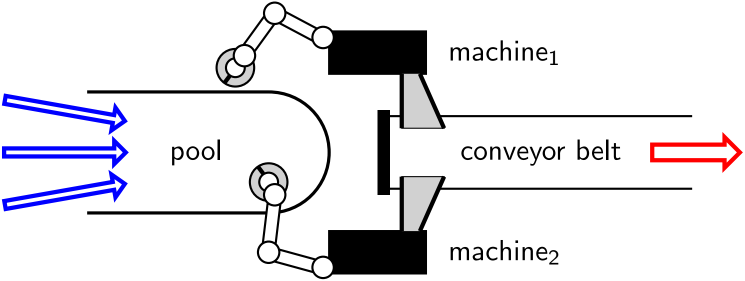
Stochastic HYPE
a stochastic hybrid process algebra

Vashti Galpin
Laboratory for Foundations of Computer Science
University of Edinburgh

Joint work with Jane Hillston (University of Edinburgh)
and Luca Bortolussi (University of Trieste)

23 October 2012

Assembly system



Outline

Introduction

- behaviours to be included
  - discrete behaviour: instantaneous events
  - continuous behaviour: ordinary differentials equations (ODEs)
  - stochastic behaviour: exponentially-distributed durations

Introduction

- behaviours to be included
  - discrete behaviour: instantaneous events
  - continuous behaviour: ordinary differentials equations (ODEs)
  - stochastic behaviour: exponentially-distributed durations
- process algebra approach
  - formal languages for expressing concurrency
  - compositional semantics
  - notions of equivalence

Introduction

- behaviours to be included
  - discrete behaviour: instantaneous events
  - continuous behaviour: ordinary differentials equations (ODEs)
  - stochastic behaviour: exponentially-distributed durations
- process algebra approach
  - formal languages for expressing concurrency
  - compositional semantics
  - notions of equivalence
- HYPE
  - only discrete and continuous behaviour
  - operational semantics define labelled transition system
  - mapping from labelled transition system to hybrid automaton

Motivation

- why?

Motivation

- why?
- why not . . .

Motivation

- why?
- why not . . .
- use hybrid PEPA?

Motivation

- why?
- why not ...
- use hybrid PEPA?
  - no instantaneous transitions

Motivation

- why?
- why not ...
- use hybrid PEPA?
  - no instantaneous transitions
- use stochastic hybrid automata?

Motivation

- why?
- why not . . .
- use hybrid PEPA?
  - no instantaneous transitions
- use stochastic hybrid automata?
  - not a compositional language (in the process algebra sense)
  - limited compositionality with respect to continuous variables

Motivation

- why?
- why not ...
- use hybrid PEPA?
  - no instantaneous transitions
- use stochastic hybrid automata?
  - not a compositional language (in the process algebra sense)
  - limited compositionality with respect to continuous variables
- add stochastic behaviour to existing hybrid process algebras?

Motivation

- why?
- why not ...
- use hybrid PEPA?
  - no instantaneous transitions
- use stochastic hybrid automata?
  - not a compositional language (in the process algebra sense)
  - limited compositionality with respect to continuous variables
- add stochastic behaviour to existing hybrid process algebras?
  - monolithic ODEs in the syntax
  - limited compositionality with respect to continuous variables

Motivation

- why?
- why not . . .
- use hybrid PEPA?
  - no instantaneous transitions
- use stochastic hybrid automata?
  - not a compositional language (in the process algebra sense)
  - limited compositionality with respect to continuous variables
- add stochastic behaviour to existing hybrid process algebras?
  - monolithic ODEs in the syntax
  - limited compositionality with respect to continuous variables

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables
- monolithic ODEs in existing hybrid process algebras

$$A \stackrel{def}{=} \ \ldots \quad [\frac{dV}{dt} = f(\mathcal{V})] \quad \ldots$$

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables
- monolithic ODEs in existing hybrid process algebras

$$A \stackrel{\mathit{def}}{=} \quad \ldots \quad [\frac{dV}{dt} = f(\mathcal{V})] \quad \ldots$$

- flows in stochastic HYPE ($W_j \subseteq \mathcal{V}$)

$$\begin{array}{lllll} A_1 & \stackrel{\mathit{def}}{=} & \ldots & (\iota_1, r_1, I_1(\mathcal{W}_1)) & \ldots \\ \vdots & \vdots & & \vdots & \\ A_n & \stackrel{\mathit{def}}{=} & \ldots & (\iota_n, r_n, I_n(\mathcal{W}_n)) & \ldots \end{array}$$

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables
- monolithic ODEs in existing hybrid process algebras

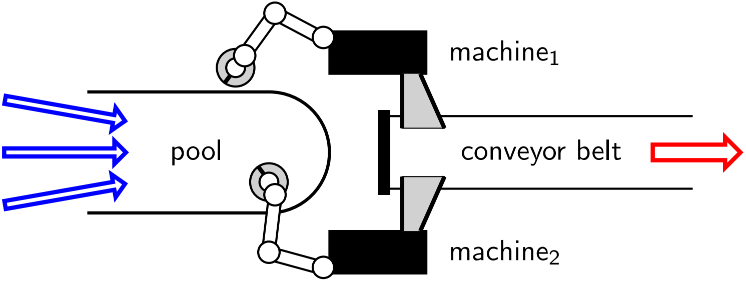
$$A \stackrel{def}{=} \quad \dots \quad [\tfrac{dV}{dt} = f(\mathcal{V})] \quad \dots$$

- flows in stochastic HYPE ($W_j \subseteq \mathcal{V}$)

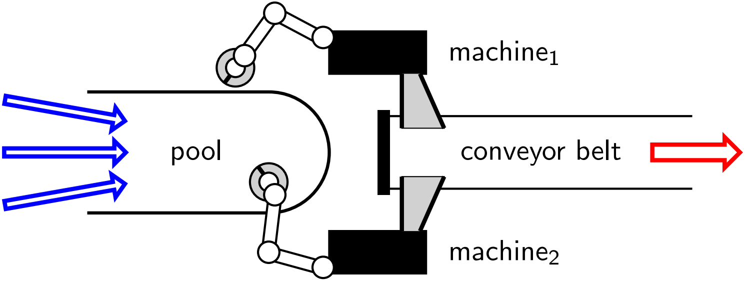
$$\begin{array}{ccccc} A_1 & \stackrel{def}{=} & \dots & (\iota_1, r_1, I_1(\mathcal{W}_1)) & \dots \\ \vdots & \vdots & & \vdots & \\ A_n & \stackrel{def}{=} & \dots & (\iota_n, r_n, I_n(\mathcal{W}_n)) & \dots \end{array}$$

and $$\frac{dV}{dt} = \sum\{r_j.I_j(\mathcal{W}_j) \mid iv(\iota_j) = V, \dots\}$$

Assembly system



Assembly system



- continuous variables
  - individual items in pool: P
  - assembled items at start of conveyor belt: B
  - power consumption of machine$_i$: W_i
  - timers: T_i, T

Stochastic HYPE model

Stochastic HYPE model

uncontrolled system

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big)$$

Stochastic HYPE model

uncontrolled system

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie}$$

Stochastic HYPE model

uncontrolled system controllers/sequencers

$$\left(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\right) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\left(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\right)$$

