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 $s \longrightarrow s'$ 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 $s \longrightarrow s'$ 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

$i := y; z := 1;$
while $i > 1$ **do**
  $z := z * i; i := i - 1;$
od
$x := z;$

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

formal relation	trace equivalence	bisimulation	simulation
complexity	PSPACE-complete	PTIME	PTIME
logical fragment	LTL	CTL*	$\forall$CTL*
preservation	strong	strong match	weak match

formal relation	stutter trace equivalence	stutter bisimulation
complexity	PSPACE-complete	PTIME
logical fragment	LTL$_{\setminus\bigcirc}$	–
preservation	strong	–

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

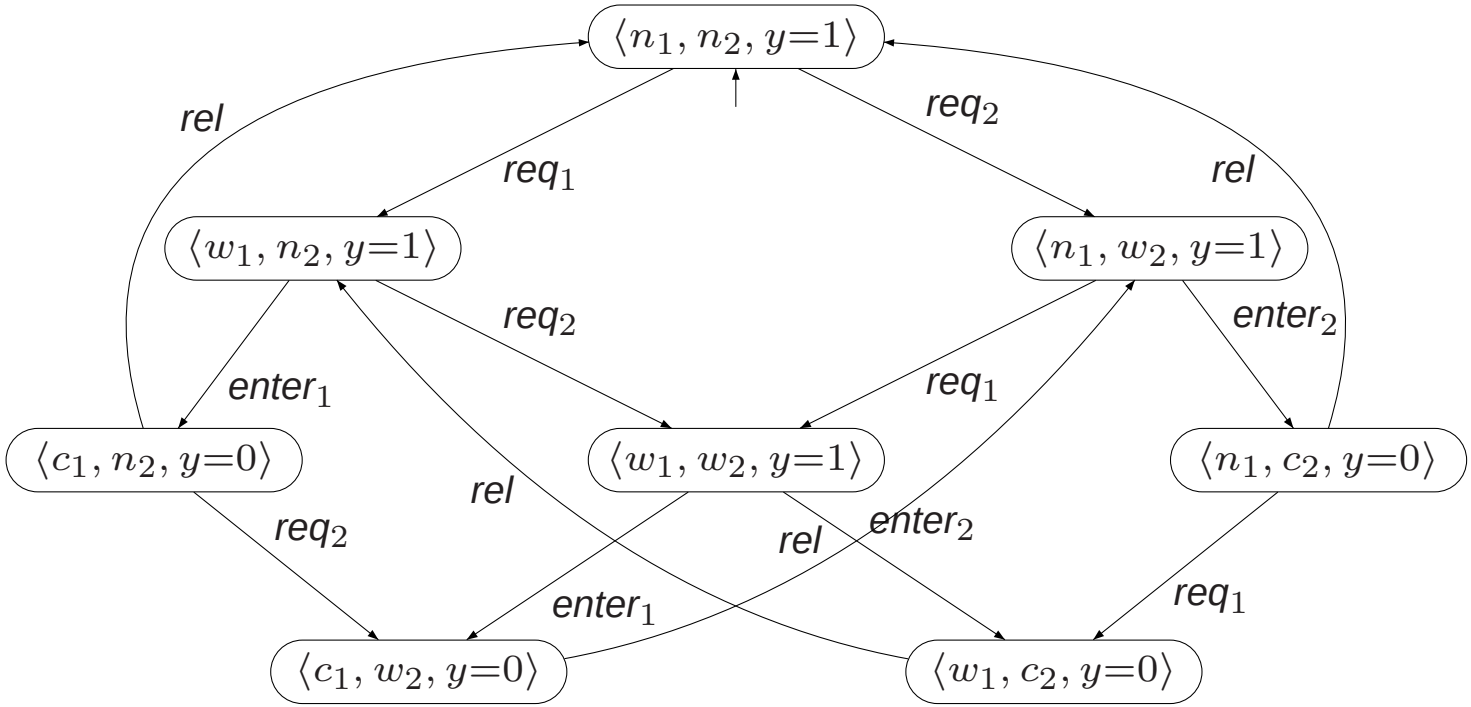
- Paths π_1 and π_2 are *stuttering equivalent*, denoted $\pi_1 \stackrel{\triangle}{=} \pi_2$:
  - if there exists an infinite sequence $A_0 A_1 A_2 \ldots$ with $A_i \subseteq AP$ and
  - natural numbers $n_0, n_1, n_2, \ldots, m_0, m_1, m_2, \ldots \geqslant 1$ such that:

$$\begin{aligned} \mathit{trace}(\pi_1) &= \underbrace{A_0 \ldots A_0}_{n_0\text{-times}} \underbrace{A_1 \ldots A_1}_{n_1\text{-times}} \underbrace{A_2 \ldots A_2}_{n_2\text{-times}} \ldots \\ \mathit{trace}(\pi_2) &= \underbrace{A_0 \ldots A_0}_{m_0\text{-times}} \underbrace{A_1 \ldots A_1}_{m_1\text{-times}} \underbrace{A_2 \ldots A_2}_{m_2\text{-times}} \ldots \end{aligned}$$

$\Rightarrow$ $\pi_1 \stackrel{\triangle}{=} \pi_2$ 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

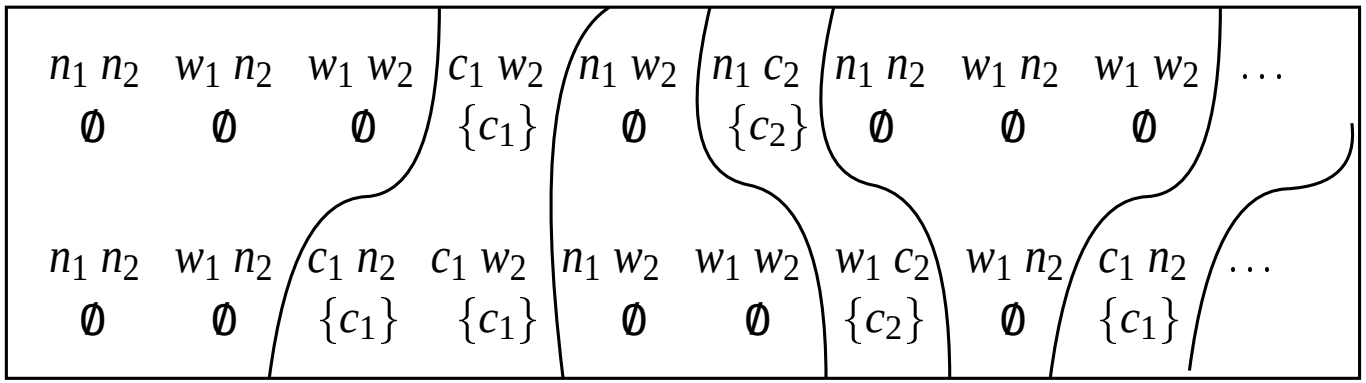
the following two infinite paths in TS_{Sem}:

$$\begin{aligned}
\pi_1 \quad = \quad & \langle n_1, n_2 \rangle \to \langle w_1, n_2 \rangle \to \langle w_1, w_2 \rangle \to \langle c_1, w_2 \rangle \to \langle n_1, w_2 \rangle \to \\
& \langle n_1, c_2 \rangle \to \langle n_1, n_2 \rangle \to \langle w_1, n_2 \rangle \to \langle w_1, w_2 \rangle \to \langle c_1, w_2 \rangle \to \ldots \\
\pi_2 \quad = \quad & \langle n_1, n_2 \rangle \to \langle w_1, n_2 \rangle \to \langle c_1, n_2 \rangle \to \langle c_1, w_2 \rangle \to \langle n_1, w_2 \rangle \to \\
& \langle w_1, w_2 \rangle \to \langle w_1, c_2 \rangle \to \langle w_1, n_2 \rangle \to \langle c_1, n_2 \rangle \to \ldots
\end{aligned}$$

