

Stuttering Congruence for χ

Bas Luttik, Nikola Trčka

`n.trcka@tue.nl`

Eindhoven University of Technology
Faculty of Mathematics and Computer Science,
Formal Methods Group

What is χ ?

- 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

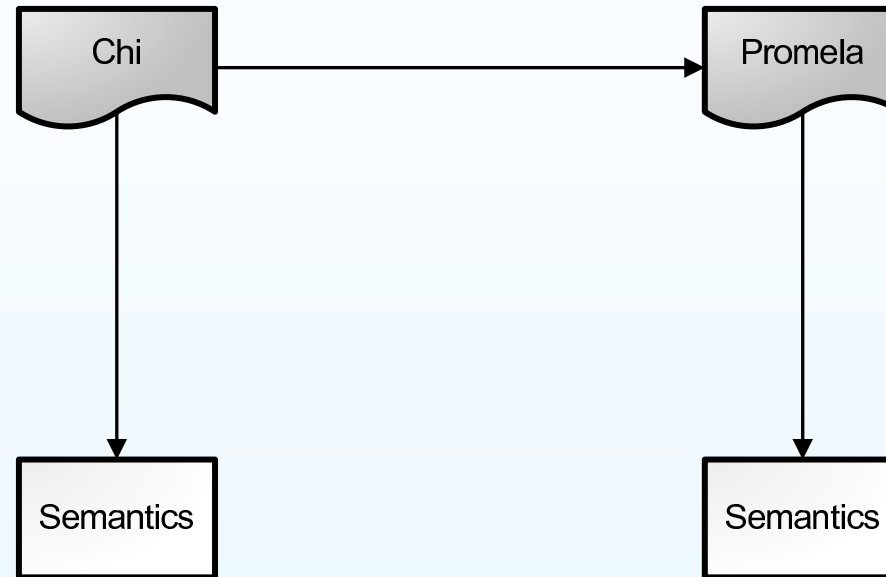
How to verify χ models?

- Model checking
- Two ways:
 1. build a model checker for χ
 2. translate a model to an input language of a popular model checker

How to verify χ models?

- Model checking
- Two ways:
 1. build a model checker for χ
 - 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 μ CRL, PROMELA and UPPAAL timed automata
 - successful verification in CADP, SPIN and UPPAAL

Translating χ to another language

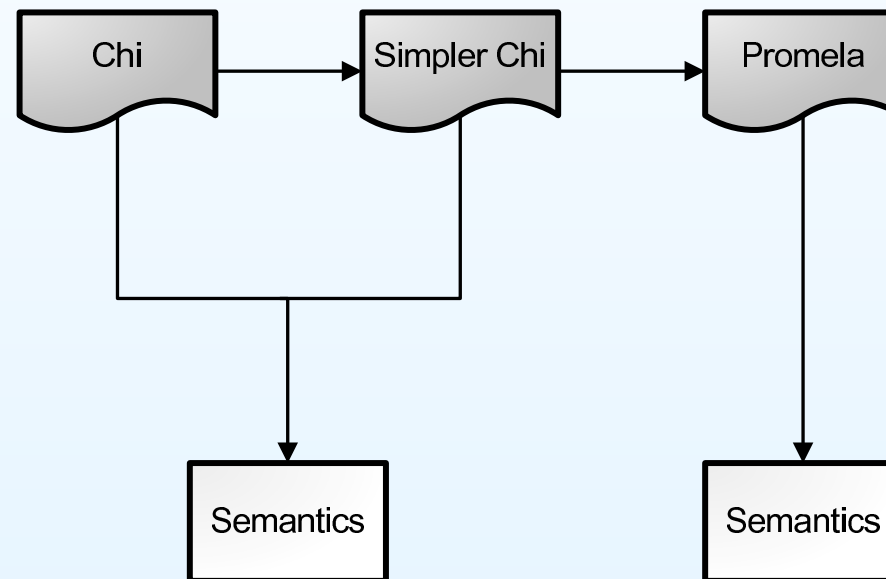


Problem:

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

Proposed solution

1. transform a χ 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 LTL_X and 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 χ is too strong:
- good candidate: Stuttering Equivalence [Browne, Clarke, Grumberg]
 - equivalence on Kripke structures
 - characterization of $\text{CTL}_{-\chi}^*$

Goal: turn stuttering equivalence into a suitable equivalence/congruence for χ processes

Introduction to χ - Syntax

- untimed, discrete-event subset

$a ::= \varepsilon \mid \delta \mid skip \mid x := e \mid m!e \mid m?x$

$p ::=$
 a
 $\mid b \rightarrow p$
 $\mid p ; p$
 $\mid p \square p$
 $\mid p^*$
 $\mid p \parallel p$
 $\mid \llbracket s \mid p \rrbracket$
 $\mid \partial(p)$