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

Stochastic HYPE model

uncontrolled system | controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{\mathit{def}}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

subcomponents are parameterised by variables

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \, . \, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \, . \, C(\mathcal{V})$$

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j . C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha . C(\mathcal{V})$$

events have event conditions: guards/durations and resets

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\left(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\right) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\left(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\right)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

events have event conditions: guards/durations and resets

$ec(\underline{\mathrm{a}}_j) = (g(\mathcal{V}), \mathcal{V}' = g'(\mathcal{V}))$ with $g : \mathbb{R}^{|\mathcal{V}|} \rightarrow \{true, false\}$ discrete

Stochastic HYPE model

uncontrolled system controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

events have event conditions: guards/durations and resets

$$ec(\underline{\mathrm{a}}_j) = (g(\mathcal{V}), \mathcal{V}' = g'(\mathcal{V})) \text{ with } g : \mathbb{R}^{|\mathcal{V}|} \to \{true, false\} \qquad \text{discrete}$$

$$ec(\overline{\mathrm{a}}_j) = (f(\mathcal{V}), \mathcal{V}' = f'(\mathcal{V})) \text{ with } f : \mathbb{R}^{|\mathcal{V}|} \to [0, \infty) \qquad \text{stochastic}$$

Stochastic HYPE model

uncontrolled system

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big)$$

$$\underset{*}{\bowtie}$$

controllers/sequencers

$$\underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{\mathit{def}}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

influences are defined by a triple

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{\mathit{def}}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

influences are defined by a triple

$$\alpha_j = (\iota_j, r_j, I_j(\mathcal{V}))$$

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

influences are defined by a triple

$$\alpha_j = (\iota_j, r_j, I_j(\mathcal{V}))$$

influence names are mapped to variables

$$iv(\iota_j) \in \mathcal{V}$$

Uncontrolled system

$$
\begin{aligned}
Machine_i(W_i) \stackrel{def}{=}\ & \underline{\mathrm{init}} : (w_i, wa_i,\ linear(W_i)).Machine_i(W_i) + \\
& \overline{\mathrm{prep}}_i : (w_i, 0, const).Machine_i(W_i) + \\
& \underline{\mathrm{take}}_i : (w_i, wt_i,\ linear(W_i)).Machine_i(W_i) + \\
& \underline{\mathrm{assem}}_i : (w_i, wa_i,\ linear(W_i)).Machine_i(W_i)
\end{aligned}
$$

Uncontrolled system

$$
\begin{aligned}
\mathit{Machine}_i(W_i) \stackrel{\mathit{def}}{=}\ & \underline{\mathrm{init}} : (w_i, wa_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
& \overline{\mathrm{prep}}_i : (w_i, 0, \mathit{const}).\mathit{Machine}_i(W_i) + \\
& \underline{\mathrm{take}}_i : (w_i, wt_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
& \underline{\mathrm{assem}}_i : (w_i, wa_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i)
\end{aligned}
$$

Uncontrolled system

$$\begin{aligned} \mathit{Machine}_i(W_i) \stackrel{\mathit{def}}{=}\ & \underline{\mathrm{init}} : (w_i, wa_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\ & \overline{\mathrm{prep}}_i : (w_i, 0, \mathit{const}).\mathit{Machine}_i(W_i) + \\ & \underline{\mathrm{take}}_i : (w_i, wt_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\ & \underline{\mathrm{assem}}_i : (w_i, wa_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) \end{aligned}$$

Uncontrolled system

$$
\begin{aligned}
\mathit{Machine}_i(W_i) \;&\stackrel{\mathit{def}}{=}\; \underline{\mathrm{init}} : (w_i, wa_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
&\quad \overline{\mathrm{prep}}_i : (w_i, 0, \mathit{const}).\mathit{Machine}_i(W_i) + \\
&\quad \underline{\mathrm{take}}_i : (w_i, wt_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
&\quad \underline{\mathrm{assem}}_i : (w_i, wa_i,\ \mathit{linear}(W_i)).\mathit{Machine}_i(W_i)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Timer}_i \;&\stackrel{\mathit{def}}{=}\; \underline{\mathrm{init}} : (t_i, 0, \mathit{const}).\mathit{Timer}_i + \\
&\quad \underline{\mathrm{take}}_i : (t_i, 1, \mathit{const}).\mathit{Timer}_i + \\
&\quad \underline{\mathrm{assem}}_i : (t_i, 0, \mathit{const}).\mathit{Timer}_i
\end{aligned}
$$

Uncontrolled system

$$
\begin{aligned}
\mathit{Machine}_i(W_i) \;\stackrel{def}{=}\;& \underline{\mathrm{init}} : (w_i, wa_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
& \overline{\mathrm{prep}}_i : (w_i, 0, \mathit{const}).\mathit{Machine}_i(W_i) + \\
& \underline{\mathrm{take}}_i : (w_i, wt_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
& \underline{\mathrm{assem}}_i : (w_i, wa_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Timer}_i \;\stackrel{def}{=}\;& \underline{\mathrm{init}} : (t_i, 0, \mathit{const}).\mathit{Timer}_i + \\
& \underline{\mathrm{take}}_i : (t_i, 1, \mathit{const}).\mathit{Timer}_i + \\
& \underline{\mathrm{assem}}_i : (t_i, 0, \mathit{const}).\mathit{Timer}_i
\end{aligned}
$$

Uncontrolled system

$$
\begin{aligned}
\mathit{Machine}_i(W_i) &\stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (w_i, wa_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
&\quad \overline{\mathrm{prep}}_i : (w_i, 0, \mathit{const}).\mathit{Machine}_i(W_i) + \\
&\quad \underline{\mathrm{take}}_i : (w_i, wt_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
&\quad \underline{\mathrm{assem}}_i : (w_i, wa_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Timer}_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (t_i, 0, \mathit{const}).\mathit{Timer}_i + \\
&\quad \underline{\mathrm{take}}_i : (t_i, 1, \mathit{const}).\mathit{Timer}_i + \\
&\quad \underline{\mathrm{assem}}_i : (t_i, 0, \mathit{const}).\mathit{Timer}_i
\end{aligned}
$$

Uncontrolled system (continued)

$$\begin{aligned} \mathit{Feed}_i & \stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (p_i, \mathit{arrivals}_i, \mathit{const}).\mathit{Feed}_i + \\ & \quad \underline{\mathrm{full}} : (p_i, 0, \mathit{const}).\mathit{Feed}_i \end{aligned}$$

Uncontrolled system (continued)

$$
\begin{array}{rcl}
\mathit{Feed}_i & \stackrel{\mathit{def}}{=} & \underline{\mathrm{init}} : (p_i, \mathit{arrivals}_i, \mathit{const}).\mathit{Feed}_i + \\
 & & \underline{\mathrm{full}} : (p_i, 0, \mathit{const}).\mathit{Feed}_i
\end{array}
$$

$$
\begin{array}{rcl}
\mathit{Output} & \stackrel{\mathit{def}}{=} & \underline{\mathrm{init}} : (b, \mathit{departures}, \mathit{const}).\mathit{Output} + \\
 & & \underline{\mathrm{full}} : (b, 0, \mathit{const}).\mathit{Output}
\end{array}
$$

