Verification

Lecture 21

Bernd Finkbeiner

UNIVERSITÄT
DES
SAARLANDES

Plan for today

- Stutter trace equivalence
- Stutter bisimulation

Motivation

- Bisimulation, simulation and trace equivalence are <u>strong</u>
  - each transition $s \to s'$ 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 $s \to s'$ 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: how to treat infinite internal computations?

Stutter 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 successor states
- Paths π_1 and π_2 are stutter equivalent, denoted $\pi_1 \triangleq \pi_2$:
  - if there exists an infinite sequence $A_0 A_1 A_2 \ldots$ with $A_i \subseteq AP$ and
  - natural numbers $n_0, n_1, n_2, \ldots, m_0, m_1, m_2, \ldots \geq 1$ such that:

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

$\pi_1 \triangleq \pi_2$ if their traces only differ in their stutter steps
i.e., if both their traces are of the form $A_0^+ A_1^+ A_2^+ \ldots$ for $A_i \subseteq AP$

Stutter-trace equivalence

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

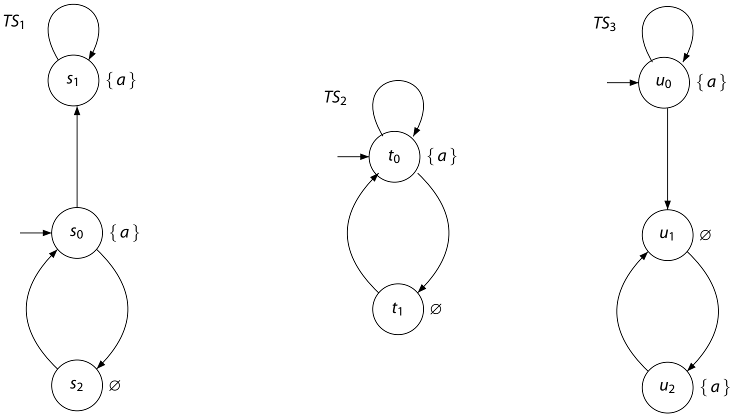
$$TS_1 \cong TS_2 \quad \text{if and only if} \quad TS_1 \sqsubseteq TS_2 \text{ and } TS_2 \sqsubseteq TS_1$$

where $\sqsubseteq$ is defined by:

$$TS_1 \sqsubseteq TS_2 \quad \text{iff} \quad \forall \sigma_1 \in Traces(TS_1) \ (\exists \sigma_2 \in Traces(TS_2). \ \sigma_1 \cong \sigma_2 \ )$$

clearly: $Traces(TS_1) = Traces(TS_2)$ implies $TS_1 \cong TS_2$, but not always the reverse

Example



The X operator

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

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

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

$$\sigma_1 \cong \sigma_2 \quad \text{but} \quad \sigma_1 \models \mathsf{X}\, b \quad \text{and} \quad \sigma_2 \not\models \mathsf{X}\, b.$$

$\Rightarrow$ a logical characterization of $\cong$ can only be obtained by omitting X

in fact, it turns out that this is the only modal operator that is not preserved by $\cong$!

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

For traces σ_1 and σ_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 $\mathrm{LTL}_{\setminus \bigcirc}$ formula φ over AP

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

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

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

(a) $TS_1 \cong TS_2$ implies $TS_1 \equiv_{\mathrm{LTL}_{\setminus\bigcirc}} TS_2$

(b) if $TS_1 \sqsubseteq TS_2$ then for any $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ: $TS_2 \models \varphi$ implies $TS_1 \models \varphi$

Stutter insensitivity

- LT property P is stutter-insensitive if $[\sigma]_{\cong} \subseteq P$, for any $\sigma \in P$
  - P is stutter insensitive if it is closed under stutter equivalence
- For any stutter-insensitive LT property P:

$$TS_1 \cong TS_2 \quad \text{implies} \quad TS_1 \models P \text{ iff } TS_2 \models P$$

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

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

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

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

Stutter bisimulation

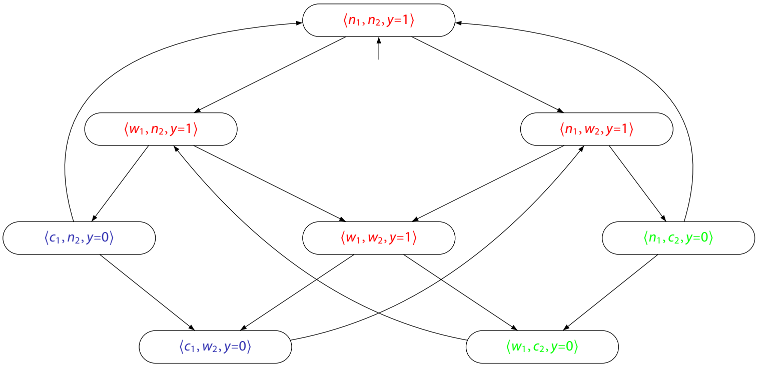
Let $TS = (S, Act, \rightarrow, I, AP, L)$ be a transition system and $\mathcal{R} \subseteq S \times S$