Hence, $\pi_1 \stackrel{\triangle}{=} \pi_2$, since for $AP = \{ \mathit{crit}_1, \mathit{crit}_2 \}$:

$$\begin{aligned}
\mathit{trace}(\pi_1) \quad &= \quad \varnothing^3 \{ \mathit{crit}_1 \} \varnothing \{ \mathit{crit}_2 \} \varnothing^3 \{ \mathit{crit}_1 \} \ldots \quad \text{and} \\
\mathit{trace}(\pi_2) \quad &= \quad \varnothing^2 (\{ \mathit{crit}_1 \})^2 \varnothing^2 \{ \mathit{crit}_2 \} \varnothing \{ \mathit{crit}_1 \} \ldots
\end{aligned}$$

Pictorially



Stutter trace equivalence

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

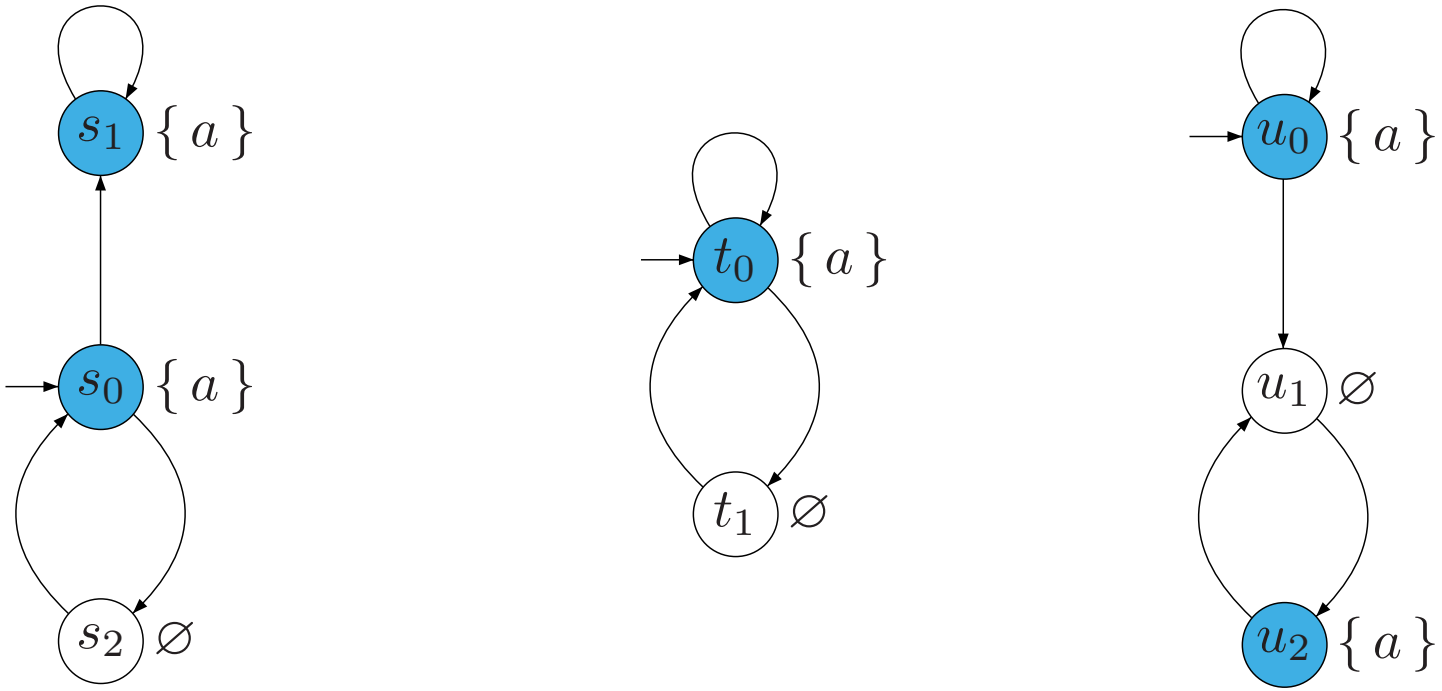
$$TS_1 \triangleq TS_2 \quad \text{if and only if} \quad TS_1 \trianglelefteq TS_2 \text{ and } TS_2 \trianglelefteq TS_1$$

