How to use HYPE to model hybrid systems: a railway crossing example

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)

13 December 2010

Outline

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- hybrid automata
  - well known, graphical
  - somewhat compositional

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- hybrid automata
  - well known, graphical
  - somewhat compositional
- process algebras for hybrid systems
  - textual
  - compositional
  - often require monolithic ODEs in syntax
  - HYPE: more fine-grained, individual flows

Train gate system

Train gate system

Train gate system

- standard example for hybrid systems modelling

Train gate system

- standard example for hybrid systems modelling
- railway track with guarded crossing

Train gate system

- standard example for hybrid systems modelling
- railway track with guarded crossing
- track to be modelled
  - starts at least 1500m before crossing
  - ends 100m after crossing

Train gate system

- standard example for hybrid systems modelling
- railway track with guarded crossing
- track to be modelled
  - starts at least 1500m before crossing
  - ends 100m after crossing
- two sensors
  - first: 1000m before crossing sends *approach* signal to controller
  - second: 100m after crossing sends *exit* signal to controller

Train gate system

- standard example for hybrid systems modelling
- railway track with guarded crossing
- track to be modelled
  - starts at least 1500m before crossing
  - ends 100m after crossing
- two sensors
  - first: 1000m before crossing sends *approach* signal to controller
  - second: 100m after crossing sends *exit* signal to controller
- train speed
  - between 48m/s and 52m/s before first sensor
  - between 40m/s and 52m/s after first sensor

Train gate system

- standard example for hybrid systems modelling
- railway track with guarded crossing
- track to be modelled
  - starts at least 1500m before crossing
  - ends 100m after crossing
- two sensors
  - first: 1000m before crossing sends *approach* signal to controller
  - second: 100m after crossing sends *exit* signal to controller
- train speed
  - between 48m/s and 52m/s before first sensor
  - between 40m/s and 52m/s after first sensor
- gate
  - opens and closes at 20°/s

Train gate system (continued)

- controller
  - on receiving *approach* signal, takes up to 5 seconds to send *lower* signal to gate
  - on receiving *exit* signal, takes up to 5 seconds to send *raise* signal to gate

Train gate system (continued)

- controller
  - on receiving *approach* signal, takes up to 5 seconds to send *lower* signal to gate
  - on receiving *exit* signal, takes up to 5 seconds to send *raise* signal to gate
- initial conditions
  - train starts at 1400m before crossing, gate is open (at 90°)

Train gate system (continued)

- controller
  - on receiving *approach* signal, takes up to 5 seconds to send *lower* signal to gate
  - on receiving *exit* signal, takes up to 5 seconds to send *raise* signal to gate
- initial conditions
  - train starts at 1400m before crossing, gate is open (at 90°)
- safety property
  - when the train is 100m before the gate, the gate is closed

Train gate system (continued)

- controller
  - on receiving *approach* signal, takes up to 5 seconds to send *lower* signal to gate
  - on receiving *exit* signal, takes up to 5 seconds to send *raise* signal to gate
- initial conditions
  - train starts at 1400m before crossing, gate is open (at 90°)
- safety property
  - when the train is 100m before the gate, the gate is closed
- assumptions for HYPE model, worst case
  - trains start at 1500m before crossing
  - signals from controller take 5 seconds to be received by gate
  - trains travel at 52m/s

HYPE syntax

- two types of actions

HYPE syntax

- two types of actions
- events: instantaneous, discrete changes

$$\underline{\mathsf{a}} \in \mathcal{E}$$

HYPE syntax

- two types of actions
- events: instantaneous, discrete changes

  $\underline{\text{a}} \in \mathcal{E}$

- activities: influences on continuous aspect of system, flows

  $\alpha \in \mathcal{A} \qquad \alpha(\mathcal{W}) = (\iota, r, I(\mathcal{W}))$

  - influence name $\iota \in IN$
  - rate $r \in \mathbb{R}$
  - influence type $I(\mathcal{W})$ with $[\![I(\mathcal{W})]\!] = f(\mathcal{W})$ with $\mathcal{W} \subseteq \mathcal{V}$