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

Example



For $AP = \{c_1, c_2\}$, $\mathcal{R}$ inducing the following partitioning of the state space is a stutter bisimulation:

$$\{\{\langle n_1, n_2\rangle, \langle n_1, w_2\rangle, \langle w_1, n_2\rangle, \langle w_1, w_2\rangle\}, \{\langle c_1, n_2\rangle, \langle c_1, w_2\rangle\}, \{\langle n_1, c_2\rangle, \langle w_1, c_2\rangle\}\}$$

(Values of y omitted here.) In fact, this is the coarsest stutter bisimulation, i.e., $\mathcal{R}$ equals $\approx_{TS}$

Stutter-bisimilar transition systems

Let $TS_i = (S_i, Act_i, \rightarrow_i, I_i, AP, L_i)$, $i = 1, 2$, be transition systems over AP
A stutter bisimulation for (TS_1, TS_2) is a stutter bisimilation relation on $TS_1 \oplus TS_2$ such that:

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

Notation: $TS_1 \oplus TS_2 = (S_1 \dot{\cup} S_2, Act_1 \cup Act_2, \rightarrow_1 \cup \rightarrow_2, I_1 \cup I_2, AP,$
$L : s \mapsto L_i(s)$ for $s \in S_i)$

TS_1 and TS_2 are stutter-bisimulation equivalent (stutter-bisimilar, for short), denoted $TS_1 \approx TS_2$, if there exists a stutter bisimulation for (TS_1, TS_2)

Stutter bisimulation quotient

For $TS = (S, Act, \rightarrow, I, AP, L)$ and stutter bisimulation $\approx_{TS} \subseteq S \times S$ let

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

where

- $S' = S/\approx_S = \{ [q]_{\approx_S} \mid q \in S \}$ with $[q]_{\approx_S} = \{ q' \in S \mid q \approx_S q' \}$
- $I' = \{ [q]_{\approx_S} \mid q \in I \}$
- $\rightarrow'$ is defined by: $$\frac{s \xrightarrow{\alpha} s' \text{ and } s \not\approx s'}{[s]_{\approx} \xrightarrow{\tau}{}' [s']_{\approx}}$$
- $L'([q]_{\approx_S}) = L(q)$

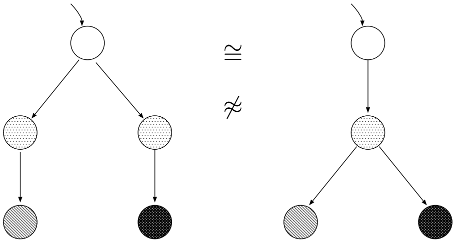
note that (a) no self-loops occur in $TS/\approx_S$ and (b) $TS \approx_S TS/\approx_S$

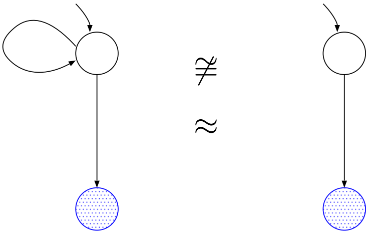
Stutter trace and stutter bisimulation

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 $TS_1 \cong TS_2$!
- 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 $\cong$ requires the existence of a stutter 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 $TS_{right} \models \Diamond a$

stutter-trace inclusion:

$TS_1 \sqsubseteq TS_2$ iff $\forall \sigma_1 \in Traces(TS_1)\ \exists \sigma_2 \in Traces(TS_2).\ \sigma_1 \cong \sigma_2$

stutter-trace equivalence:

$TS_1 \cong TS_2$ iff $TS_1 \sqsubseteq TS_2$ and $TS_2 \sqsubseteq TS_1$

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)

Divergence sensitivity

- Stutter paths are paths that only consist of stutter steps
  - no restrictions are imposed on such paths by stutter bisimulation
  - $\Rightarrow$ stutter trace-equivalence ($\cong$) 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 $\cong$ (and $\approx$)
  - $\Rightarrow$ $\approx^{div}$ and $\mathrm{CTL}^*_{\setminus X}$ equivalence coincide

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 $s\, s_1\, s_2 \ldots \in Paths(s)$ 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 $(s_1, s_2) \in \mathcal{R}$:

$$s_1 \text{ is } \mathcal{R}\text{-divergent implies } s_2 \text{ is } \mathcal{R}\text{-divergent}$$

  - $\mathcal{R}$ is divergence-sensitive if in any $[s]_{\mathcal{R}}$ either all or none of the states are $\mathcal{R}$-divergent