where $\trianglelefteq$, pronounced *stutter trace inclusion*, is defined by:

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

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

Example



$TS_1 \triangleq TS_2$, $TS_1 \not\trianglelefteq TS_3$ and $TS_2 \not\trianglelefteq TS_3$, but $TS_3 \trianglelefteq TS_2$ and $TS_3 \trianglelefteq TS_1$

The $\bigcirc$ operator

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

$\sigma_1 = A\,B\,B\,B\ldots$ and $\sigma_2 = A\,A\,A\,B\,B\,B\,B\ldots$ for $A, B \subseteq AP$ and $A \neq B$

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

$$\sigma_1 \stackrel{\triangle}{=} \sigma_2 \quad \text{but} \quad \sigma_1 \models \bigcirc b \quad \text{and} \quad \sigma_2 \not\models \bigcirc b.$$

$\Rightarrow$ a logical characterization of $\stackrel{\triangle}{=}$ can only be obtained by omitting $\bigcirc$

in fact, it turns out that this is the only modal operator that is not preserved by $\stackrel{\triangle}{=}$!

Stutter trace and $\mathrm{LTL}_{\setminus\bigcirc}$ equivalence

For traces σ_1 and σ_2 over 2^{AP} it holds:

$$\sigma_1 \stackrel{\triangle}{=} \sigma_2 \;\Rightarrow\; (\sigma_1 \models \varphi \text{ if and only if } \sigma_2 \models \varphi)$$

for any $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ over *AP*

$\mathrm{LTL}_{\setminus\bigcirc}$ denotes the class of LTL formulas without the next step operator $\bigcirc$

Proof

Stutter trace and LTL$_{\setminus\bigcirc}$ equivalence

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

(a) $TS_1 \triangleq TS_2$ implies $\Big(TS_1 \equiv_{LTL\setminus\bigcirc} TS_2\Big)$

(b) if $TS_1 \trianglelefteq TS_2$ then for any LTL$_{\setminus\bigcirc}$ formula φ: $TS_2 \models \varphi$ 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: $TS_1 \trianglelefteq TS_2$ implies $(TS_2 \models P \quad \text{implies} \quad TS_1 \models P)$
- For any $\text{LTL}_{\setminus\bigcirc}$ formula φ, LT property $\mathit{Words}(\varphi)$ is stutter insensitive
  - but: some stutter insensitive LT properties cannot be expressed in $\text{LTL}_{\setminus\bigcirc}$
  - for LTL formula φ with $\mathit{Words}(\varphi)$ stutter insensitive:

$$\text{there exists } \psi \in \text{LTL}_{\setminus\bigcirc} \text{ such that } \psi \equiv_{LTL} \varphi$$

Stutter bisimulation

Let $TS = (S, Act, \rightarrow, I, AP, L)$ 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 $s_1' \in Post(s_1)$ with $(s_1, s_1') \notin \mathcal{R}$, then there exists a finite path fragment $s_2\, u_1 \ldots u_n\, s_2'$ with $n \geqslant 0$ and $(s_2, u_i) \in \mathcal{R}$ and $(s_1', s_2') \in \mathcal{R}$

3. if $s_2' \in Post(s_2)$ with $(s_2, s_2') \notin \mathcal{R}$, then there exists a finite path fragment $s_1\, v_1 \ldots v_n\, s_1'$ with $n \geqslant 0$ 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