HYPE syntax

- two types of actions
- events: instantaneous, discrete changes

  $\underline{\mathrm{a}} \in \mathcal{E}$

- activities: influences on continuous aspect of system, flows

  $\alpha \in \mathcal{A} \qquad \alpha(\mathcal{W}) = (\iota, r, I(\mathcal{W}))$

  - influence name $\iota \in IN$
  - rate $r \in \mathbb{R}$
  - influence type $I(\mathcal{W})$ with $[\![I(\mathcal{W})]\!] = f(\mathcal{W})$ with $\mathcal{W} \subseteq \mathcal{V}$
- well-defined subcomponent: $\underline{\mathrm{a}}_i \neq \underline{\mathrm{a}}_j$ for $i \neq j$

$$C_s(\mathcal{W}) \stackrel{def}{=} \sum_{i=1}^{n} \underline{\mathrm{a}}_i : (\iota, r_i, I_i(\mathcal{W})).C_s(\mathcal{W}) + \underline{\mathrm{init}} : (\iota, r, I(\mathcal{W})).C_s(\mathcal{W})$$

Continuous aspects of train gate system

- train

$$Train \stackrel{def}{=} \underline{\mathrm{init}}:(d, r_{tr}, c).Train$$

Continuous aspects of train gate system

- train

$$Train \stackrel{def}{=} \underline{\text{init}}:(d, r_{tr}, c).Train$$

- continuous aspects of gate

$$\begin{aligned} Gate \stackrel{def}{=} \; & \underline{\text{lower}}:(g, -r_{gt}, c).Gate + \underline{\text{closed}}:(g, 0, c).Gate \; + \\ & \underline{\text{raise}}:(g, r_{gt}, c).Gate + \underline{\text{open}}:(g, 0, c).Gate \; + \\ & \underline{\text{init}}:(g, 0, c).Gate \end{aligned}$$

Continuous aspects of train gate system

- train

$$Train \stackrel{def}{=} \underline{\text{init}}:(d, r_{tr}, c).Train$$

- continuous aspects of gate

$$\begin{aligned} Gate \stackrel{def}{=} \; & \underline{\text{lower}}:(g, -r_{gt}, c).Gate + \underline{\text{closed}}:(g, 0, c).Gate \; + \\ & \underline{\text{raise}}:(g, r_{gt}, c).Gate + \underline{\text{open}}:(g, 0, c).Gate \; + \\ & \underline{\text{init}}:(g, 0, c).Gate \end{aligned}$$

- two timers

$$\begin{aligned} Timer_L \stackrel{def}{=} \; & \underline{\text{appr}}:(t_L, 1, const).Timer_L + \underline{\text{lower}}:(t_L, 0, const).Timer_L + \\ & \underline{\text{init}}:(t_L, 0, const).Timer_L \\ Timer_R \stackrel{def}{=} \; & \underline{\text{exit}}:(t_R, 1, const).Timer_R + \underline{\text{raise}}:(t_R, 0, const).Timer_R + \\ & \underline{\text{init}}:(t_R, 0, const).Timer_R \end{aligned}$$

HYPE syntax (continued)

- components: $P ::= C_s(\mathcal{W}) \mid C(\mathcal{W}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$

HYPE syntax (continued)

- components: $P ::= C_s(\mathcal{W}) \mid C(\mathcal{W}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \qquad L \subseteq \mathcal{E}$

HYPE syntax (continued)

- components: $P ::= C_s(\mathcal{W}) \mid C(\mathcal{W}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \qquad L \subseteq \mathcal{E}$
- controller: $M ::= \underline{\mathrm{a}}.M \mid 0 \mid M + M \qquad \underline{\mathrm{a}} \in \mathcal{E}$

  $Con ::= M \mid Con \underset{L}{\bowtie} Con \qquad L \subseteq \mathcal{E}$

HYPE syntax (continued)

- components: $P ::= C_s(\mathcal{W}) \mid C(\mathcal{W}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \qquad L \subseteq \mathcal{E}$
- controller: $M ::= \underline{\mathrm{a}}.M \mid 0 \mid M + M \qquad \underline{\mathrm{a}} \in \mathcal{E}$
  $Con ::= M \mid Con \underset{L}{\bowtie} Con \qquad L \subseteq \mathcal{E}$
- controlled system: $ConSys ::= \Sigma \underset{L}{\bowtie} \underline{\mathrm{init}}.Con \qquad L \subseteq \mathcal{E}$

HYPE syntax (continued)