Uncontrolled system (continued)

$$\begin{aligned} \mathit{Feed}_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (p_i, \mathit{arrivals}_i, \mathit{const}).\mathit{Feed}_i + \\ &\quad\; \underline{\mathrm{full}} : (p_i, 0, \mathit{const}).\mathit{Feed}_i \end{aligned}$$

$$\begin{aligned} \mathit{Output} &\stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (b, \mathit{departures}, \mathit{const}).\mathit{Output} + \\ &\quad\; \underline{\mathrm{full}} : (b, 0, \mathit{const}).\mathit{Output} \end{aligned}$$

Uncontrolled system (continued)

$$
\begin{aligned}
\mathit{Feed}_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (p_i, \mathit{arrivals}_i, \mathit{const}).\mathit{Feed}_i + \\
&\phantom{\stackrel{\mathit{def}}{=}} \underline{\mathrm{full}} : (p_i, 0, \mathit{const}).\mathit{Feed}_i
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Output} &\stackrel{\mathit{def}}{=} \underline{\mathrm{init}} : (b, \mathit{departures}, \mathit{const}).\mathit{Output} + \\
&\phantom{\stackrel{\mathit{def}}{=}} \underline{\mathrm{full}} : (b, 0, \mathit{const}).\mathit{Output}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Sys} \stackrel{\mathit{def}}{=} & (\mathit{Feed}_1 \underset{*}{\bowtie} \mathit{Feed}_2 \underset{*}{\bowtie} \mathit{Feed}_3) \underset{*}{\bowtie} \\
& \mathit{Output} \underset{*}{\bowtie} \\
& (\mathit{Timer}_1 \underset{*}{\bowtie} \mathit{Machine}_1(W_1)) \underset{*}{\bowtie} \\
& (\mathit{Timer}_2 \underset{*}{\bowtie} \mathit{Machine}_2(W_2))
\end{aligned}
$$

Mapping of influences, event conditions, influence types

$$iv(p_i) \;=\; P \quad iv(b) = B \quad iv(w_i) = W_i \quad iv(t_i) = T_i$$

Mapping of influences, event conditions, influence types

$$iv(p_i) \;=\; P \quad iv(b) \;=\; B \quad iv(w_i) \;=\; W_i \quad iv(t_i) \;=\; T_i$$

$$ec(\underline{\mathrm{init}}) \;=\; (true, \qquad P' = P_0 \wedge T_i' = 0 \wedge W_i' = 10 \wedge B' = B_0)$$

Mapping of influences, event conditions, influence types

$$iv(p_i) = P \quad iv(b) = B \quad iv(w_i) = W_i \quad iv(t_i) = T_i$$

$$\begin{aligned}
ec(\underline{\mathrm{init}}) &= (\mathit{true}, & & P' = P_0 \wedge T_i' = 0 \wedge W_i' = 10 \wedge B' = B_0) \\[2ex]
ec(\underline{\mathrm{full}}) &= (B \geq B_f, & & \mathit{true}) \\
ec(\underline{\mathrm{take}}_i) &= (P \geq n_i, & & P' = P - n_i \wedge T_i' = 0) \\
ec(\underline{\mathrm{assem}}_i) &= (T_i \geq \mathit{atime}_i, & & B' = B + m_i)
\end{aligned}$$

Mapping of influences, event conditions, influence types

$$iv(p_i) \;=\; P \quad iv(b) \;=\; B \quad iv(w_i) \;=\; W_i \quad iv(t_i) \;=\; T_i$$

$$\begin{array}{rcll}
ec(\underline{\mathrm{init}}) & = & (true, & P' = P_0 \wedge T_i' = 0 \wedge W_i' = 10 \wedge B' = B_0) \\
 & & & \\
ec(\underline{\mathrm{full}}) & = & (B \geq B_f, & true) \\
ec(\underline{\mathrm{take}}_i) & = & (P \geq n_i, & P' = P - n_i \wedge T_i' = 0) \\
ec(\underline{\mathrm{assem}}_i) & = & (T_i \geq atime_i, & B' = B + m_i) \\
 & & & \\
ec(\overline{\mathrm{prep}}_i) & = & (prepare, & true)
\end{array}$$

Mapping of influences, event conditions, influence types

$$iv(p_i) = P \quad iv(b) = B \quad iv(w_i) = W_i \quad iv(t_i) = T_i$$

$$\begin{aligned}
ec(\underline{\mathrm{init}}) &= (true, && P' = P_0 \wedge T_i' = 0 \wedge W_i' = 10 \wedge B' = B_0) \\
ec(\underline{\mathrm{full}}) &= (B \geq B_f, && true) \\
ec(\underline{\mathrm{take}}_i) &= (P \geq n_i, && P' = P - n_i \wedge T_i' = 0) \\
ec(\underline{\mathrm{assem}}_i) &= (T_i \geq atime_i, && B' = B + m_i) \\
ec(\overline{\mathrm{prep}}_i) &= (prepare, && true)
\end{aligned}$$

$$[\![const]\!] = 1 \qquad [\![linear(X)]\!] = X$$

Stochastic HYPE model

uncontrolled system — controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

Stochastic HYPE model

uncontrolled system controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(\mathit{Con}_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} \mathit{Con}_m\big)$$

Stochastic HYPE model

uncontrolled system | controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(\mathit{Con}_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} \mathit{Con}_m\big)$$

controller grammar

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

controller grammar

$$M ::= \mathrm{a}.M \mid 0 \mid M + M$$

Stochastic HYPE model

uncontrolled system

controllers/sequencers

$$\left(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\right) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\left(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\right)$$

controller grammar

$$M ::= \mathrm{a}.M \mid 0 \mid M + M$$

Stochastic HYPE model

uncontrolled system | controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(\mathit{Con}_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} \mathit{Con}_m\big)$$

controller grammar

$$M ::= \mathrm{a}.M \mid 0 \mid M + M$$
$$\mathit{Con} ::= M \mid \mathit{Con} \underset{*}{\bowtie} \mathit{Con}$$

Controllers and controlled system

$$
\begin{aligned}
AOff_i &\stackrel{\mathit{def}}{=} \overline{\mathrm{prep}}_i.AOn_i \\
AOn_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{take}}_i.AProc_i \\
AProc_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{assem}}_i.AOff_i
\end{aligned}
$$

Controllers and controlled system

$$
\begin{aligned}
\mathit{AOff}_i &\stackrel{\mathit{def}}{=} \overline{\mathrm{prep}}_i.\mathit{AOn}_i \\
\mathit{AOn}_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{take}}_i.\mathit{AProc}_i \\
\mathit{AProc}_i &\stackrel{\mathit{def}}{=} \underline{\mathrm{assem}}_i.\mathit{AOff}_i
\end{aligned}
$$

$$
\mathit{FC} \stackrel{\mathit{def}}{=} \underline{\mathrm{full}}.0
$$

Controllers and controlled system

$$
\begin{aligned}
AOff_i &\stackrel{def}{=} \overline{\mathrm{prep}}_i.AOn_i \\
AOn_i &\stackrel{def}{=} \underline{\mathrm{take}}_i.AProc_i \\
AProc_i &\stackrel{def}{=} \underline{\mathrm{assem}}_i.AOff_i
\end{aligned}
$$

$$
FC \stackrel{def}{=} \underline{\mathrm{full}}.0
$$

$$
Assembler \stackrel{def}{=} Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC)
$$