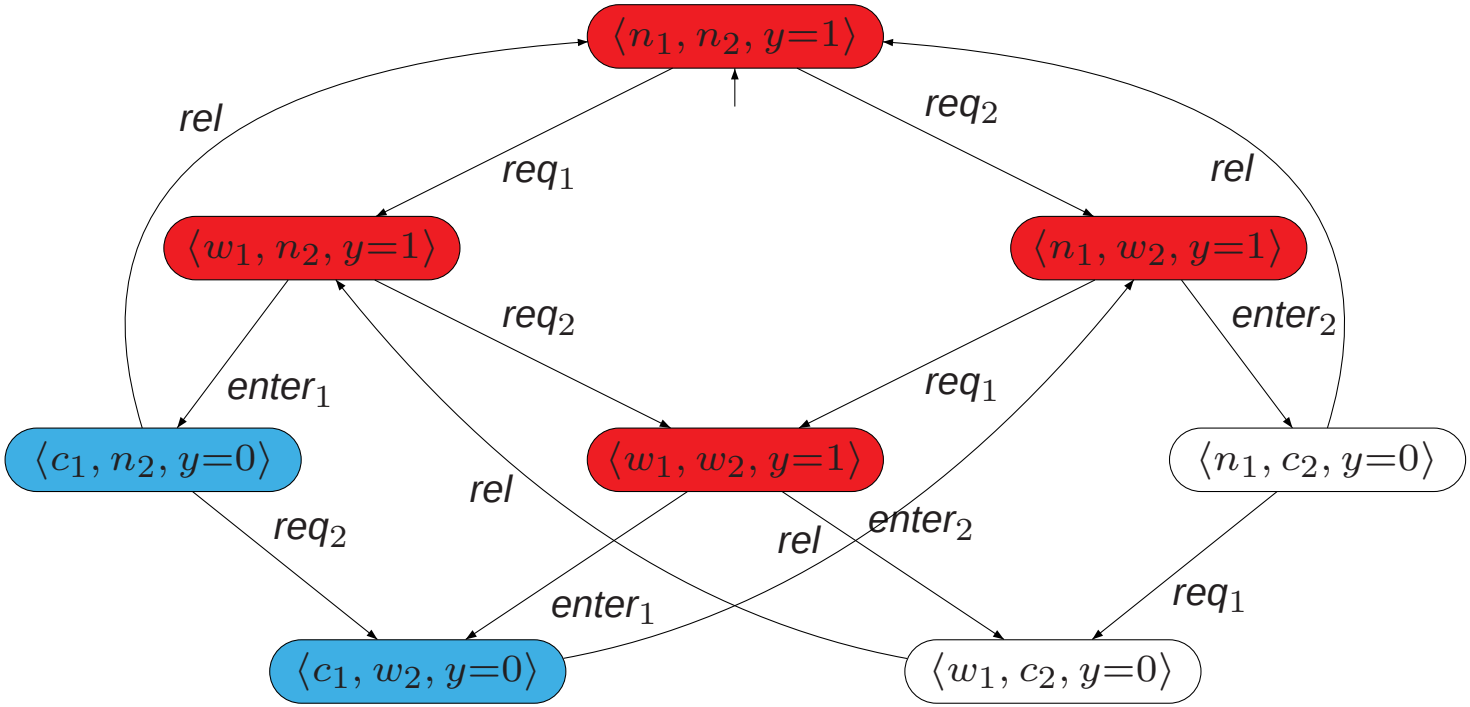
$$\begin{array}{ccc} s_1 & \approx & s_2 \\ \downarrow & & \\ s_1' & & \end{array}$$