- components: $P ::= C_s(\mathcal{W}) \mid C(\mathcal{W}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \qquad L \subseteq \mathcal{E}$
- controller: $M ::= \underline{\mathrm{a}}.M \mid 0 \mid M + M \qquad \underline{\mathrm{a}} \in \mathcal{E}$
  $Con ::= M \mid Con \underset{L}{\bowtie} Con \qquad L \subseteq \mathcal{E}$
- controlled system: $ConSys ::= \Sigma \underset{L}{\bowtie} \underline{\mathrm{init}}.Con \qquad L \subseteq \mathcal{E}$
- HYPE model: $(ConSys, \mathcal{V}, IN, IT, \mathcal{E}, \mathcal{A}, \mathrm{ec}, \mathrm{iv}, EC, ID)$
  - IN influence names, IT influence types
  - $\mathrm{ec} : \mathcal{E} \to EC$, associates events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $\mathrm{iv} : IN \to \mathcal{V}$, associates influence names with variables
  - ID influence descriptions, $[\![I(\vec{X})]\!] = f(\vec{X})$,

Discrete aspects of the train gate system

- sequencing of train events

$$Seq_a \stackrel{def}{=} \underline{\text{appr}}.Seq_p \quad Seq_p \stackrel{def}{=} \underline{\text{pass}}.Seq_e \quad Seq_e \stackrel{def}{=} \underline{\text{exit}}.Seq_a$$

Discrete aspects of the train gate system

- sequencing of train events

$$Seq_a \stackrel{def}{=} \underline{\text{appr}}.Seq_p \quad Seq_p \stackrel{def}{=} \underline{\text{pass}}.Seq_e \quad Seq_e \stackrel{def}{=} \underline{\text{exit}}.Seq_a$$

- gate internal controller

$$\begin{aligned}
GC_o &\stackrel{def}{=} \underline{\text{raise}}.GC_o + \underline{\text{lower}}.GC_l \\
GC_l &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_l + \underline{\text{closed}}.GC_c \\
GC_c &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_c \\
GC_r &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_l + \underline{\text{open}}.GC_o
\end{aligned}$$

Discrete aspects of the train gate system

- sequencing of train events

$$Seq_a \stackrel{def}{=} \underline{\text{appr}}.Seq_p \quad Seq_p \stackrel{def}{=} \underline{\text{pass}}.Seq_e \quad Seq_e \stackrel{def}{=} \underline{\text{exit}}.Seq_a$$

- gate internal controller

$$\begin{aligned} GC_o &\stackrel{def}{=} \underline{\text{raise}}.GC_o + \underline{\text{lower}}.GC_l \\ GC_l &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_l + \underline{\text{closed}}.GC_c \\ GC_c &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_c \\ GC_r &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_l + \underline{\text{open}}.GC_o \end{aligned}$$

- event conditions

$$\mathrm{ec}(\underline{\text{appr}}) = (D = -1000,\ T_L' = 0) \qquad \mathrm{ec}(\underline{\text{pass}}) = (D = 0, \mathit{true})$$

$$\mathrm{ec}(\underline{\text{exit}}) = (D = 100,\ T_R' = 0 \wedge D' = -1500)$$

Discrete aspects of the train gate system

- sequencing of train events

$$Seq_a \stackrel{def}{=} \underline{\text{appr}}.Seq_p \quad Seq_p \stackrel{def}{=} \underline{\text{pass}}.Seq_e \quad Seq_e \stackrel{def}{=} \underline{\text{exit}}.Seq_a$$

- gate internal controller

$$\begin{aligned}
GC_o &\stackrel{def}{=} \underline{\text{raise}}.GC_o + \underline{\text{lower}}.GC_l \\
GC_l &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_l + \underline{\text{closed}}.GC_c \\
GC_c &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_c \\
GC_r &\stackrel{def}{=} \underline{\text{raise}}.GC_r + \underline{\text{lower}}.GC_l + \underline{\text{open}}.GC_o
\end{aligned}$$

- event conditions

$$\begin{aligned}
\mathrm{ec}(\underline{\text{appr}}) &= (D = -1000, T'_L = 0) \qquad & \mathrm{ec}(\underline{\text{pass}}) &= (D = 0, \mathit{true}) \\
\mathrm{ec}(\underline{\text{exit}}) &= (D = 100, T'_R = 0 \wedge D' = -1500) & &
\end{aligned}$$

$$\begin{aligned}
\mathrm{ec}(\underline{\text{lower}}) &= (T_L = 5, T'_L = 0) \qquad & \mathrm{ec}(\underline{\text{closed}}) &= (G = 0, \mathit{true}) \\
\mathrm{ec}(\underline{\text{raise}}) &= (T_R = 5, T'_R = 0) & \mathrm{ec}(\underline{\text{open}}) &= (G = 90, \mathit{true})
\end{aligned}$$

Discrete aspects of the train gate system (continued)