Divergence-sensitive stutter bisimulation

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

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

$\approx_{TS}^{div}$ is an equivalence, the coarsest divergence-sensitive stutter bisimulation for TS

and the union of all divergence-sensitive stutter bisimulations for TS

Quotient transition system under $\approx^{div}$

For $TS = (S, Act, \rightarrow, I, AP, L)$ and divergent-sensitive stutter bisimulation $\approx^{div} \subseteq S \times S$,

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

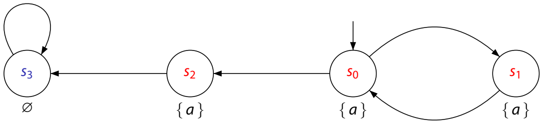
where

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

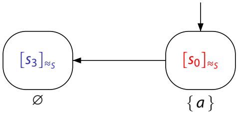
$$\frac{s \xrightarrow{\alpha} s' \;\wedge\; s \not\approx^{div} s'}{[s]_{div} \xrightarrow{\tau}{}'_{div} [s']_{div}} \quad \text{and} \quad \frac{s \text{ is } \approx^{div}\text{-divergent}}{[s]_{div} \xrightarrow{\tau}{}'_{div} [s]_{div}}$$

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

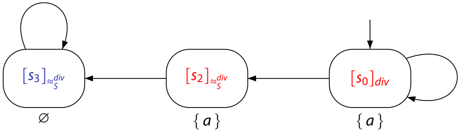
Example



TS



$TS/\approx_S$



$TS/\approx_S^{div}$

$\approx^{div}$ on paths

For infinite path fragments $\pi_i = s_{0,i}\, s_{1,i}\, s_{2,i} \ldots$, $i = 1, 2$, in TS:

$$\pi_1 \approx^{div}_{TS} \pi_2$$

if and only if there exists an infinite sequence of indexes

$$0 = j_0 < j_1 < j_2 < \ldots \quad \text{and} \quad 0 = k_0 < k_1 < k_2 < \ldots$$

with:

$$s_{j,1} \approx^{div}_{TS} s_{k,2} \text{ for all } j_{r-1} \leq j < j_r \text{ and } k_{r-1} \leq k < k_r \text{ with } r = 1, 2, \ldots.$$

Comparing paths by $\approx^{div}$

$$\text{Let } TS = (S, Act, \to, I, AP, L),\ s, t \in S. \text{ Then:}$$

$$s \approx^{div}_{TS} t \quad \text{implies} \quad \forall \pi_1 \in Paths(s).\ \big(\exists \pi_2 \in Paths(t).\ \pi_1 \approx^{div}_{TS} \pi_2\big)$$

Stutter equivalence versus $\approx^{div}$

Let TS_1 and TS_2 be transition systems over AP. Then:

$$\underbrace{TS_1 \approx^{div} TS_2}_{\text{stutter-bisimulation equivalence with divergence}} \quad \text{implies} \quad \underbrace{TS_1 \cong TS_2}_{\text{stutter-trace equivalence}}$$

whereas the reverse implication does not hold in general

$\text{CTL}^*_{\setminus X}$ equivalence and $\approx^{div}$

For finite transition systems *TS* without terminal states, and s_1, s_2 in *TS*:

$$s_1 \approx^{div}_{TS} s_2 \quad \text{iff} \quad s_1 \equiv_{\text{CTL}^*_{\setminus X}} s_2 \quad \text{iff} \quad s_1 \equiv_{\text{CTL}_{\setminus X}} s_2$$

divergent-sensitive stutter bisimulation coincides with $\text{CTL}_{\setminus X}$ and $\text{CTL}^*_{\setminus X}$ equivalence

Comparative semantics

$$
\begin{tikzcd}
\text{CTL}^* \text{ equivalence} \arrow[r] \arrow[d, leftrightarrow] & \text{LTL equivalence} \arrow[d, leftrightarrow] & \\
\begin{array}{c}\text{bisimulation equivalence}\\ TS_1 \sim TS_2\end{array} \arrow[r] \arrow[d] & \begin{array}{c}\text{trace equivalence}\\ \mathit{Traces}(T_1) = \mathit{Traces}(TS_2)\end{array} \arrow[r] \arrow[d] & \begin{array}{c}\text{trace inclusion}\\ \mathit{Traces}(T_1) \subseteq \mathit{Traces}(TS_2)\end{array} \arrow[d] \\
\begin{array}{c}\text{divergence sensitive}\\ \text{stutter bisimulation equivalence}\\ TS_1 \approx^{\div} TS_2\end{array} \arrow[r] \arrow[d, leftrightarrow] & \begin{array}{c}\text{stutter trace-equivalence}\\ TS_1 \cong TS_2\end{array} \arrow[r] \arrow[d, leftrightarrow] & \begin{array}{c}\text{stutter trace inclusion}\\ TS_1 \sqsubseteq TS_2\end{array} \\
\text{CTL}^*_{\setminus X} \text{ equivalence} \arrow[r] & \text{LTL}_{\setminus \bigcirc} \text{ equivalence} &
\end{tikzcd}
$$