$s = \{x_1 \mapsto c_1, \dots, x_n \mapsto c_n\}$ - state

Introduction to χ - 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:

$$\frac{}{\langle \varepsilon, \sigma \rangle \downarrow} \quad 1 \quad \frac{}{\langle \text{skip}, \sigma \rangle \xrightarrow{\tau} \langle \varepsilon, \sigma \rangle} \quad 2 \quad \frac{\sigma(e) = d}{\langle x := e, \sigma \rangle \xrightarrow{aa(x,d)} \langle \varepsilon, \gamma(\{x \mapsto d\}, \sigma) \rangle} \quad 3$$

$$\frac{\sigma(e) = d}{\langle m!e, \sigma \rangle \xrightarrow{sa(m,d)} \langle \varepsilon, \sigma \rangle} \quad 4 \quad \frac{\langle p, \sigma \rangle \xrightarrow{sa(m,d)} \langle p', \sigma \rangle, \langle q, \sigma \rangle \xrightarrow{ra(m,d)} \langle q', \sigma' \rangle}{\langle p \parallel q, \sigma \rangle \xrightarrow{ca(m,d)} \langle p' \parallel q', \sigma' \rangle, \langle q \parallel p, \sigma \rangle \xrightarrow{ca(m,d)} \langle q' \parallel p', \sigma' \rangle} \quad 5$$

Semantical model - example

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

τ

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

$aa(x,7)$

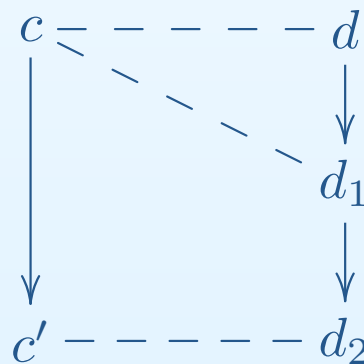
$\langle \varepsilon, \{x \mapsto 7, y \mapsto 5\} \rangle$