- system controller

$$
\begin{aligned}
Con_a &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_a \\
Con_l &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_l + \underline{\text{lower}}.Con_e \\
Con_e &\stackrel{def}{=} \underline{\text{appr}}.Con_e + \underline{\text{exit}}.Con_r \\
Con_r &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r + \underline{\text{raise}}.Con_a
\end{aligned}
$$

Discrete aspects of the train gate system (continued)

- system controller

$$
\begin{aligned}
Con_a &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_a \\
Con_l &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_l + \underline{\text{lower}}.Con_e \\
Con_e &\stackrel{def}{=} \underline{\text{appr}}.Con_e + \underline{\text{exit}}.Con_r \\
Con_r &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r + \underline{\text{raise}}.Con_a
\end{aligned}
$$

- additional items

$$
\begin{array}{lclclcl}
\mathrm{iv}(g) & = & G & \quad & \mathrm{iv}(t_L) & = & T_L \\
\mathrm{iv}(d) & = & D & \quad & \mathrm{iv}(t_R) & = & T_R
\end{array}
$$

Discrete aspects of the train gate system (continued)

- system controller

$$\begin{aligned}
Con_a &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_a \\
Con_l &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_l + \underline{\text{lower}}.Con_e \\
Con_e &\stackrel{def}{=} \underline{\text{appr}}.Con_e + \underline{\text{exit}}.Con_r \\
Con_r &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r + \underline{\text{raise}}.Con_a
\end{aligned}$$

- additional items

$$\begin{array}{lclclclcl}
\mathrm{iv}(g) & = & G & \quad \mathrm{iv}(t_L) & = & T_L & & & \\
\mathrm{iv}(d) & = & D & \quad \mathrm{iv}(t_R) & = & T_R & \quad [\![c]\!] & = & 1
\end{array}$$

Discrete aspects of the train gate system (continued)

- system controller

$$\begin{aligned}
Con_a &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_a \\
Con_l &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_l + \underline{\text{lower}}.Con_e \\
Con_e &\stackrel{def}{=} \underline{\text{appr}}.Con_e + \underline{\text{exit}}.Con_r \\
Con_r &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r + \underline{\text{raise}}.Con_a
\end{aligned}$$

- additional items

$$\begin{array}{rclrclrcl}
\mathrm{iv}(g) & = & G & \mathrm{iv}(t_L) & = & T_L & & & \\
\mathrm{iv}(d) & = & D & \mathrm{iv}(t_R) & = & T_R & [\![c]\!] & = & 1
\end{array}$$

$$\mathrm{ec}(\underline{\text{init}}) = (true, G' = 90 \wedge D' = -1400 \wedge T_L' = 0 \wedge T_R' = 0)$$

Discrete aspects of the train gate system (continued)

- system controller

$$
\begin{aligned}
Con_a &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_a \\
Con_l &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_l + \underline{\text{lower}}.Con_e \\
Con_e &\stackrel{def}{=} \underline{\text{appr}}.Con_e + \underline{\text{exit}}.Con_r \\
Con_r &\stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r + \underline{\text{raise}}.Con_a
\end{aligned}
$$

- additional items

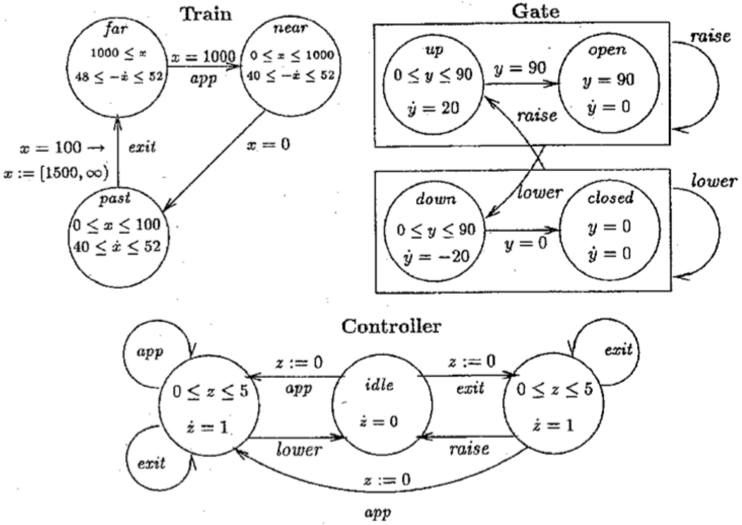
$$
\begin{array}{rclrclrcl}
\mathrm{iv}(g) & = & G & \mathrm{iv}(t_L) & = & T_L & & & \\
\mathrm{iv}(d) & = & D & \mathrm{iv}(t_R) & = & T_R & [\![c]\!] & = & 1
\end{array}
$$

$$
\mathrm{ec}(\underline{\text{init}}) = (true, G' = 90 \wedge D' = -1400 \wedge T_L' = 0 \wedge T_R' = 0)
$$