Transition-driven stochastic hybrid automata

- semantics of stochastic HYPE models
- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes

Transition-driven stochastic hybrid automata

- semantics of stochastic HYPE models
- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes
- set of modes, Q and set of continuous variables, $\mathbf{X}$

Transition-driven stochastic hybrid automata

- semantics of stochastic HYPE models
- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes
- set of modes, Q and set of continuous variables, $\mathbf{X}$
- instantaneous transitions
  - source mode, target mode, event name
  - guard: activation condition over variables
  - reset: function determining new values of variables
  - priority/weight: to resolve non-determinism

Transition-driven stochastic hybrid automata

- semantics of stochastic HYPE models
- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes
- set of modes, Q and set of continuous variables, $\mathbf{X}$
- instantaneous transitions
  - source mode, target mode, event name
  - guard: activation condition over variables
  - reset: function determining new values of variables
  - priority/weight: to resolve non-determinism
- stochastic transitions
  - source mode, target mode, event name
  - rate: function defining speed of transition
  - guard: activation condition over variables
  - reset: function determining new values of variables

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$
- instantaneous behaviour: fire when guard becomes true

Transition-driven stochastic hybrid automata (continued)

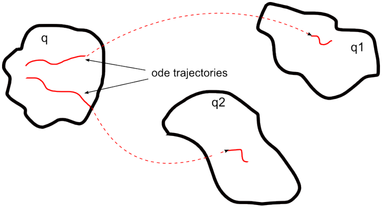
- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$
- instantaneous behaviour: fire when guard becomes true
- stochastic behaviour: fire according to rate

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$
- instantaneous behaviour: fire when guard becomes true
- stochastic behaviour: fire according to rate
- product of TDSHAs
  - pairs of modes and union of variables
  - combining transitions
    (with conditions on resets and initial values)

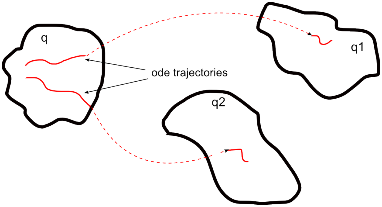
Piecewise deterministic Markov processes

- class of stochastic processes
- continuous trajectories over subsets of $\mathbb{R}^{|\mathbf{X}|}$
- instantaneous jumps at boundaries of regions
- stochastic jumps when guards are true



Piecewise deterministic Markov processes

- class of stochastic processes
- continuous trajectories over subsets of $\mathbb{R}^{|\mathbf{X}|}$
- instantaneous jumps at boundaries of regions
- stochastic jumps when guards are true



- jumps to boundaries are prohibited

Two equivalent semantics

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_{\sigma} = \sum \left\{ r \cdot [\![I(\overrightarrow{W})]\!] \;\middle|\; iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \right\}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_{\sigma} = \sum \left\{ r \cdot [\![I(\overrightarrow{W})]\!] \;\middle|\; iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \right\}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_\sigma = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \bowtie_{*} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_{\sigma} = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_{\sigma} = \sum \left\{ r \cdot [\![I(\overrightarrow{W})]\!] \;\middle|\; iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \right\}$$

Operational semantics

Prefix with influence:

$$\frac{}{\langle \underline{\mathrm{a}}\!:\!(\iota, r, I).E, \sigma\rangle \xrightarrow{\mathsf{a}} \langle E, \sigma[\iota \mapsto (r, I)]\rangle}$$

Prefix without influence:

$$\frac{}{\langle \underline{\mathrm{a}}.E, \sigma\rangle \xrightarrow{\mathsf{a}} \langle E, \sigma\rangle}$$

Choice:

$$\frac{\langle E, \sigma\rangle \xrightarrow{\mathsf{a}} \langle E', \sigma'\rangle}{\langle E + F, \sigma\rangle \xrightarrow{\mathsf{a}} \langle E', \sigma'\rangle} \qquad \frac{\langle F, \sigma\rangle \xrightarrow{\mathsf{a}} \langle F', \sigma'\rangle}{\langle E + F, \sigma\rangle \xrightarrow{\mathsf{a}} \langle F', \sigma'\rangle}$$

Constant:

$$\frac{\langle E, \sigma\rangle \xrightarrow{\mathsf{a}} \langle E', \sigma'\rangle}{\langle A, \sigma\rangle \xrightarrow{\mathsf{a}} \langle E', \sigma'\rangle}(A \stackrel{\mathit{def}}{=} E)$$

Operational semantics

Prefix with influence:

$$\frac{}{\langle \underline{\mathrm{a}}:(\iota, r, I).E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E, \sigma[\iota \mapsto (r, I)] \rangle}$$

Prefix without influence:

$$\frac{}{\langle \underline{\mathrm{a}}.E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E, \sigma \rangle}$$

Choice:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle}{\langle E + F, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle} \qquad \frac{\langle F, \sigma \rangle \xrightarrow{\mathrm{a}} \langle F', \sigma' \rangle}{\langle E + F, \sigma \rangle \xrightarrow{\mathrm{a}} \langle F', \sigma' \rangle}$$

Constant:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle}{\langle A, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle} (A \stackrel{def}{=} E)$$

Operational semantics

Prefix with influence:

$$\frac{}{\langle \underline{\mathrm{a}} : (\iota, r, I).E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E, \sigma[\iota \mapsto (r, I)] \rangle}$$

Prefix without influence:

$$\frac{}{\langle \underline{\mathrm{a}}.E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E, \sigma \rangle}$$

Choice:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle}{\langle E + F, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle} \qquad \frac{\langle F, \sigma \rangle \xrightarrow{\mathrm{a}} \langle F', \sigma' \rangle}{\langle E + F, \sigma \rangle \xrightarrow{\mathrm{a}} \langle F', \sigma' \rangle}$$

Constant:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle}{\langle A, \sigma \rangle \xrightarrow{\mathrm{a}} \langle E', \sigma' \rangle}(A \stackrel{def}{=} E)$$

Operational semantics (continued)

Parallel without synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\underline{a}} \langle E', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{\underline{a}} \langle E' \underset{M}{\bowtie} F, \sigma' \rangle} \quad \underline{a} \notin M$$

$$\frac{\langle F, \sigma \rangle \xrightarrow{\underline{a}} \langle F', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{\underline{a}} \langle E \underset{M}{\bowtie} F', \sigma' \rangle} \quad \underline{a} \notin M$$

Parallel with synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\underline{a}} \langle E', \tau \rangle \quad \langle F, \sigma \rangle \xrightarrow{\underline{a}} \langle F', \tau' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{\underline{a}} \langle E' \underset{M}{\bowtie} F', \Gamma(\sigma, \tau, \tau') \rangle}$$

$\underline{a} \in M, \Gamma$ defined

Operational semantics (continued)

Parallel without synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle E', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle E' \underset{M}{\bowtie} F, \sigma' \rangle} \quad \underline{\mathrm{a}} \notin M$$

$$\frac{\langle F, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle F', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle E \underset{M}{\bowtie} F', \sigma' \rangle} \quad \underline{\mathrm{a}} \notin M$$

