TS})$:

if $ample(s)$ satisfies (A2) and $\{ \beta_1, \ldots, \beta_n \} \cap ample(s) = \varnothing$, then:

α is independent of $\{ \beta_1, \ldots, \beta_n \}$ and $\alpha \in Act(s_i)$ for $0 \leqslant i \leqslant n$

A too simplistic dependency condition (1)

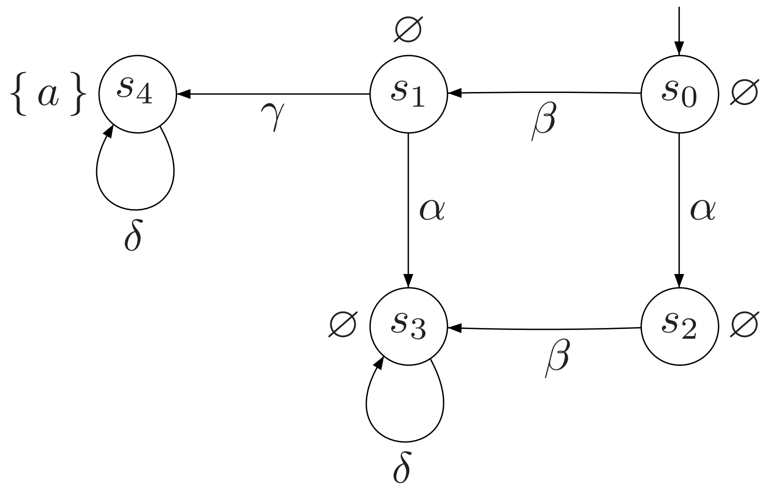
(A2')

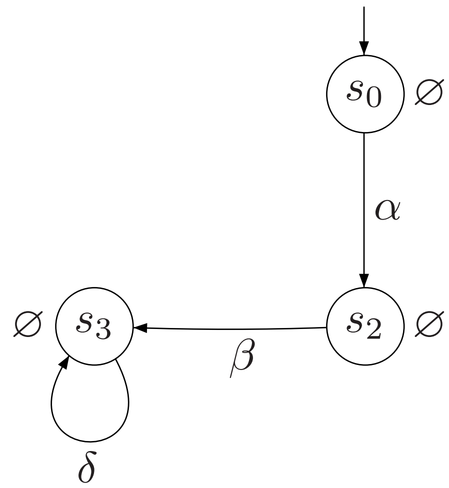
If *ample*$(s) \neq$ *Act*(s)

then $\alpha \in$ *ample*(s) is independent of *Act*$(s) \setminus$ *ample*(s).

this is a consequence of (A2), but in itself too weak: cf. next example

A too simplistic dependency condition (2)





Stutter condition

(A3) Stutter condition

If *ample*$(s) \neq$ *Act*(s) then any $\alpha \in$ *ample*(s) is a stutter action.

- All ample actions of a non-fully expanded state are stutter actions
- (A3) ensures that:
  - changing $\beta_1 \,, \ldots \, \beta_n \, \alpha$ into $\alpha \, \beta_1 \, \ldots \, \beta_n$, and
  - changing $\beta_1 \, \beta_2 \, \beta_3 \ldots$ into $\alpha \, \beta_1 \, \beta_2 \, \beta_3 \ldots$

  yields stutter-equivalent executions

Correctness of transformation (1)

Let ϱ be a finite execution fragment in *Reach*(*TS*) of the form

$$s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_n} s_n \xrightarrow{\alpha} t$$

where $\beta_i \notin$ *ample*(s), for $0 < i \leqslant n$, and $\alpha \in$ *ample*(s).