- controlled system

$$
(Gate \underset{*}{\bowtie} Train \underset{*}{\bowtie} Timer_L \underset{*}{\bowtie} Timer_R) \underset{*}{\bowtie} \underline{\text{init}}.((Con_a \underset{*}{\bowtie} GC_o) \underset{*}{\bowtie} Seq_a)
$$

Train gate system as a hybrid automata product



Operational semantics

- state: $\sigma : \mathit{IN} \rightarrow (\mathbb{R} \times \mathit{IT})$

Operational semantics

- state: $\sigma : \mathit{IN} \rightarrow (\mathbb{R} \times \mathit{IT})$
- configuration: $\langle \mathit{ConSys}, \sigma \rangle$

Operational semantics

- state: $\sigma : \mathit{IN} \to (\mathbb{R} \times \mathit{IT})$
- configuration: $\langle \mathit{ConSys}, \sigma \rangle$
- labelled transition system: $(\mathcal{F}, \mathcal{E}, \to \subseteq \mathcal{F} \times \mathcal{E} \times \mathcal{F})$

Operational semantics

- state: $\sigma : IN \to (\mathbb{R} \times IT)$
- configuration: $\left\langle ConSys, \sigma \right\rangle$
- labelled transition system: $(\mathcal{F}, \mathcal{E}, \to \subseteq \mathcal{F} \times \mathcal{E} \times \mathcal{F})$
- 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

- state: $\sigma : \mathit{IN} \to (\mathbb{R} \times \mathit{IT})$
- configuration: $\big\langle \mathit{ConSys}, \sigma \big\rangle$
- labelled transition system: $(\mathcal{F}, \mathcal{E}, \to \subseteq \mathcal{F} \times \mathcal{E} \times \mathcal{F})$
- 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 (cont.)

Prefix with influence:

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

Prefix without influence:

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

Choice:

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

Constant:

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

Operational semantics (cont.)

Prefix with influence:

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

Prefix without influence:

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

Choice:

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

Constant:

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

Operational semantics (cont.)

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 (cont.)

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

Hybrid semantics

- extract ODEs from each state σ for each configuration

$$CS_\sigma = \left\{ \text{ODE for variable } V \;\middle|\; V \in \mathcal{V} \right\} \text{ where}$$

$$\frac{dV}{dt} = \sum \left\{ r[\![\mathcal{W}]\!] \;\middle|\; \mathrm{iv}(\iota) = V \text{ and } \sigma(\iota) = (r, \mathcal{W}) \right\}$$

  - for any influence name associated with V
  - determine from σ its rate and influence type
  - multiply its rate and influence function together
  - sum these over all associated influence names

Hybrid semantics

- extract ODEs from each state σ for each configuration

$$CS_\sigma = \left\{ \text{ODE for variable } V \;\middle|\; V \in \mathcal{V} \right\} \;\text{ where}$$

$$\frac{dV}{dt} = \sum \left\{ r[\![\mathcal{W}]\!] \;\middle|\; \mathrm{iv}(\iota) = V \text{ and } \sigma(\iota) = (r, \mathcal{W}) \right\}$$

  - for any influence name associated with V
  - determine from σ its rate and influence type
  - multiply its rate and influence function together
  - sum these over all associated influence names
- map labelled transition system to hybrid automaton
  - configurations are modes, transitions are edges
  - ODEs are flows at configurations

Train gate system

- configurations have the form

$$\Big\langle \big(Gate \underset{*}{\bowtie} Train \underset{*}{\bowtie} Timer_L \underset{*}{\bowtie} Timer_R\big) \underset{*}{\bowtie} \underline{\mathrm{init}}.\big((Con_x \underset{*}{\bowtie} GC_y) \underset{*}{\bowtie} Seq_z\big), \sigma \Big\rangle$$

Train gate system

- configurations have the form

$$\left\langle (Gate \underset{*}{\bowtie} Train \underset{*}{\bowtie} Timer_L \underset{*}{\bowtie} Timer_R) \underset{*}{\bowtie} \underline{\mathrm{init}}.((Con_x \underset{*}{\bowtie} GC_y) \underset{*}{\bowtie} Seq_z), \sigma \right\rangle$$