Parallel with synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle E', \tau \rangle \quad \langle F, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle F', \tau' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{\underline{\text{a}}} \langle E' \underset{M}{\bowtie} F', \Gamma(\sigma, \tau, \tau') \rangle}$$

$\underline{\mathrm{a}} \in M, \Gamma$ defined

Operational semantics (continued)

- updating function: $\sigma[\iota \mapsto (r, I)]$

$$\sigma[\iota \mapsto (r, I)](x) = \begin{cases} (r, I) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise} \end{cases}$$

Operational semantics (continued)

- updating function: $\sigma[\iota \mapsto (r, I)]$

$$\sigma[\iota \mapsto (r, I)](x) = \begin{cases} (r, I) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise} \end{cases}$$

- change identifying function: $\Gamma : \mathcal{S} \times \mathcal{S} \times \mathcal{S} \to \mathcal{S}$

$$(\Gamma(\sigma, \tau, \tau'))(\iota) = \begin{cases} \tau(\iota) & \text{if } \sigma(\iota) = \tau'(\iota) \\ \tau'(\iota) & \text{if } \sigma(\iota) = \tau(\iota) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Operational semantics (continued)

- updating function: $\sigma[\iota \mapsto (r, I)]$

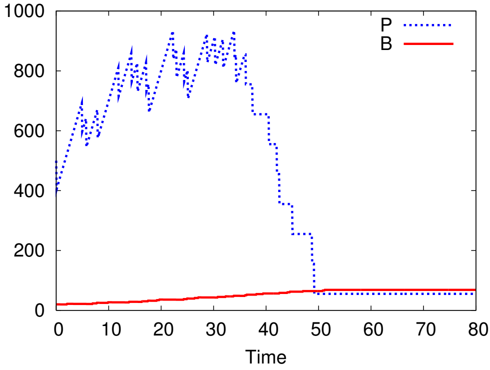
$$\sigma[\iota \mapsto (r, I)](x) = \begin{cases} (r, I) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise} \end{cases}$$

- change identifying function: $\Gamma : \mathcal{S} \times \mathcal{S} \times \mathcal{S} \to \mathcal{S}$

$$(\Gamma(\sigma, \tau, \tau'))(\iota) = \begin{cases} \tau(\iota) & \text{if } \sigma(\iota) = \tau'(\iota) \\ \tau'(\iota) & \text{if } \sigma(\iota) = \tau(\iota) \\ \text{undefined} & \text{otherwise} \end{cases}$$

- Γ is defined for all well-defined stochastic HYPE models
  - syntactic restrictions on influences and events

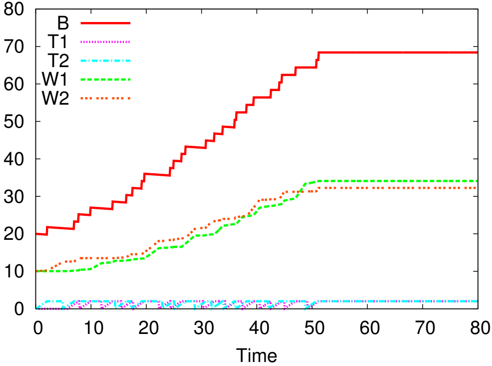
Simulation of assembly system using SimHyA



$$Sys \bowtie_{*} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC)$$

$(arrivals_i{=}20,\ departures{=}-0.1,\ atime_i{=}2,\ prepare{=}0.6,\ n_i{=}100,\ m_i{=}2,\ wt_i{=}0.01,\ wa_i{=}0.06)$

Simulation of assembly system using SimHyA



$$Sys \bowtie_{*} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC)$$

$(arrivals_i{=}20,\ departures{=}{-0.1},\ atime_i{=}2,\ prepare{=}0.6,\ n_i{=}100,\ m_i{=}2,\ wt_i{=}0.01,\ wa_i{=}0.06)$

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution
- when is a stochastic HYPE model well-behaved?

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution
- when is a stochastic HYPE model well-behaved?
  - construct I-graph (instantaneous activation graph)

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution
- when is a stochastic HYPE model well-behaved?
  - construct I-graph (instantaneous activation graph)
  - check for cycles in I-graph

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution
- when is a stochastic HYPE model well-behaved?
    - construct I-graph (instantaneous activation graph)
    - check for cycles in I-graph
    - can be done without simulating the model

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid *instantaneous Zeno behaviour:* infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution
- when is a stochastic HYPE model well-behaved?
  - construct I-graph (instantaneous activation graph)
  - check for cycles in I-graph
  - can be done without simulating the model
- well-behaved results for overapproximations and compositions

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \textit{true}\}$

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{\mathrm{a}}} = g'_{\underline{\mathrm{a}}}(G_{\underline{\mathrm{a}}})$

I-graph construction

- use guards to define $G_{\underline{a}} = \{\mathbf{x} \mid g_{\underline{a}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{a}} = g'_{\underline{a}}(G_{\underline{a}})$
- enabled events are those whose guards can be satisfied

I-graph construction

- use guards to define $G_{\underline{a}} = \{\mathbf{x} \mid g_{\underline{a}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{a}} = g'_{\underline{a}}(G_{\underline{a}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{\mathrm{a}}} = g'_{\underline{\mathrm{a}}}(G_{\underline{\mathrm{a}}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events
- consider each sequence of $\underline{\mathrm{a}}$ and $\underline{\mathrm{b}}$ that the controller allows

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{\mathrm{a}}} = g'_{\underline{\mathrm{a}}}(G_{\underline{\mathrm{a}}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events
- consider each sequence of $\underline{\mathrm{a}}$ and $\underline{\mathrm{b}}$ that the controller allows
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} = \emptyset$, event $\underline{\mathrm{b}}$ is inhibited, so there no arc in I-graph

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{\mathrm{a}}} = g'_{\underline{\mathrm{a}}}(G_{\underline{\mathrm{a}}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events
- consider each sequence of $\underline{\mathrm{a}}$ and $\underline{\mathrm{b}}$ that the controller allows
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} = \emptyset$, event $\underline{\mathrm{b}}$ is inhibited, so there no arc in I-graph
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} \neq \emptyset$, event $\underline{\mathrm{b}}$ is enabled, so there is an arc in I-graph

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{\mathrm{a}}} = g'_{\underline{\mathrm{a}}}(G_{\underline{\mathrm{a}}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events
- consider each sequence of $\underline{\mathrm{a}}$ and $\underline{\mathrm{b}}$ that the controller allows
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} = \emptyset$, event $\underline{\mathrm{b}}$ is inhibited, so there no arc in I-graph
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} \neq \emptyset$, event $\underline{\mathrm{b}}$ is enabled, so there is an arc in I-graph
- initially assume every instantaneous event is enabled

I-graph construction

- use guards to define $G_{\underline{\mathrm{a}}} = \{\mathbf{x} \mid g_{\underline{\mathrm{a}}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{\mathrm{a}}} = g'_{\underline{\mathrm{a}}}(G_{\underline{\mathrm{a}}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events
- consider each sequence of $\underline{\mathrm{a}}$ and $\underline{\mathrm{b}}$ that the controller allows
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} = \emptyset$, event $\underline{\mathrm{b}}$ is inhibited, so there no arc in I-graph
- if $R_{\underline{\mathrm{a}}} \cap G_{\underline{\mathrm{b}}} \neq \emptyset$, event $\underline{\mathrm{b}}$ is enabled, so there is an arc in I-graph
- initially assume every instantaneous event is enabled
- no cycle implies no instantaneous Zeno behaviour