If *ample*(s) satisfies (A1) through (A3), then there exists an execution fragment ϱ':

$$s \stackrel{\alpha}{\Longrightarrow} t_0 \xrightarrow{\beta_1} t_1 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_{n-1}} t_{n-1} \xrightarrow{\beta_n} t$$

such that $\boxed{\varrho \triangleq \varrho'}$

Proof

Correctness of transformation (2)

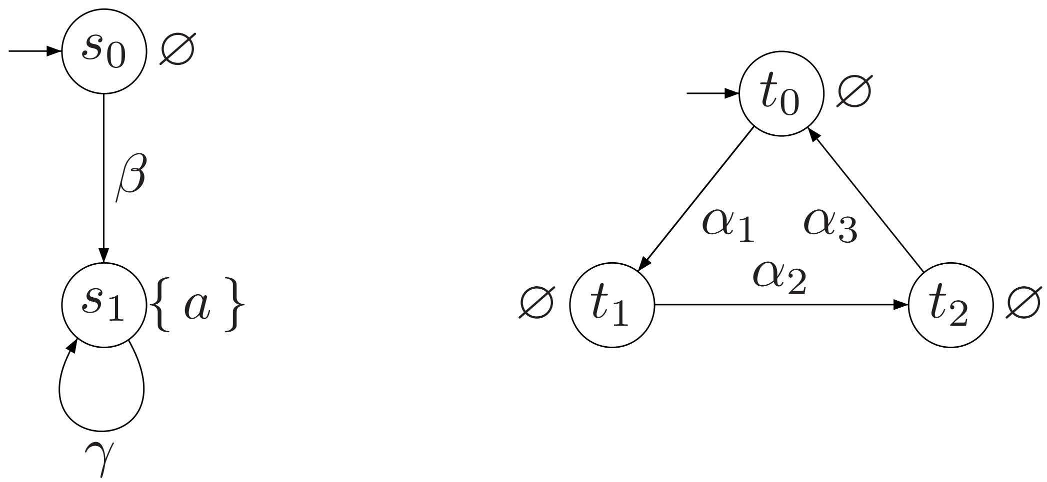
Let $\rho = s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} s_2 \xrightarrow{\beta_3} \ldots$ be an infinite execution fragment in *Reach*(*TS*) where $\beta_i \notin$ *ample*(s), for $i > 0$.

If *ample*(s) satisfies (A1) through (A3), then there exists an execution fragment ρ':

$$s \overset{\alpha}{\Longrightarrow} t_0 \xrightarrow{\beta_1} t_1 \xrightarrow{\beta_2} t_2 \xrightarrow{\beta_3} \ldots$$