- which will be abbreviated to

$$\left\langle CxGySz, \sigma \right\rangle$$

Train gate system

- configurations have the form

$$\left\langle (Gate \underset{*}{\bowtie} Train \underset{*}{\bowtie} Timer_L \underset{*}{\bowtie} Timer_R) \underset{*}{\bowtie} \underline{\text{init}}.((Con_x \underset{*}{\bowtie} GC_y) \underset{*}{\bowtie} Seq_z), \sigma \right\rangle$$

- which will be abbreviated to

$$\left\langle CxGySz, \sigma \right\rangle$$

- first configuration after init

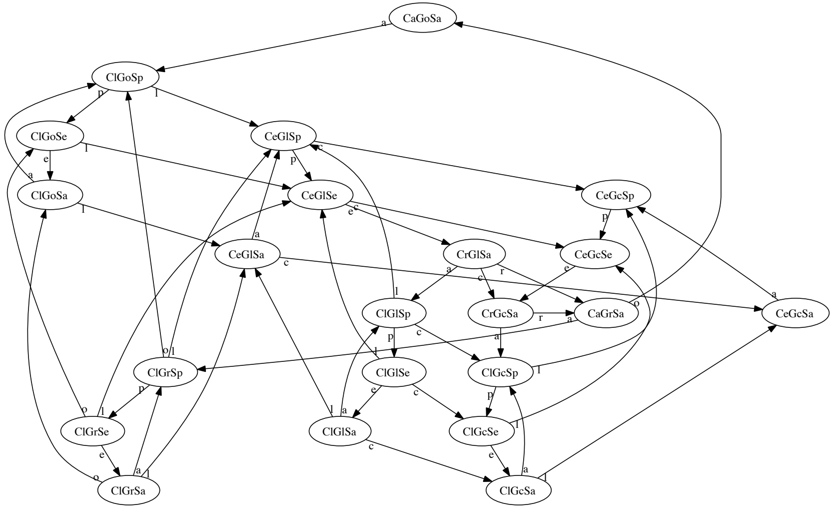
$$\left\langle CaGoSa, \{d \mapsto (r_{tr}, c), g \mapsto (0, c), t_l \mapsto (0, c), t_r \mapsto (0, c)\} \right\rangle$$

Train gate system

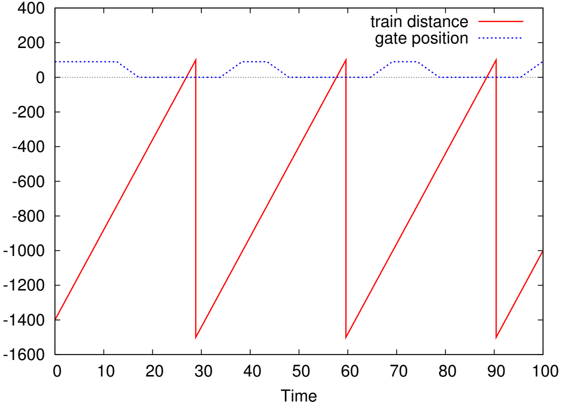
- configurations have the form
$$\Big\langle (Gate \underset{*}{\bowtie} Train \underset{*}{\bowtie} Timer_L \underset{*}{\bowtie} Timer_R) \underset{*}{\bowtie} \underline{\mathrm{init}}.((Con_x \underset{*}{\bowtie} GC_y) \underset{*}{\bowtie} Seq_z), \sigma \Big\rangle$$
- which will be abbreviated to
$$\Big\langle CxGySz, \sigma \Big\rangle$$
- first configuration after <u>init</u>
$$\Big\langle CaGoSa, \{d \mapsto (r_{tr}, c), g \mapsto (0, c), t_l \mapsto (0, c), t_r \mapsto (0, c)\} \Big\rangle$$
- first configuration has the following ODEs

$$\frac{dD}{dt} = r_{tr} \qquad \frac{dG}{dt} = 0 \qquad \frac{dT_L}{dt} = 0 \qquad \frac{dT_R}{dt} = 0$$

Labelled transition system of train gate system



Train gate system



Gate closes when train is 505.99m before crossing

Modification for faster trains

- new trains can travel significantly faster

Modification for faster trains

- new trains can travel significantly faster
- gate speed and behaviour remains unchanged