(with $s_1 \not\approx s_1'$)

can be completed to

$$\begin{array}{ccc} s_1 & \approx & s_2 \\ & & \downarrow \\ s_1 & \approx & u_1 \\ & & \downarrow \\ s_1 & \approx & u_2 \\ & & \downarrow \\ & & \vdots \\ & & \downarrow \\ s_1 & \approx & u_n \\ \downarrow & & \downarrow \\ s_1' & \approx & s_2' \end{array}$$

Semaphore-based mutual exclusion



stutter-bisimilar states for $AP = \{ crit_1, crit_2 \}$

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 $\mathcal{R}$ on $TS_1 \oplus TS_2$ such that:

$$\forall s_1 \in I_1 . (\exists s_2 \in I_2 . (s_1, s_2) \in \mathcal{R}) \text{ and } \forall s_2 \in I_2 . (\exists s_1 \in I_1 . (s_1, s_2) \in \mathcal{R})$$

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:

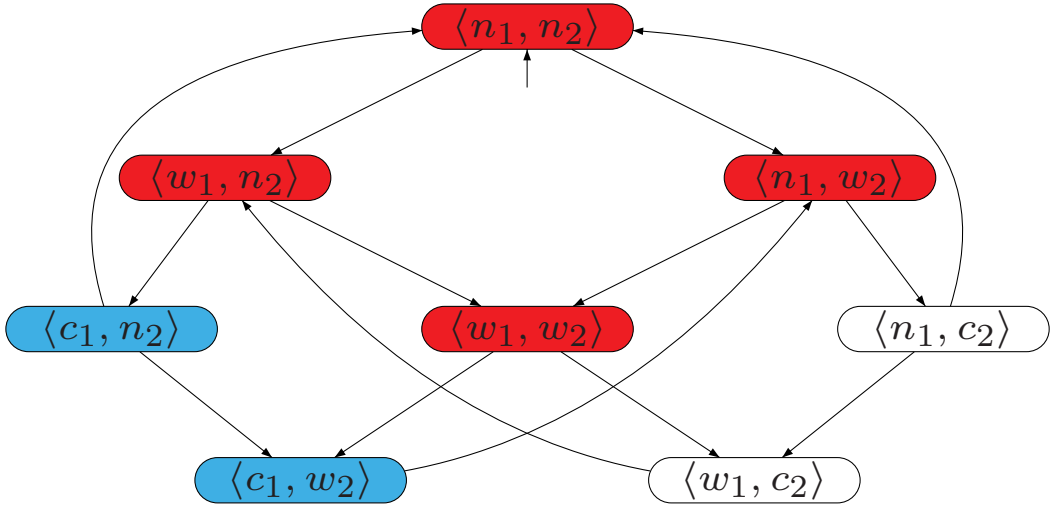
$$TS/\mathcal{R} = (S', \{\tau\}, \rightarrow', I', AP, L')$$

where

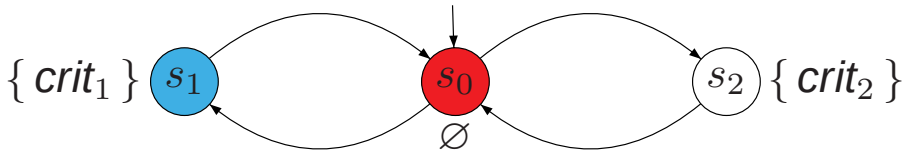
- $S' = S/\mathcal{R} = \{ [s]_\mathcal{R} \mid s \in S \}$ with $[s]_\mathcal{R} = \{ s' \in S \mid (s, s') \in \mathcal{R} \}$
- $I' = \{ [s]_\mathcal{R} \mid s \in I \}$
- $L'([s]_\mathcal{R}) = L(s)$
- $\rightarrow'$ is defined by: $$\frac{s \xrightarrow{\alpha} s' \text{ and } (s, s') \notin \mathcal{R}}{[s]_\mathcal{R}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

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:

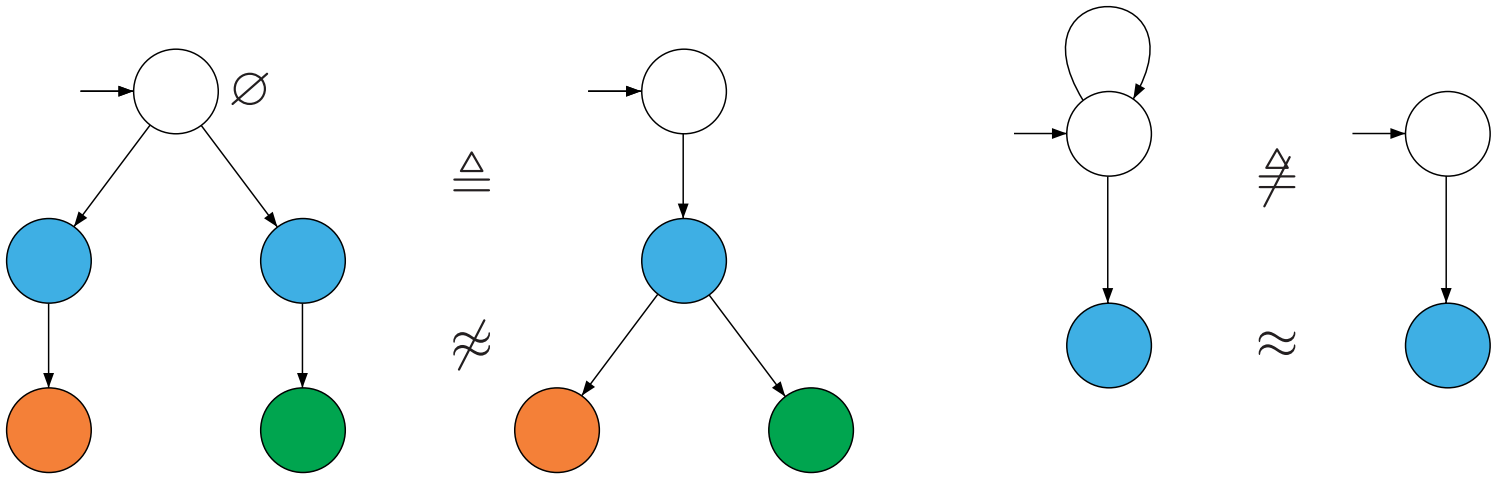


Stutter trace and stutter bisimulation

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



Stutter bisimulation does not preserve LTL$_{\setminus\bigcirc}$

$TS_{left} \approx TS_{right}$ but $TS_{left} \not\models \Diamond a$ and $TS_{right} \models \Diamond a$

reason: presence of infinite stutter paths in TS_{left}

Divergence sensitivity

- *Stutter paths* are paths that only consist of stutter steps
  - no restrictions are imposed on such paths by a stutter bisimulation

  $\Rightarrow$ stutter trace-equivalence ($\triangleq$) and stutter bisimulation ($\approx$) are incomparable

  $\Rightarrow$ $\approx$ and $\mathrm{LTL}_{\setminus\bigcirc}$ 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

  $\Rightarrow$ relate states only if they either both have stutter paths or none of them

- This yields *divergence-sensitive stutter bisimulation* ($\approx^{div}$)

  $\Rightarrow$ $\approx^{div}$ is strictly finer than $\triangleq$ (and $\approx$)

Outlook

formal relation	trace equivalence	bisimulation	simulation
complexity	PSPACE-complete	PTIME	PTIME
logical fragment	LTL	CTL*	$\forall$CTL*
preservation	strong	strong match	weak match

formal relation	stutter trace equivalence	diergence-sensitive stutter bisimulation
complexity	PSPACE-complete	PTIME
logical fragment	LTL$_{\setminus\bigcirc}$	CTL$^*_{\setminus\bigcirc}$
preservation	strong	strong match