Stuttering Bisimilarity \sim_{st}

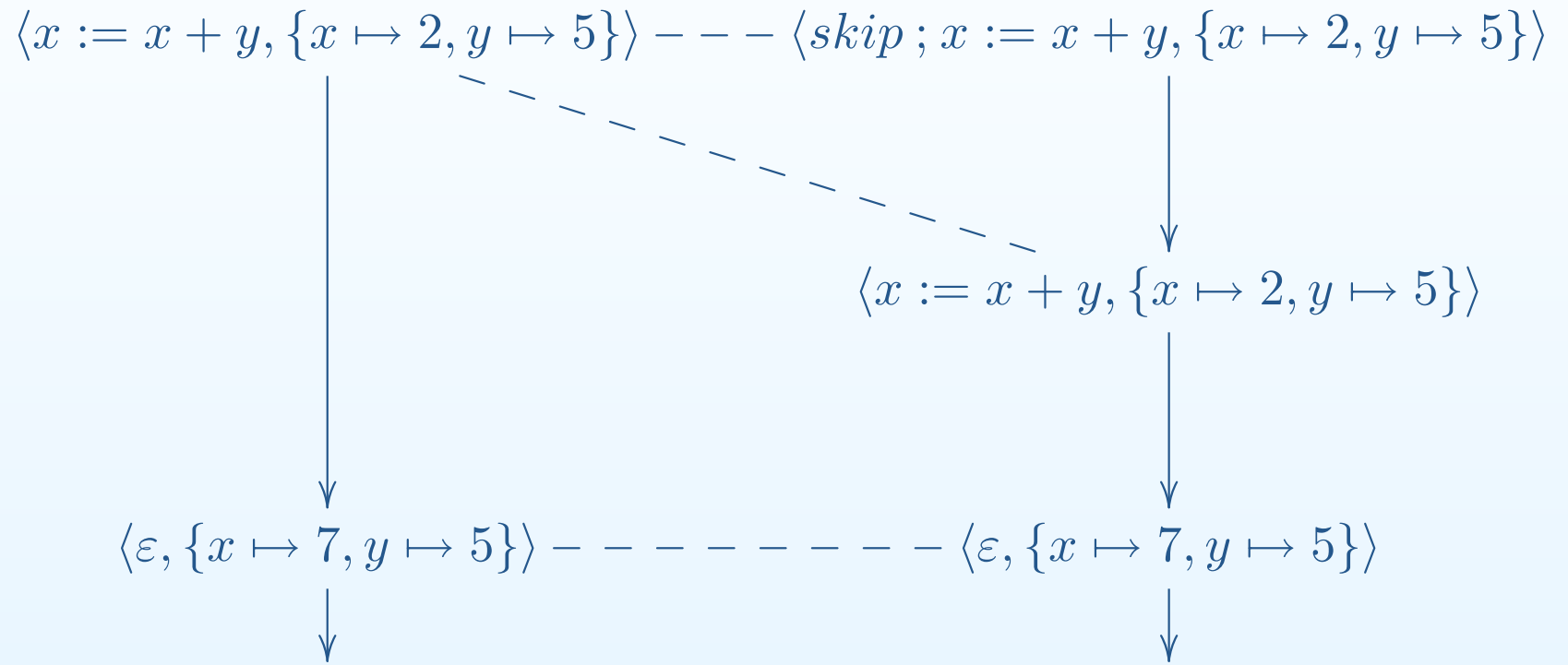
Stuttering bisimulation

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

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



Stuttering Bisimilarity - Example



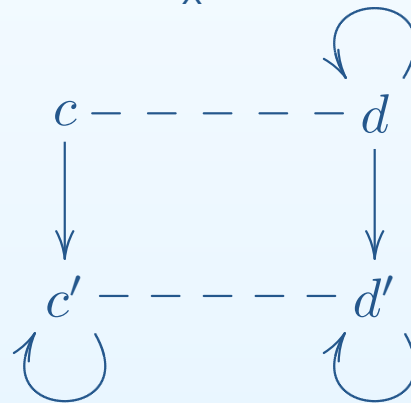
Divergence

Stuttering bisimilarity

- equates deadlock and divergence



- does not always preserve CTL_{-X}^*



Divergence condition (infinite paths simulated by infinite paths):

3. if $c = c_0 \rightarrow c_1 \rightarrow \dots$ and $\forall i \geq 0 : c_i R d$, then $\exists d', j > 0 : d \rightarrow d'$ and $c_j R d'$.

Successful Termination

Stuttering bisimilarity

- equates successful termination and deadlock

$$c \text{ --- } d$$

\downarrow

Termination condition:

4. if $c \downarrow$, then

$\exists d_0, \dots, d_n : d = d_0 \rightarrow \dots \rightarrow d_n, d_n \downarrow$ and $\forall i \leq n : c R d_i$.

Main properties of (the modified) \sim_{st}

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

Extending \sim_{st} to χ process terms

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

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

Example:

$$\llbracket x \mapsto 0 \mid x := 7 \rrbracket \sim_{st} \text{skip}$$

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:

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

2. completely ignores action labels:

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

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

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 $\begin{cases} c \hookrightarrow c', & a \text{ is not send or receive} \\ c \xrightarrow{a} c', & a \text{ is a send or receive} \end{cases}$

Then,

1. if $c \downarrow$, then $\exists d_0, \dots, d_n :$
 $d = d_0 \hookrightarrow \dots \hookrightarrow d_n, d_n \downarrow$ and $\forall i \leq n : c R d_i,$
2. if $c \xrightarrow{a} c'$, then $\exists d_0, \dots, d_n :$ such that
 $d = d_0 \hookrightarrow \dots \hookrightarrow d_{n-1} \xrightarrow{(a)} d_n, \forall i \leq n - 1 : c R d_i$ and $c' R d_n,$
3. if $c = c_0 \hookrightarrow c_1 \hookrightarrow c_2 \hookrightarrow \dots$ and $\forall i \geq 0 : c_i R d,$ then
 $\exists d', j > 0 : d \hookrightarrow d'$ and $c_j R d'.$

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

$\exists d' : d \xrightarrow{(a)} d'$ and $c' \sim_{isst} d'$,

3. if $d \xrightarrow{a} d'$, then

$\exists c' : c \xrightarrow{(a)} c'$ and $c' \sim_{isst} d'$.

Stuttering Congruence \cong_{st}

$$p \cong_{st} q \text{ iff } \forall \sigma : \langle p, \sigma \rangle \sim_{riss} \langle q, \sigma \rangle$$

Main theorem:

1. if $p \cong_{st} q$ then $p \sim_{st} q$
2. \cong_{st} is an equivalence relation
3. \cong_{st} is a congruence
4. many reductions proved correct modulo \cong_{st}

Example - Translating χ to PROMELA

Have to eliminate:

- nested parallelism

$$\begin{array}{c} (p \parallel q) ; r \\ \downarrow \\ \llbracket w \mapsto 0 \mid p ; w := w + 1 \parallel q ; w := w + 1 \parallel w = 2 : \rightarrow r \rrbracket \end{array}$$

- nested scopes

$$\begin{array}{c} \llbracket x \mapsto 0 \mid p \rrbracket^* \\ \downarrow \\ \llbracket x \mapsto 0 \mid (p ; x := 0)^* \rrbracket \end{array}$$

Future work

- find more reductions that are correct modulo stuttering congruence
- extend the stuttering congruence to cover the continuous time support of χ
- investigate further applications in domains outside χ