Modification for faster trains

- new trains can travel significantly faster
- gate speed and behaviour remains unchanged
- trains must slow as they approach gate

Modification for faster trains

- new trains can travel significantly faster
- gate speed and behaviour remains unchanged
- trains must slow as they approach gate
- use first sensor as slowing point, unnecessary to add event

Modification for faster trains

- new trains can travel significantly faster
- gate speed and behaviour remains unchanged
- trains must slow as they approach gate
- use first sensor as slowing point, unnecessary to add event
- add new subcomponent

$$\begin{aligned} TrainRS \quad &\stackrel{def}{=} \quad \underline{\text{appr}}:(d_s, -r_{sl}, c).TrainRS + \underline{\text{exit}}:(d_s, 0, c).TrainRS \; + \\ &\quad\;\; \underline{\text{init}}:(d_s, 0, c).TrainRS \end{aligned}$$

Modification for faster trains

- new trains can travel significantly faster
- gate speed and behaviour remains unchanged
- trains must slow as they approach gate
- use first sensor as slowing point, unnecessary to add event
- add new subcomponent

$$\begin{aligned} \mathit{TrainRS} \stackrel{\mathit{def}}{=}\ & \underline{\mathrm{appr}}:(d_s, -r_{sl}, c).\mathit{TrainRS} + \underline{\mathrm{exit}}:(d_s, 0, c).\mathit{TrainRS}\ + \\ & \underline{\mathrm{init}}:(d_s, 0, c).\mathit{TrainRS} \end{aligned}$$

- associate new influence with distance: $\mathrm{iv}(d_s) = D$

Modification for faster trains

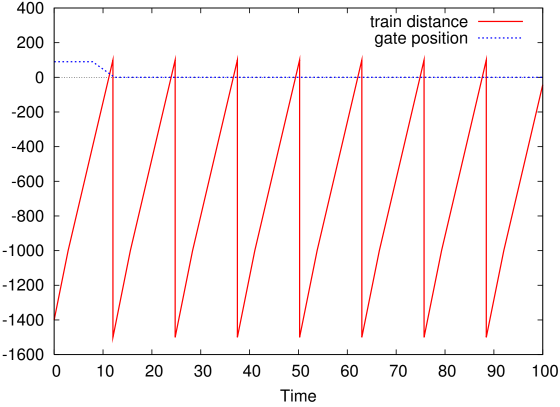
- new trains can travel significantly faster
- gate speed and behaviour remains unchanged
- trains must slow as they approach gate
- use first sensor as slowing point, unnecessary to add event
- add new subcomponent

$$\mathit{TrainRS} \stackrel{\mathit{def}}{=} \underline{\mathrm{appr}}:(d_s, -r_{sl}, c).\mathit{TrainRS} + \underline{\mathrm{exit}}:(d_s, 0, c).\mathit{TrainRS} + \underline{\mathrm{init}}:(d_s, 0, c).\mathit{TrainRS}$$

- associate new influence with distance: $\mathrm{iv}(d_s) = D$
- distance ODE for configurations where slowing is in effect

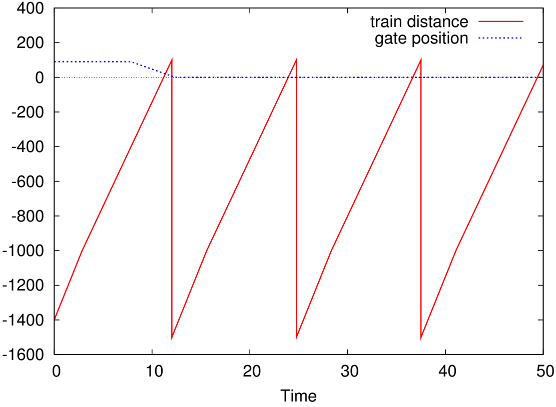
$$\frac{dD}{dt} = r_{tr} - r_{sl}$$

Modified train gate system



parameter modifications: $r_{tr} = 140, \ r_{sl} = 20$

Modified train gate system



parameter modifications: $r_{tr} = 140$, $r_{sl} = 20$

Modification to identify failure

- violation of safety condition

Modification to identify failure

- violation of safety condition
- introduce a new event <u>fail</u>

Modification to identify failure

- violation of safety condition
- introduce a new event fail
- $\mathrm{ec}(\underline{\mathsf{fail}}) = (-100 \leq D \wedge G \neq 0, \mathit{true})$

