Stutter Trace and Bisimulation Equivalence
Lecture #5 of Advanced Model Checking

Joost-Pieter Katoen
Lehrstuhl 2: Software Modeling & Verification
E-mail: katoen@cs.rwth-aachen.de

April 29, 2009
Motivation

- Bisimulation, simulation and trace equivalence are strong
  - each transition $s \rightarrow s'$ must be matched by a transition of a related state
  - for comparing models at different abstraction levels, this is too fine
  - consider e.g., modeling an abstract action by a sequence of concrete actions

- Idea: allow for sequences of “invisible” actions
  - each transition $s \rightarrow s'$ must be matched by a path fragment of a related state
  - matching means: ending in a state related to $s'$, and all previous states invisible

- Abstraction of such internal computations yields coarser quotients
  - but: what kind of properties are preserved?
  - but: can such quotients still be obtained efficiently?
  - but: how to treat infinite internal computations?
Motivating example

Let $TS_{conc}$ model the concrete program fragment

\[
\begin{align*}
  i &:= y; z := 1; \\
  \textbf{while} \ i > 1 \ \textbf{do} & \\
  & \quad z := z \times i; i := i - 1; \\
  \textbf{od} & \\
  x &:= z;
\end{align*}
\]

that computes the factorial of $y$ iteratively.

Let $TS_{abs}$ be the transition system of the (abstract) program $x := y!$

Clearly, $TS_{abs}$ and $TS_{conc}$ are in some sense equivalent
### Outlook of today’s lecture

<table>
<thead>
<tr>
<th>formal relation</th>
<th>trace equivalence</th>
<th>bisimulation</th>
<th>simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>complexity</td>
<td>PSPACE-complete</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>logical fragment</td>
<td>LTL</td>
<td>CTL*</td>
<td>∀CTL*</td>
</tr>
<tr>
<td>preservation</td>
<td>strong</td>
<td>strong match</td>
<td>weak match</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>formal relation</th>
<th>stutter trace equivalence</th>
<th>stutter bisimulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>complexity</td>
<td>PSPACE-complete</td>
<td>PTIME</td>
</tr>
<tr>
<td>logical fragment</td>
<td>LTL(\neg)</td>
<td>–</td>
</tr>
<tr>
<td>preservation</td>
<td>strong</td>
<td>–</td>
</tr>
</tbody>
</table>
Stuttering equivalence

- \( s \rightarrow s' \) in transition system \( TS \) is a **stutter step** if \( L(s) = L(s') \)
  - stutter steps do not affect the state labels of successive states

- Paths \( \pi_1 \) and \( \pi_2 \) are **stuttering equivalent**, denoted \( \pi_1 \triangleq \pi_2 \):
  - if there exists an infinite sequence \( A_0A_1A_2 \ldots \) with \( A_i \subseteq AP \) and
  - natural numbers \( n_0, n_1, n_2, \ldots, m_0, m_1, m_2, \ldots \geq 1 \) such that:

  \[
  \begin{align*}
  trace(\pi_1) &= A_0 \ldots A_0 \underbrace{A_1 \ldots A_1}_{n_1\text{-times}} \underbrace{A_2 \ldots A_2}_{n_2\text{-times}} \ldots \\
  trace(\pi_2) &= A_0 \ldots A_0 \underbrace{A_1 \ldots A_1}_{m_1\text{-times}} \underbrace{A_2 \ldots A_2}_{m_2\text{-times}} \ldots 
  \end{align*}
  \]

  \( \Rightarrow \) \( \pi_1 \triangleq \pi_2 \) if their traces only differ in their stutter steps