I-graph construction

- use guards to define $G_{\underline{a}} = \{\mathbf{x} \mid g_{\underline{a}}(\mathbf{x}) \text{ is } \mathit{true}\}$
- use resets to define $R_{\underline{a}} = g'_{\underline{a}}(G_{\underline{a}})$
- enabled events are those whose guards can be satisfied
- keep track of enabled and inhibited events
- consider each sequence of $\underline{a}$ and $\underline{b}$ that the controller allows
- if $R_{\underline{a}} \cap G_{\underline{b}} = \emptyset$, event $\underline{b}$ is inhibited, so there no arc in I-graph
- if $R_{\underline{a}} \cap G_{\underline{b}} \neq \emptyset$, event $\underline{b}$ is enabled, so there is an arc in I-graph
- initially assume every instantaneous event is enabled
- no cycle implies no instantaneous Zeno behaviour
- I-graph construction is not always necessary

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{def}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{\mathit{def}}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

- machine controllers are cycles with a stochastic event

$$AOff_i \stackrel{\mathit{def}}{=} \overline{\mathrm{prep}}_i.\underline{\mathrm{take}}_i.\underline{\mathrm{assem}}_i.AOff_i \qquad \text{well-behaved}$$

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{\mathit{def}}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

- machine controllers are cycles with a stochastic event

$$AOff_i \stackrel{\mathit{def}}{=} \overline{\mathrm{prep}}_i.\underline{\mathrm{take}}_i.\underline{\mathrm{assem}}_i.AOff_i \qquad \text{well-behaved}$$

- $AOff_1 \parallel AOff_2$ is well-behaved

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{def}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

- machine controllers are cycles with a stochastic event

$$AOff_i \stackrel{def}{=} \overline{\mathrm{prep}}_i.\underline{\mathrm{take}}_i.\underline{\mathrm{assem}}_i.AOff_i \qquad \text{well-behaved}$$

- $AOff_1 \parallel AOff_2$ is well-behaved
  - neither $\underline{\mathrm{take}}_1$ or $\underline{\mathrm{assem}}_1$ enable $\underline{\mathrm{take}}_2$ or $\underline{\mathrm{assem}}_2$ or *vice versa*

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{def}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

- machine controllers are cycles with a stochastic event

$$AOff_i \stackrel{def}{=} \overline{\mathrm{prep}}_i.\underline{\mathrm{take}}_i.\underline{\mathrm{assem}}_i.AOff_i \qquad \text{well-behaved}$$

- $AOff_1 \parallel AOff_2$ is well-behaved
  - neither $\underline{\mathrm{take}}_1$ or $\underline{\mathrm{assem}}_1$ enable $\underline{\mathrm{take}}_2$ or $\underline{\mathrm{assem}}_2$ or *vice versa*
- $AOff_1 \parallel AOff_2 \parallel FC$ is well-behaved

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{def}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

- machine controllers are cycles with a stochastic event

$$AOff_i \stackrel{def}{=} \overline{\mathrm{prep}}_i.\underline{\mathrm{take}}_i.\underline{\mathrm{assem}}_i.AOff_i \qquad \text{well-behaved}$$

- $AOff_1 \parallel AOff_2$ is well-behaved
  - neither $\underline{\mathrm{take}}_1$ or $\underline{\mathrm{assem}}_1$ enable $\underline{\mathrm{take}}_2$ or $\underline{\mathrm{assem}}_2$ or *vice versa*
- $AOff_1 \parallel AOff_2 \parallel FC$ is well-behaved
  - none of $\underline{\mathrm{take}}_i$ or $\underline{\mathrm{assem}}_i$ enable $\underline{\mathrm{full}}$ or *vice versa*

Well-behavedness of assembly system

- controller that checks for full belt has only one event

$$FC \stackrel{\mathit{def}}{=} \underline{\mathrm{full}}.0 \qquad \text{well-behaved}$$

- machine controllers are cycles with a stochastic event

$$AOff_i \stackrel{\mathit{def}}{=} \overline{\mathrm{prep}}_i.\underline{\mathrm{take}}_i.\underline{\mathrm{assem}}_i.AOff_i \qquad \text{well-behaved}$$

- $AOff_1 \parallel AOff_2$ is well-behaved
  - neither $\underline{\mathrm{take}}_1$ or $\underline{\mathrm{assem}}_1$ enable $\underline{\mathrm{take}}_2$ or $\underline{\mathrm{assem}}_2$ or *vice versa*
- $AOff_1 \parallel AOff_2 \parallel FC$ is well-behaved
  - none of $\underline{\mathrm{take}}_i$ or $\underline{\mathrm{assem}}_i$ enable $\underline{\mathrm{full}}$ or *vice versa*
- $Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC)$ is well-behaved

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever
     $\langle P, \sigma\rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma'\rangle \in C$, $\exists\ \langle Q', \tau'\rangle \in C$ with $\langle Q, \tau\rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau'\rangle$
     $\langle Q, \tau\rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau'\rangle \in C$, $\exists\ \langle P', \sigma'\rangle \in C$ with $\langle P, \sigma\rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma'\rangle$.

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever
  $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle \in C$, $\exists\ \langle Q', \tau' \rangle \in C$ with $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle$
  $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle \in C$, $\exists\ \langle P', \sigma' \rangle \in C$ with $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle$.
  2. for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, $r(\langle P, \sigma \rangle, \overline{\mathrm{a}}, C) = r(\langle Q, \tau \rangle, \overline{\mathrm{a}}, C)$.

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever
     $\langle P, \sigma\rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma'\rangle \in C$, $\exists$ $\langle Q', \tau'\rangle \in C$ with $\langle Q, \tau\rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau'\rangle$
     $\langle Q, \tau\rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau'\rangle \in C$, $\exists$ $\langle P', \sigma'\rangle \in C$ with $\langle P, \sigma\rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma'\rangle$.
  2. for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, $r(\langle P, \sigma\rangle, \overline{\mathrm{a}}, C) = r(\langle Q, \tau\rangle, \overline{\mathrm{a}}, C)$.

- notation: $P \sim^{\equiv} Q$

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (for models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever
     $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle \in C$, $\exists\ \langle Q', \tau' \rangle \in C$ with $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle$
     $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle \in C$, $\exists\ \langle P', \sigma' \rangle \in C$ with $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle$.
  2. for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, $r(\langle P, \sigma \rangle, \overline{\mathrm{a}}, C) = r(\langle Q, \tau \rangle, \overline{\mathrm{a}}, C)$.

