Stuttering Congruence for $\chi$

Bas Luttik, Nikola Trčka

n.trcka@tue.nl

Eindhoven University of Technology
Faculty of Mathematics and Computer Science,
Formal Methods Group
What is $\chi$?

- language for modeling, simulation and control of manufacturing systems (machines, production cells, factories ...)
- developed by Systems Engineering Group, Faculty of Mechanical Engineering, TU/e
- process algebra like
- data types and continuous-time support
- discrete-event and hybrid models
- many industrial cases
How to verify $\chi$ models?

- Model checking
- Two ways:
  1. build a model checker for $\chi$
  2. translate a model to an input language of a popular model checker
How to verify $\chi$ models?

• Model checking

• Two ways:

  1. build a model checker for $\chi$
     ○ hard to beat the existing ones
     ○ must chose CTL/LTL
     ○ etc.

  2. translate a model to an input language of a popular model checker
     ○ case study: turntable machine
     ○ model translated to $\mu$CRL, PROMELA and UPPAAL timed automata
     ○ successful verification in CADP, SPIN and UPPAAL
Translating $\chi$ to another language

Problem:
- hard to prove the translation correct (not common semantics, no semantics at all, etc.)
Proposed solution

1. transform a $\chi$ model to a syntactically simpler form
2. map it to a target language easily
3. prove the reduction correct

Extra gain:
- reusability
Correctness criterion

Two important things:

1. deadlock preservation
2. preservation of temporal logic formulas
   - want both $\text{LTL}^{-X}$ and $\text{CTL}^{-X}$
   - state based
New observational equivalence

- hard to work directly with the criterion
- easier to work with a bisimulation-like equivalence
- strong bisimulation we have for $\chi$ is too strong:
- good candidate: Stuttering Equivalence [Browne, Clarke, Grümberg]
  - equivalence on Kripke structures
  - characterization of $\text{CTL}^*_\chi$

Goal: turn stuttering equivalence into a suitable equivalence/congruence for $\chi$ processes
Introduction to $\chi$ - Syntax

- untimed, discrete-event subset

\[
\begin{align*}
a & ::= \varepsilon | \delta | \text{skip} | x ::= e | m!e | m?x \\
p & ::= a \\
    & | b :\rightarrow p \\
    & | p ; p \\
    & | p \parallel p \\
    & | p^* \\
    & | p | p \\
    & | [s | p] \\
    & | \partial(p) \\
s = \{x_1 \mapsto c_1, \ldots, x_n \mapsto c_n\} - \text{state}
\end{align*}
\]
Introduction to $\chi$ - Semantics

- operational semantics in terms of configurations
- configuration = process + context ($c = \langle p, \sigma \rangle$)

Semantical predicates

- immediate termination: $c \downarrow$
- action step: $c \xrightarrow{a} c'$

Some SOS rules:

1. $\langle \varepsilon, \sigma \rangle \downarrow \langle \text{skip}, \sigma \rangle \xrightarrow{\tau} \langle \varepsilon, \sigma \rangle$
2. $\sigma(e) = d$
   $\langle x := e, \sigma \rangle \xrightarrow{aa(x,d)} \langle \varepsilon, \gamma(\{x \mapsto d\}, \sigma) \rangle$
3. $\sigma(e) = d$
   $\langle m!e, \sigma \rangle \xrightarrow{sa(m,d)} \langle \varepsilon, \sigma \rangle$
4. $\langle p, \sigma \rangle \xrightarrow{sa(m,d)} \langle p', \sigma \rangle$
   $\langle q, \sigma \rangle \xrightarrow{ra(m,d)} \langle q', \sigma' \rangle$
5. $\langle p \parallel q, \sigma \rangle \xrightarrow{ca(m,d)} \langle p' \parallel q', \sigma' \rangle$
   $\langle q \parallel p, \sigma \rangle \xrightarrow{ca(m,d)} \langle q' \parallel p', \sigma' \rangle$
Semantical model - example

\[
\langle \text{skip} \; ; \; x := x + y, \; \{ x \mapsto 2, \; y \mapsto 5 \} \rangle
\]

\[
\tau
\]

\[
\langle x := x + y, \; \{ x \mapsto 2, \; y \mapsto 5 \} \rangle
\]

\[
\text{aa}(x,7)
\]

\[
\langle \varepsilon, \; \{ x \mapsto 7, \; y \mapsto 5 \} \rangle
\]
Stuttering Bisimilarity $\sim_{st}$

Stuttering bisimulation

- symmetric binary relation on configurations
- $cRd$ iff
  1. $\text{state}(c) = \text{state}(d)$
  2. if $c \rightarrow c'$ then
     $\exists d_0, \ldots, d_n : d = d_0 \rightarrow \ldots \rightarrow d_n, c'Rd_n \text{ and } \forall i < n : cRd_i$.