  \( \Rightarrow \) i.e., if both their traces are of the form \( A_0^+A_1^+A_2^+ \ldots \) for \( A_i \subseteq AP \)
Semaphore-based mutual exclusion
Stutter equivalent traces

the following two infinite paths in $TS_{Sem}$:

$$\pi_1 = \langle n_1, n_2 \rangle \rightarrow \langle w_1, n_2 \rangle \rightarrow \langle w_1, w_2 \rangle \rightarrow \langle c_1, w_2 \rangle \rightarrow \langle n_1, w_2 \rangle \rightarrow$$
$$\langle n_1, c_2 \rangle \rightarrow \langle n_1, n_2 \rangle \rightarrow \langle w_1, n_2 \rangle \rightarrow \langle w_1, w_2 \rangle \rightarrow \langle c_1, w_2 \rangle \rightarrow \ldots$$

$$\pi_2 = \langle n_1, n_2 \rangle \rightarrow \langle w_1, n_2 \rangle \rightarrow \langle c_1, n_2 \rangle \rightarrow \langle c_1, w_2 \rangle \rightarrow \langle n_1, w_2 \rangle \rightarrow$$
$$\langle w_1, w_2 \rangle \rightarrow \langle w_1, c_2 \rangle \rightarrow \langle w_1, n_2 \rangle \rightarrow \langle c_1, n_2 \rangle \rightarrow \ldots$$

Hence, $\pi_1 \triangleq \pi_2$, since for $AP = \{ \text{crit}_1, \text{crit}_2 \}$:

$$\text{trace}(\pi_1) = \emptyset^3 \{ \text{crit}_1 \} \emptyset \{ \text{crit}_2 \} \emptyset^3 \{ \text{crit}_1 \} \ldots \text{ and}$$

$$\text{trace}(\pi_2) = \emptyset^2 (\{ \text{crit}_1 \})^2 \emptyset^2 \{ \text{crit}_2 \} \emptyset \{ \text{crit}_1 \} \ldots$$
Pictorially

\[
\begin{array}{cccccccc}
 n_1 & n_2 & w_1 & n_2 & w_1 & w_2 & c_1 & w_2 \\
 0 & 0 & 0 & \{c_1\} & 0 & \{c_2\} & 0 & 0 \\
 n_1 & n_2 & w_1 & n_2 & c_1 & n_2 & c_1 & w_2 \\
 0 & 0 & \{c_1\} & \{c_1\} & 0 & 0 & \{c_2\} & 0 \\
 \end{array}
\]
Stutter trace equivalence

Transition systems $TS_i$ over $AP$, $i=1, 2$, are \textit{stutter-trace equivalent}:

$$TS_1 \triangleq TS_2 \iff TS_1 \subseteq TS_2 \text{ and } TS_2 \subseteq TS_1$$

where $\subseteq$, pronounced \textit{stutter trace inclusion}, is defined by:

$$TS_1 \subseteq TS_2 \iff \forall \sigma_1 \in \text{Traces}(TS_1) \left( \exists \sigma_2 \in \text{Traces}(TS_2). \sigma_1 \triangleq \sigma_2 \right)$$

$$\text{Traces}(TS_1) = \text{Traces}(TS_2) \text{ implies } TS_1 \triangleq TS_2, \text{ but not always the converse}$$
Example

\[ TS_1 \triangleq TS_2, \quad TS_1 \not\leq TS_3 \text{ and } TS_2 \not\leq TS_3, \text{ but } TS_3 \leq TS_2 \text{ and } TS_3 \leq TS_1 \]
The $\bigcirc$ operator

Stuttering equivalence does not preserve the validity of next-formulas:

\[ \sigma_1 = ABBB \ldots \text{ and } \sigma_2 = AAABBBB \ldots \] for $A, B \subseteq AP$ and $A \neq B$

Then for $b \in B \setminus A$:

\[ \sigma_1 \triangleq \sigma_2 \quad \text{but} \quad \sigma_1 \models \bigcirc b \quad \text{and} \quad \sigma_2 \not\models \bigcirc b. \]

$\Rightarrow$ a logical characterization of $\triangleq$ can only be obtained by omitting $\bigcirc$

in fact, it turns out that this is the only modal operator that is not preserved by $\triangleq$!
Stutter trace and LTL\(\Box\) equivalence

For traces \(\sigma_1\) and \(\sigma_2\) over \(2^{AP}\) it holds:
\[\sigma_1 \triangleq \sigma_2 \Rightarrow (\sigma_1 \models \varphi \text{ if and only if } \sigma_2 \models \varphi)\]
for any LTL\(\Box\) formula \(\varphi\) over \(AP\)

LTL\(\Box\) denotes the class of LTL formulas without the next step operator \(\bigcirc\)
Proof
Stutter trace and $\text{LTL}\setminus\text{O}$ equivalence