- notation: $P \sim^{\equiv} Q$
- equivalence defined in terms of labelled transition system and without reference to variable values

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

  then for all $((q_1, \mathbf{x}_1), (q_2, \mathbf{x}_2)) \in B$

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

  then for all $((q_1, \mathbf{x}_1), (q_2, \mathbf{x}_2)) \in B$

  - $\mathrm{out}_1(\mathbf{x}_1) = \mathrm{out}_2(\mathbf{x}_2)$
  - exit rates of q_1 and q_2 must be equal
  - disjunction of guards must evaluate to the same for $\mathbf{x}_1$ and $\mathbf{x}_2$
  - disjunction of guards must become true at the same time
  - for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, one step priorities must match
  - for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, one step probabilities must match

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

  then for all $((q_1, \mathbf{x}_1), (q_2, \mathbf{x}_2)) \in B$
  - $\mathrm{out}_1(\mathbf{x}_1) = \mathrm{out}_2(\mathbf{x}_2)$
  - exit rates of q_1 and q_2 must be equal
  - disjunction of guards must evaluate to the same for $\mathbf{x}_1$ and $\mathbf{x}_2$
  - disjunction of guards must become true at the same time
  - for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, one step priorities must match
  - for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, one step probabilities must match
- notation: $\mathcal{T}_1 \sim_T^{\ell} \mathcal{T}_2$

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)
- if $\mathit{Con}_1 \sim^{\equiv} \mathit{Con}_2$ then $\mathit{Sys} \underset{*}{\bowtie} \underline{\mathrm{init}}.\mathit{Con}_1 \sim^{\equiv} \mathit{Sys} \underset{*}{\bowtie} \underline{\mathrm{init}}.\mathit{Con}_2$

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)
- if $Con_1 \sim^{\equiv} Con_2$ then $Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.Con_1 \sim^{\equiv} Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.Con_2$
- *additively equivalent*: $\sigma \doteq \tau$ iff for all $V \in \mathcal{V}$ and $f(\mathcal{W})$

$$\mathrm{sum}(\sigma, V, f(\mathcal{W})) = \mathrm{sum}(\tau, V, f(\mathcal{W}))$$

where $\mathrm{sum}(\sigma, V, f(W)) \; =$

$$\sum \{\!|\, r \mid iv(\iota) = V, \sigma(\iota) = (r, I(W)), f(\mathcal{W}) = [\![I(W)]\!] \,|\!\}$$

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)
- if $Con_1 \sim^{\equiv} Con_2$ then $Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.Con_1 \sim^{\equiv} Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.Con_2$
- *additively equivalent*: $\sigma \doteq \tau$ iff for all $V \in \mathcal{V}$ and $f(\mathcal{W})$

  $$\mathrm{sum}(\sigma, V, f(\mathcal{W})) = \mathrm{sum}(\tau, V, f(\mathcal{W}))$$

  where $\mathrm{sum}(\sigma, V, f(W)) \; =$

  $$\sum \{\!| \, r \mid iv(\iota) = V, \sigma(\iota) = (r, I(W)), f(\mathcal{W}) = [\![I(W)]\!] \, |\!\}$$

- $P_1 \sim^{\doteq} P_2$ implies $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$

Results applied to assembly system

- *ABOff*: single controller of two machines

Results applied to assembly system

- *ABOff*: single controller of two machines
- can prove that $\mathit{AOff}_1 \parallel \mathit{AOff}_2 \sim^{=} \mathit{ABOff}$

Results applied to assembly system

- *ABOff*: single controller of two machines
- can prove that $AOff_1 \parallel AOff_2 \sim^{=} ABOff$
- hence using congruence

$$Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC) \sim^{=} Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(ABOff \parallel FC)$$

Results applied to assembly system

- ABOff: single controller of two machines
- can prove that $\mathit{AOff}_1 \parallel \mathit{AOff}_2 \sim^{=} \mathit{ABOff}$
- hence using congruence

$$\mathit{Sys} \underset{*}{\bowtie} \underline{\mathrm{init}}.(\mathit{AOff}_1 \parallel \mathit{AOff}_2 \parallel \mathit{FC}) \sim^{=} \mathit{Sys} \underset{*}{\bowtie} \underline{\mathrm{init}}.(\mathit{ABOff} \parallel \mathit{FC})$$

- define a single feed subcomponent with $iv(p) = P$

$$\begin{aligned} \mathit{SFeed} \stackrel{\mathit{def}}{=} \; & \underline{\mathrm{init}} : (p, \textstyle\sum_{k=1}^{3} \mathit{arrivals}_i, \mathit{const}).\mathit{SFeed} + \\ & \underline{\mathrm{full}} : (p, 0, \mathit{const}).\mathit{SFeed} \end{aligned}$$

Results applied to assembly system

- $ABOff$: single controller of two machines
- can prove that $AOff_1 \parallel AOff_2 \sim^{=} ABOff$
- hence using congruence

$$Sys \bowtie_{*} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC) \sim^{=} Sys \bowtie_{*} \underline{\mathrm{init}}.(ABOff \parallel FC)$$

- define a single feed subcomponent with $iv(p) = P$

$$\begin{aligned} SFeed \quad &\stackrel{def}{=} \quad \underline{\mathrm{init}} : (p, \textstyle\sum_{k=1}^{3} arrivals_i, const).SFeed + \\ & \qquad \underline{\mathrm{full}} : (p, 0, const).SFeed \end{aligned}$$

- Sys_{SF} has $(Feed_1 \bowtie_{*} Feed_2 \bowtie_{*} Feed_3)$ replaced with $SFeed$

Results applied to assembly system

- *ABOff*: single controller of two machines
- can prove that $AOff_1 \parallel AOff_2 \; \sim^{=} \; ABOff$
- hence using congruence

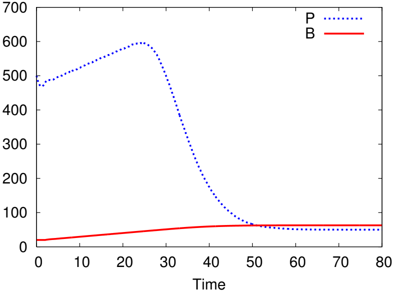
$$Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(AOff_1 \parallel AOff_2 \parallel FC) \sim^{=} Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(ABOff \parallel FC)$$

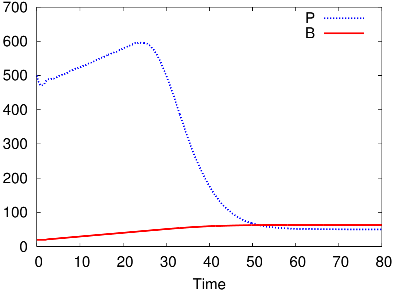
- define a single feed subcomponent with $iv(p) = P$

$$\begin{aligned} SFeed \quad &\stackrel{def}{=} \quad \underline{\mathrm{init}} : (p, \textstyle\sum_{k=1}^{3} arrivals_i, const).SFeed \, + \\ & \qquad \underline{\mathrm{full}} : (p, 0, const).SFeed \end{aligned}$$

- Sys_{SF} has $(Feed_1 \underset{*}{\bowtie} Feed_2 \underset{*}{\bowtie} Feed_3)$ replaced with *SFeed*
- then $Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.Con \;\; \sim^{\doteq} \;\; Sys_{SF} \underset{*}{\bowtie} \underline{\mathrm{init}}.Con$

Two equivalent controllers





$Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.((AOff_1 \parallel AOff_2 \parallel FC)$