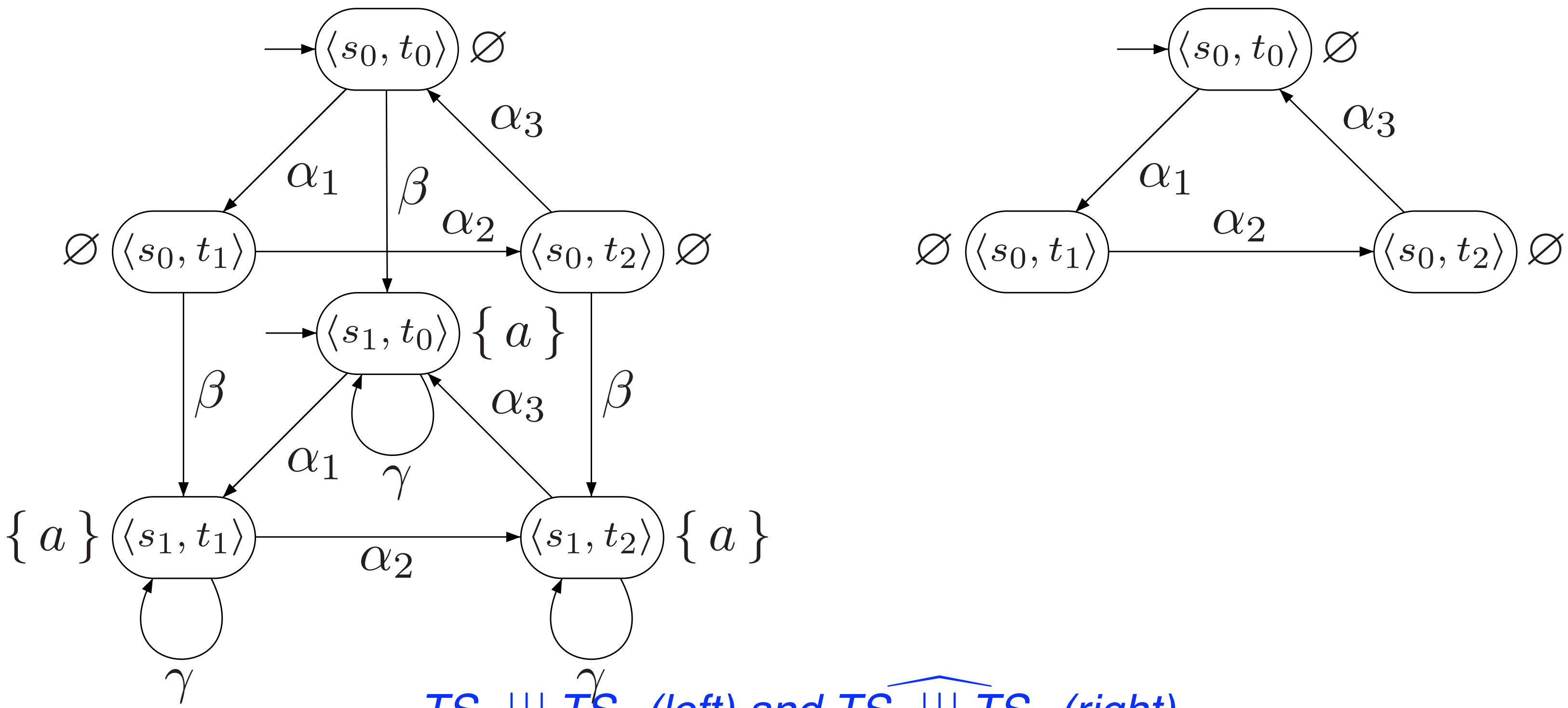
where $\alpha \in$ *ample*(s) and $\boxed{\rho \triangleq \rho'}$

Necessity of cycle condition: example (1)



transition systems TS_1 and TS_2

Necessity of cycle condition: example (2)



$TS_1 \,|||\, TS_2$ (left) and $\widehat{TS_1 \,|||\, TS_2}$ (right)

$TS_1 \,|||\, TS_2 \not\models \Box\neg a$ but $\widehat{TS_1 \,|||\, TS_2} \models \Box\neg a$

Cycle condition

(A4) Cycle condition

For any cycle $s_0\, s_1\, \ldots\, s_n$ in $\widehat{TS}$ and $\alpha \in Act(s_i)$, for some $0 < i \leqslant n$, there exists $j \in \{1, \ldots, n\}$ such that $\alpha \in ample(s_j)$.

any enabled action in some state on a cycle must be selected in some state on that cycle

Overview of ample-set conditions

(A1) **Nonemptiness condition**

$\varnothing \neq \mathit{ample}(s) \subseteq \mathit{Act}(s)$

(A2) **Dependency condition**

Let $s \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_n} s_n \xrightarrow{\alpha} t$ be a finite execution fragment in *TS* such that α depends on $\mathit{ample}(s)$. Then: $\beta_i \in \mathit{ample}(s)$ for some $0 < i \leqslant n$.

(A3) **Stutter condition**

If $\mathit{ample}(s) \neq \mathit{Act}(s)$ then any $\alpha \in \mathit{ample}(s)$ is a stutter action.

(A4) **Cycle condition**

For any cycle $s_0\, s_1 \, \ldots \, s_n$ in $\widehat{TS}$ and $\alpha \in \mathit{Act}(s_i)$, for some $0 < i \leqslant n$, there exists $j \in \{ 1, \ldots, n \}$ such that $\alpha \in \mathit{ample}(s_j)$.

Correctness theorem

For action-deterministic, finite TS without terminal states:

if conditions (A1) through (A4) are satisfied, then $\widehat{TS} \triangleq TS$.

as $Traces(\widehat{TS}) \subseteq Traces(TS)$, it follows $\widehat{TS} \trianglelefteq TS$

Proof