For transition systems $TS_1$, $TS_2$ (over $AP$) without terminal states:

(a) $TS_1 \triangleq TS_2$ implies $\left( TS_1 \equiv_{\text{LTL}\setminus\text{O}} TS_2 \right)$

(b) if $TS_1 \preceq TS_2$ then for any $\text{LTL}\setminus\text{O}$ formula $\varphi$: $TS_2 \models \varphi$ implies $TS_1 \models \varphi$

A more general result can be established by considering stutter-insensitive LT properties . . . . . .
Stutter insensitivity

- LT property $P$ is **stutter-insensitive** if $[\sigma] \triangleq \subseteq P$, for any $\sigma \in P$
  - $P$ is stutter insensitive if it is closed under stutter equivalence

- For any stutter-insensitive LT property $P$:
  \[
  TS_1 \triangleq TS_2 \quad \text{implies} \quad (TS_1 \models P \text{ iff } TS_2 \models P)
  \]

- Moreover: $TS_1 \preceq TS_2$ implies $(TS_2 \models P \text{ implies } TS_1 \models P)$

- For any LTL $\emptyset$ formula $\varphi$, LT property $Words(\varphi)$ is stutter insensitive
  - but: some stutter insensitive LT properties cannot be expressed in LTL $\emptyset$
  - for LTL formula $\varphi$ with $Words(\varphi)$ stutter insensitive:
    - there exists $\psi \in LTL\emptyset$ such that $\psi \equiv_{LTL} \varphi$
Stutter bisimulation

Let $TS = (S, \text{Act}, \rightarrow, I, AP, L)$ be a transition system and $\mathcal{R} \subseteq S \times S$.

$\mathcal{R}$ is a **stutter-bisimulation** for $TS$ if for all $(s_1, s_2) \in \mathcal{R}$:

1. $L(s_1) = L(s_2)$

2. if $s'_1 \in \text{Post}(s_1)$ with $(s_1, s'_1) \not\in \mathcal{R}$, then there exists a finite path fragment $s_2 u_1 \ldots u_n s'_2$ with $n \geq 0$ and $(s_2, u_i) \in \mathcal{R}$ and $(s'_1, s'_2) \in \mathcal{R}$

3. if $s'_2 \in \text{Post}(s_2)$ with $(s_2, s'_2) \not\in \mathcal{R}$, then there exists a finite path fragment $s_1 v_1 \ldots v_n s'_1$ with $n \geq 0$ and $(s_1, v_i) \in \mathcal{R}$ and $(s'_1, s'_2) \in \mathcal{R}$

$s_1, s_2$ are **stutter-bisimulation equivalent**, denoted $s_1 \approx_{TS} s_2$, if there exists a stutter bisimulation $\mathcal{R}$ for $TS$ with $(s_1, s_2) \in \mathcal{R}$.
Stutter bisimulation

\( s_1 \approx s_2 \)
\( \downarrow \)
\( s_1' \) (with \( s_1 \not\approx s_1' \))

\( s_1 \approx s_2 \)
\( \downarrow \)
\( s_1 \approx u_1 \)
\( \downarrow \)
\( s_1 \approx u_2 \)
\( \downarrow \)
\( s_1 \approx u_n \)
\( \downarrow \)
\( s_1' \approx s_2' \)
Advanced model checking

Semaphore-based mutual exclusion

stutter-bisimilar states for \( AP = \{ \text{crit}_1, \text{crit}_2 \} \)
Stutter-bisimilar transition systems

Let \( TS_i = (S_i, Act_i, \rightarrow_i, I_i, AP, L_i) \), \( i = 1, 2 \), be transition systems \( TS_1 \) and \( TS_2 \) are stutter bisimilar, denoted \( TS_1 \approx TS_2 \), if there exists a stutter bisimulation \( R \) on \( TS_1 \oplus TS_2 \) such that:

\[
\forall s_1 \in I_1. (\exists s_2 \in I_2. (s_1, s_2) \in R) \quad \text{and} \quad \forall s_2 \in I_2. (\exists s_1 \in I_1. (s_1, s_2) \in R)
\]
Stutter bisimulation quotient

Let $TS = (S, Act, \rightarrow, I, AP, L)$ and stutter bisimulation $\mathcal{R} \subseteq S \times S$ be an equivalence.