Stuttering bisimilarity: $c \sim_{st} d$ iff $\exists R : cRd$. 

\[
\begin{array}{c}
\text{c} \quad \text{d} \\
\downarrow \\
\text{d}_1 \\
\downarrow \\
\text{c'} \quad \text{d}_2
\end{array}
\]
Stuttering Bisimilarity - Example

\[ \langle x := x + y, \{ x \mapsto 2, y \mapsto 5 \} \rangle \quad \text{---} \quad \langle \text{skip}; x := x + y, \{ x \mapsto 2, y \mapsto 5 \} \rangle \]

\[ \langle x := x + y, \{ x \mapsto 2, y \mapsto 5 \} \rangle \]

\[ \langle \varepsilon, \{ x \mapsto 7, y \mapsto 5 \} \rangle \quad \text{---} \quad \langle \varepsilon, \{ x \mapsto 7, y \mapsto 5 \} \rangle \]
Divergence

Stuttering bisimilarity

• equates deadlock and divergence

\[ c \cdot \cdot \cdot d \]

• does not always preserve $\text{CTL}^*_\neg x$

\[ c \cdot \cdot \cdot d \]

\[ c' \cdot \cdot \cdot d' \]

Divergence condition (infinite paths simulated by infinite paths):

3. if $c = c_0 \rightarrow c_1 \rightarrow \ldots$ and $\forall i \geq 0 : c_iRd$, then $\exists d', j > 0 : d \rightarrow d'$ and $c_jRd'$. 
Successful Termination

Stuttering bisimilarity

- equates successful termination and deadlock

\[ c \rightarrow \rightarrow \rightarrow \rightarrow d \]

Termination condition:

4. if \( c \downarrow \), then

\[ \exists d_0, \ldots, d_n : d = d_0 \rightarrow \ldots \rightarrow d_n, d_n \downarrow \text{ and } \forall i \leq n : cRd_i. \]
Main properties of (the modified) $\sim_{st}$

- $\sim_{st}$ is an equivalence relation (transitivity proof not trivial)
- if $c \sim_{st} d$, then
  1. $\text{deadlock}(c)$ iff $\text{deadlock}(d)$,
  2. $c \models \varphi$ iff $d \models \varphi$ for all $\varphi \in \text{CTL}^*_X$. 
Extending $\sim_{st}$ to $\chi$ process terms

Two processes are equivalent iff they are equivalent in every context:

$$p \sim_{st} q \iff \forall \sigma : \langle p, \sigma \rangle \sim_{st} \langle q, \sigma \rangle.$$

Example:

$$[[x \mapsto 0 \mid x := 7]] \sim_{st} \text{skip}$$

Remark: Equivalence must be stateless
\( \sim_{st} \) is not a congruence!

Two reasons:

1. equates processes that can influence a choice with those that cannot:

   \[ \delta \sim_{st} \text{skip} ; \delta \quad \text{but} \quad \text{skip} \parallel \delta \not\sim_{st} \text{skip} \parallel \text{skip} ; \delta \]

2. completely ignores action labels:

   \[ a!0 \sim_{st} \text{skip} \quad \text{but} \]

   \[ a!0 \parallel a?x \not\sim_{st} \text{skip} \parallel a?x \quad \text{and} \quad \partial(a!0) \not\sim_{st} \partial(\text{skip}) \]

Solution:

1. add a root condition
2. do not ignore send and receive action labels
Interaction Sensitive Stuttering Bisimilarity $\sim_{isst}$

- a send/receive action must be simulated by the same action

- $c \xrightarrow{(a)} c'$ means \[
\begin{cases}
    c \xrightarrow{c'}, & a \text{ is not send or receive} \\
    c \xrightarrow{a} c', & a \text{ is a send or receive}
\end{cases}
\]

Then,

1. if $c \downarrow$, then $\exists d_0, \ldots, d_n$:
   
   $d = d_0 \leftarrow \cdots \leftarrow d_n$, $d_n \downarrow$ and $\forall i \leq n : cRd_i$,

2. if $c \xrightarrow{a} c'$, then $\exists d_0, \ldots, d_n$ : such that
   
   $d = d_0 \leftarrow \cdots \leftarrow d_{n-1} \xrightarrow{(a)} d_n$, $\forall i \leq n - 1 : cRd_i$ and $c'Rd_n$,

3. if $c = c_0 \leftarrow c_1 \leftarrow c_2 \leftarrow \cdots$ and $\forall i \geq 0 : c_i Rd$, then
   
   $\exists d, j > 0 : d \leftarrow d'$ and $c_j Rd'$.
Rooted $\sim_{isst} = \sim_{riss}$

- first steps of equivalent processes must be the same

$c \sim_{riss} d$ iff

1. $c \downarrow$ iff $d \downarrow$

2. if $c \xrightarrow{a} c'$, then
   \[ \exists d' : d \xrightarrow{(a)} d' \text{ and } c' \sim_{isst} d', \]

3. if $d \xrightarrow{a} d'$, then
   \[ \exists c' : c \xrightarrow{(a)} c' \text{ and } c' \sim_{isst} d'. \]
Stuttering Congruence $\approx_{st}$

$p \approx_{st} q$ iff $\forall \sigma : \langle p, \sigma \rangle \approx_{riss} \langle q, \sigma \rangle$

Main theorem:

1. if $p \approx_{st} q$ then $p \approx_{st} q$
2. $\approx_{st}$ is an equivalence relation
3. $\approx_{st}$ is a congruence
4. many reductions proved correct modulo $\approx_{st}$
Example - Translating $\chi$ to PROMELA

Have to eliminate:

- **nested parallelism**

\[
(p \parallel q) \; r
\]

\[
[w \mapsto 0 \mid p ; w := w + 1 \parallel q ; w := w + 1 \parallel w = 2 : \rightarrow r]
\]

- **nested scopes**

\[
[x \mapsto 0 \mid p]^*
\]

\[
[x \mapsto 0 \mid (p ; x := 0)^*]
\]
Future work

• find more reductions that are correct modulo stuttering congruence
• extend the stuttering congruence to cover the continuous time support of $\chi$
• investigate further applications in domains outside $\chi$