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
  $s\, s_1\, s_2 \ldots \in \mathit{Paths}(s)$ 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 *divergent-sensitive stutter-bisimilar*, denoted $s_1 \approx^{div}_{TS} s_2$, if:

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

$\approx^{div}_{TS}$ 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 $\approx^{div}$

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

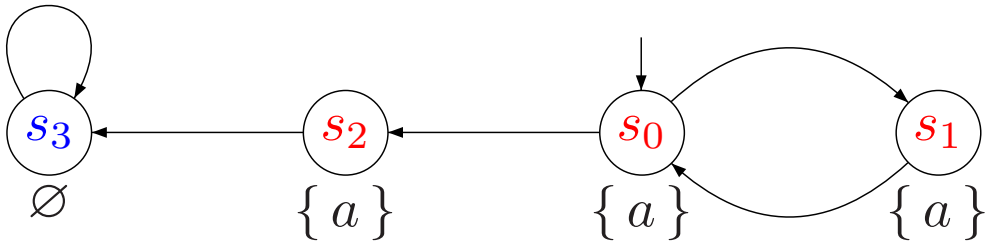
where

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

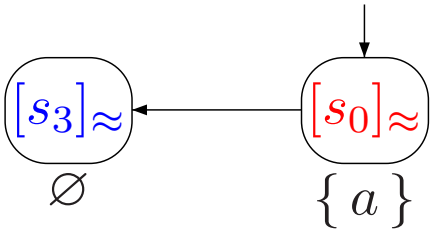
$$\frac{s \xrightarrow{\alpha} s' \ \wedge \ s \not\approx^{div} s'}{[s]_{div} \xrightarrow{\tau}{}'_{div} [s']_{div}} \quad \text{and} \quad \frac{s \text{ is } \approx^{div}\text{-divergent}}{[s]_{div} \xrightarrow{\tau}{}'_{div} [s]_{div}}$$

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

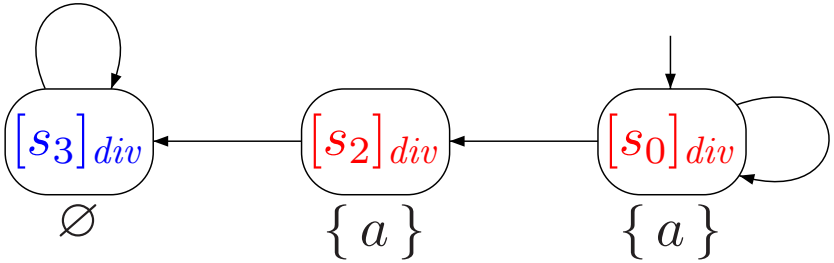
Example



transition system *TS*



transition system $TS/\approx$



transition system $TS/\approx^{div}$

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 $s_{term} \approx_{TS} s_{pd}$ and $s_{term} \not\approx^{div}_{TS} s_{pd}$
- $s_{term} \approx^{div}_{TS} s$ implies
  - $L(s) = L(s_{term})$ and each path of s is finite and divergent

Summary

stutter trace inclusion:

$TS_1 \trianglelefteq TS_2$ iff $\forall \sigma_1 \in \mathit{Traces}(TS_1)\ \exists \sigma_2 \in \mathit{Traces}(TS_2).\ \sigma_1 \stackrel{\triangle}{=} \sigma_2$

stutter trace equivalence:

$TS_1 \stackrel{\triangle}{=} TS_2$ iff $TS_1 \trianglelefteq TS_2$ and $TS_2 \trianglelefteq TS_1$

stutter bisimulation equivalence:

$TS_1 \approx TS_2$ iff there exists a stutter bisimulation for (TS_1, TS_2)

stutter bisimulation equivalence with divergence:

$TS_1 \approx^{div} TS_2$ iff there exists a divergence-sensitive stutter bisimulation for (TS_1, TS_2)

Relationship between equivalences