The quotient of $TS$ under $\mathcal{R}$ is defined by:

$$TS/\mathcal{R} = (S', \{ \tau \}, \rightarrow', I', AP, L')$$

where

- $S' = S/\mathcal{R} = \{ [s]_\mathcal{R} \mid s \in S \}$ with $[s]_\mathcal{R} = \{ s' \in S \mid (s, s') \in \mathcal{R} \}$
- $I' = \{ [s]_\mathcal{R} \mid s \in I \}$
- $L'([s]_\mathcal{R}) = L(s)$
- $\rightarrow'$ is defined by:

$$s \xrightarrow{\alpha} s' \text{ and } (s, s') \notin \mathcal{R} \quad \Rightarrow \quad [s]_\mathcal{R} \xrightarrow{\tau'} [s']_\mathcal{R}$$

note that (a) no self-loops occur in $TS/\approx_{TS}$ and (b) $TS \approx TS/\approx_{TS}$
Semaphore-based mutual exclusion

The stutter-bisimulation quotient:
Stutter trace and stutter bisimulation

For transition systems $TS_1$ and $TS_2$ over $AP$:

- Known fact: $TS_1 \sim TS_2$ implies $Traces(TS_1) = Traces(TS_2)$

- But $not$: $TS_1 \approx TS_2$ implies $TS_1 \triangleq TS_2$!

- So:
  - bisimilar transition systems are trace equivalent
  - but stutter-bisimilar transition systems are not always stutter trace-equivalent!

- Why? Stutter paths!
  - stutter bisimulation does not impose any constraint on such paths
  - but $\triangleq$ requires the existence of a stuttering equivalent trace
Stutter trace and stutter bisimulation are incomparable

\[ \varnothing \not\approx \sim \triangleleft \]

\[ TS_1 \quad TS_2 \quad TS_3 \quad TS_4 \]
Stutter bisimulation does not preserve LTL

\[ TS_{\text{left}} \approx TS_{\text{right}} \text{ but } TS_{\text{left}} \not\models \Diamond a \text{ and } TS_{\text{right}} \models \Diamond a \]

reason: presence of infinite stutter paths in \( TS_{\text{left}} \)
Divergence sensitivity

- **Stutter paths** are paths that only consist of stutter steps
  - no restrictions are imposed on such paths by a stutter bisimulation
  ⇒ stutter trace-equivalence (≜) and stutter bisimulation (∼) are incomparable
  ⇒ ∼ and LTL\(\bigcirc\) equivalence are incomparable

- Stutter paths **diverge**: they never leave an equivalence class

- Remedy: only relate **divergent** states or **non-divergent** states
  - divergent state = a state that has a stutter path
  ⇒ relate states only if they either both have stutter paths or none of them

- This yields **divergence-sensitive stutter bisimulation** (∼\text{div})
  ⇒ ∼\text{div} is strictly finer than Δ (and ∼)
## Outlook

<table>
<thead>
<tr>
<th>Formal Relation</th>
<th>Trace Equivalence</th>
<th>Bisimulation</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>PSPACE-complete</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Logical Fragment</td>
<td>LTL</td>
<td>CTL*</td>
<td>∀CTL*</td>
</tr>
<tr>
<td>Preservation</td>
<td>Strong</td>
<td>Strong Match</td>
<td>Weak Match</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formal Relation</th>
<th>Stutter Trace Equivalence</th>
<th>Divergence-sensitive Stutter Bisimulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>PSPACE-complete</td>
<td>PTIME</td>
</tr>
<tr>
<td>Logical Fragment</td>
<td>LTL(\bigcirc)</td>
<td>CTL*(\bigcirc)</td>
</tr>
<tr>
<td>Preservation</td>
<td>Strong</td>
<td>Strong Match</td>
</tr>
</tbody>
</table>
Divergence sensitivity

Let $TS$ be a transition system and $\mathcal{R}$ an equivalence relation on $S$

- $s$ is $\mathcal{R}$-divergent if there exists an infinite path fragment $s \ s_1 \ s_2 \ldots \in \text{Paths}(s)$ such that $(s, s_j) \in \mathcal{R}$ for all $j > 0$
  - $s$ is $\mathcal{R}$-divergent if there is an infinite path starting in $s$ that only visits $[s]_{\mathcal{R}}$