$Sys \underset{*}{\bowtie} \underline{\mathrm{init}}.(ABOff \parallel FC)$

averages of 5000 simulations

$(arrivals_i{=}20,\ departures{=}-0.1,\ atime_i{=}2,\ prepare{=}0.6,\ n_i{=}100,\ m_i{=}2,\ wt_i{=}0.01)$

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?
- no, consider two different assembly system models

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?
- no, consider two different assembly system models
- M: individual timers T_i that are set to zero as assembly starts, with guards to check whether $T_i \geq atime_i$

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?
- no, consider two different assembly system models
- M: individual timers T_i that are set to zero as assembly starts, with guards to check whether $T_i \geq atime_i$
- M': single timer T whose value is stored in S_i as assembly starts, with guards to check whether $T \geq S_i + atime_i$

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?
- no, consider two different assembly system models
- M: individual timers T_i that are set to zero as assembly starts, with guards to check whether $T_i \geq \mathit{atime}_i$
- M': single timer T whose value is stored in S_i as assembly starts, with guards to check whether $T \geq S_i + \mathit{atime}_i$
- can show that at the TDSHA level, $\mathcal{T}(M) \sim^{\ell}_{T} \mathcal{T}(M')$

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?
- no, consider two different assembly system models
- M: individual timers T_i that are set to zero as assembly starts, with guards to check whether $T_i \geq \mathit{atime}_i$
- M': single timer T whose value is stored in S_i as assembly starts, with guards to check whether $T \geq S_i + \mathit{atime}_i$
- can show that at the TDSHA level, $\mathcal{T}(M) \sim^{\ell}_{T} \mathcal{T}(M')$
- but $M \not\sim^{\doteq} M'$ since $\underline{\mathrm{take}}_i$ and $\underline{\mathrm{assem}}_i$ have different event conditions in M_1 and M_2 so definition does not apply

Results applied to assembly system (continued)

- does $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$ imply $P_1 \sim^{\doteq} P_2$?
- no, consider two different assembly system models
- M: individual timers T_i that are set to zero as assembly starts, with guards to check whether $T_i \geq atime_i$
- M': single timer T whose value is stored in S_i as assembly starts, with guards to check whether $T \geq S_i + atime_i$
- can show that at the TDSHA level, $\mathcal{T}(M) \sim^{\ell}_{T} \mathcal{T}(M')$
- but $M \not\sim^{\doteq} M'$ since $\underline{\mathrm{take}}_i$ and $\underline{\mathrm{assem}}_i$ have different event conditions in M_1 and M_2 so definition does not apply
- correct definition of bisimilarity?

Other applications of stochastic HYPE

- biological systems
  - Repressilator: 3 gene system with inhibition
  - circadian clock of *Ostreococcus tauri*

Other applications of stochastic HYPE

- biological systems
  - Repressilator: 3 gene system with inhibition
  - circadian clock of *Ostreococcus tauri*
- human-constructed systems
  - planetary orbiter
  - railway crossing (train gate)
  - opportunistic networks

Other applications of stochastic HYPE

- biological systems
  - Repressilator: 3 gene system with inhibition
  - circadian clock of *Ostreococcus tauri*
- human-constructed systems
  - planetary orbiter
  - railway crossing (train gate)
  - opportunistic networks
- combined systems
  - Zebranet: MSc dissertation of Cheng Feng

Zebranet modelling

- animal-based opportunistic network
  - collect movement data from zebra with low human intervention
  - data is communicated from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model

Zebranet modelling

- animal-based opportunistic network
  - collect movement data from zebra with low human intervention
  - data is communicated from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model
- syntactic extension to allow automatic generation of repeated subcomponents and controllers

Zebranet modelling

- animal-based opportunistic network
  - collect movement data from zebra with low human intervention
  - data is communicated from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model
- syntactic extension to allow automatic generation of repeated subcomponents and controllers
- model elements

Zebranet modelling

- animal-based opportunistic network
  - collect movement data from zebra with low human intervention
  - data is communicated from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model
- syntactic extension to allow automatic generation of repeated subcomponents and controllers
- model elements
  - two-dimensional model of zebra movement in terms of seconds

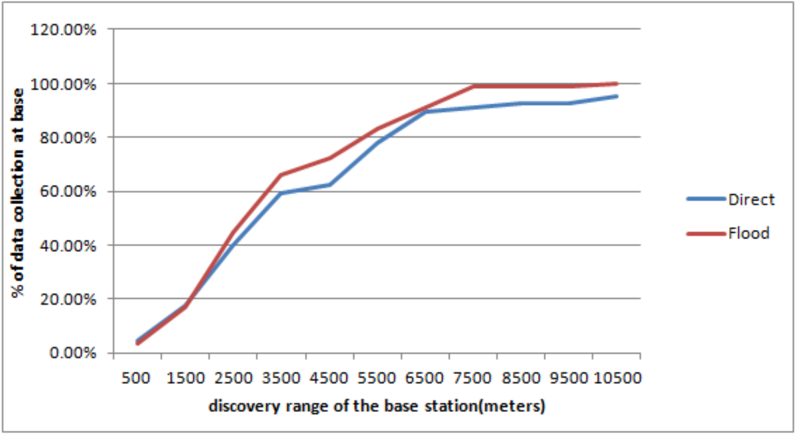
Zebranet modelling

- animal-based opportunistic network
  - collect movement data from zebra with low human intervention
  - data is communicated from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model
- syntactic extension to allow automatic generation of repeated subcomponents and controllers
- model elements
  - two-dimensional model of zebra movement in terms of seconds
  - model of energy consumption for collar equipment

Zebranet modelling

- animal-based opportunistic network
  - collect movement data from zebra with low human intervention
  - data is communicated from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model
- syntactic extension to allow automatic generation of repeated subcomponents and controllers
- model elements
  - two-dimensional model of zebra movement in terms of seconds
  - model of energy consumption for collar equipment
  - model of transmission protocol: direct and flooding

Data collected by protocol



Conclusions

- stochastic HYPE
  - process algebra for stochastic hybrid systems
  - semantics given by TDSHA and PDMPs
  - illustrated through assembly system model and Zebranet model

Conclusions

- stochastic HYPE
  - process algebra for stochastic hybrid systems
  - semantics given by TDSHA and PDMPs
  - illustrated through assembly system model and Zebranet model
- well-behaved stochastic HYPE models
  - contain no instantaneous Zeno behaviour
  - can be checked this without model simulation

Conclusions

- stochastic HYPE
  - process algebra for stochastic hybrid systems
  - semantics given by TDSHA and PDMPs
  - illustrated through assembly system model and Zebranet model
- well-behaved stochastic HYPE models
  - contain no instantaneous Zeno behaviour
  - can be checked this without model simulation
- two semantic equivalences
  - stochastic HYPE: equivalence with abstraction over states
  - TDSHA: equivalence based on modes and variable values

Conclusions

- stochastic HYPE
  - process algebra for stochastic hybrid systems
  - semantics given by TDSHA and PDMPs
  - illustrated through assembly system model and Zebranet model
- well-behaved stochastic HYPE models
  - contain no instantaneous Zeno behaviour
  - can be checked this without model simulation
- two semantic equivalences
  - stochastic HYPE: equivalence with abstraction over states
  - TDSHA: equivalence based on modes and variable values
- main results
  - congruence and corollary about equivalent controllers
  - relationship between two equivalences

Thank you