- $\mathcal{R}$ is divergence sensitive if for any $(s_1, s_2) \in \mathcal{R}$:
  - $s_1$ is $\mathcal{R}$-divergent implies $s_2$ is $\mathcal{R}$-divergent
  - $\mathcal{R}$ is divergence-sensitive if in any $[s]_{\mathcal{R}}$ either all or none states are $\mathcal{R}$-divergent
Divergent-sensitive stutter bisimulation

$s_1, s_2$ are \textit{divergent-sensitive stutter-bisimilar}, denoted $s_1 \approx_{\text{div}}^{TS} s_2$, if:

\[ \exists \text{ divergent-sensitive stutter bisimulation } \mathcal{R} \text{ on } TS \text{ such that } (s_1, s_2) \in \mathcal{R} \]

\( \approx_{\text{div}}^{TS} \) is an equivalence, the coarsest divergence-sensitive stutter bisimulation for \( TS \)
and the union of all divergence-sensitive stutter bisimulations for \( TS \)
Example
Quotient transition system under $\approx^{\text{div}}$

$TS/\approx^{\text{div}} = (S', \{\tau\}, \to', I', AP, L')$, the quotient of $TS$ under $\approx^{\text{div}}$

where

• $S'$, $I'$ and $L'$ are defined as usual (for eq. classes $[s]_{\text{div}}$ under $\approx^{\text{div}}$)

• $\to'$ is defined by:

\[
\frac{s \xrightarrow{\alpha} s' \land s \not\approx^{\text{div}} s'}{[s]_{\text{div}} \xrightarrow{\tau}' [s']_{\text{div}}}
\quad \text{and} \quad
\frac{s \text{ is } \approx^{\text{div}}\text{-divergent}}{[s]_{\text{div}} \not\xrightarrow{\tau}' [s]_{\text{div}}}
\]

note that $TS \approx^{\text{div}} TS/\approx^{\text{div}}$
Example

transition system $TS$

transition system $TS/\sim$

transition system $TS/\sim^{div}$
A remark on purely divergent states

- \( s_{pd} \) is purely divergent if all paths of \( s \) are infinite and divergent.
- \( s_{term} \) is a terminal state if it has no outgoing transitions.
- If \( L(s_{pd}) = L(s_{term}) \) then \( s_{term} \simeq_{TS} s_{pd} \) and \( s_{term} \not\simeq_{TS}^{div} s_{pd} \).
- \( s_{term} \simeq_{TS}^{div} s \) implies
  - \( L(s) = L(s_{term}) \) and each path of \( s \) is finite and divergent.
Summary

**stutter trace inclusion:**
\[ TS_1 \preceq TS_2 \iff \forall \sigma_1 \in \text{Traces}(TS_1) \exists \sigma_2 \in \text{Traces}(TS_2). \sigma_1 \triangleq \sigma_2 \]

**stutter trace equivalence:**
\[ TS_1 \triangleq TS_2 \iff TS_1 \preceq TS_2 \text{ and } TS_2 \preceq TS_1 \]

**stutter bisimulation equivalence:**
\[ TS_1 \approx TS_2 \iff \text{there exists a stutter bisimulation for } (TS_1, TS_2) \]

**stutter bisimulation equivalence with divergence:**
\[ TS_1 \approx^{\text{div}} TS_2 \iff \text{there exists a divergence-sensitive stutter bisimulation for } (TS_1, TS_2) \]
Relationship between equivalences

bisimulation \( TS_1 \sim TS_2 \)

divergence sensitive
stutter bisimulation \( TS_1 \approx_{\text{div}} TS_2 \)

stutter bisimulation \( TS_1 \approx TS_2 \)

trace equivalence \( \text{Traces}(T_1) = \text{Traces}(TS_2) \)

stutter trace-equivalence \( TS_1 \triangleq TS_2 \)

trace inclusion \( \text{Traces}(T_1) \subseteq \text{Traces}(TS_2) \)

stutter trace inclusion \( TS_1 \trianglelefteq TS_2 \)