Modification to identify failure

- violation of safety condition
- introduce a new event fail
- $\mathrm{ec}(\underline{\mathsf{fail}}) = (-100 \leq D \wedge G \neq 0, \mathit{true})$
- requires controller

Modification to identify failure

- violation of safety condition
- introduce a new event $\underline{\text{fail}}$
- $\mathrm{ec}(\underline{\text{fail}}) = (-100 \leq D \wedge G \neq 0, \mathit{true})$
- requires controller
- $FC \stackrel{\mathit{def}}{=} \underline{\text{fail}}.0$

Modification to identify failure

- violation of safety condition
- introduce a new event $\underline{\text{fail}}$
- $\mathrm{ec}(\underline{\text{fail}}) = (-100 \leq D \wedge G \neq 0, \mathit{true})$
- requires controller
- $FC \stackrel{\mathit{def}}{=} \underline{\text{fail}}.0$
- doubles labelled transition system and hybrid automata size

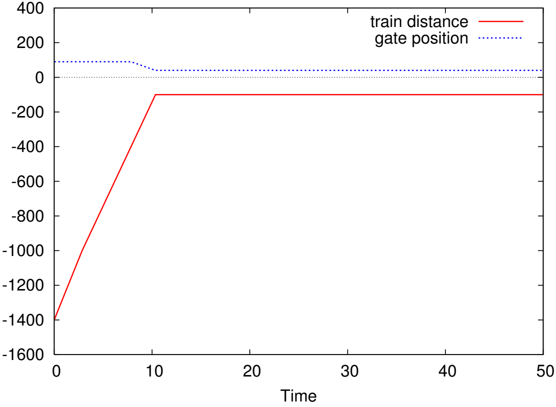
Modification to identify failure

- violation of safety condition
- introduce a new event $\underline{\text{fail}}$
- $\mathrm{ec}(\underline{\text{fail}}) = (-100 \leq D \wedge G \neq 0, \mathit{true})$
- requires controller
- $FC \stackrel{\mathit{def}}{=} \underline{\text{fail}}.0$
- doubles labelled transition system and hybrid automata size
- alternative approach adds only one state
  - add $\underline{\text{fail}}$ to all subcomponents
  - set all influences to zero

Modification to identify failure

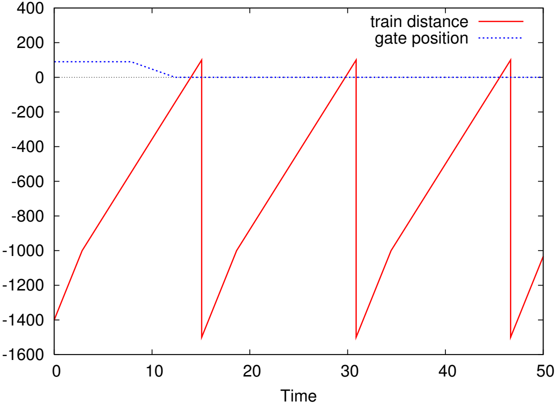
- violation of safety condition
- introduce a new event fail
- $\mathrm{ec}(\underline{\text{fail}}) = (-100 \leq D \wedge G \neq 0, \mathit{true})$
- requires controller
- $FC \stackrel{\mathit{def}}{=} \underline{\text{fail}}.0$
- doubles labelled transition system and hybrid automata size
- alternative approach adds only one state
  - add fail to all subcomponents
  - set all influences to zero
- graph based on alternative approach

Train gate system with failure event



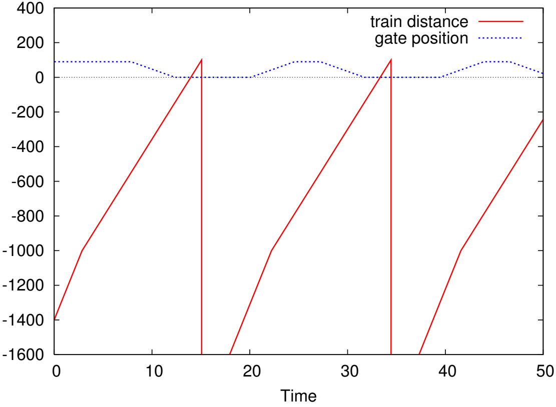
parameter modifications: $r_{tr} = 140$, $r_{sl} = 20$

Train gate system: slower speed



parameter modifications: $r_{tr} = 140$, $r_{sl} = 50$

Train gate system: larger orginating distance



parameter modifications: $r_{tr} = 140$, $r_{sl} = 50$, $D = -2000$

Well-behaved HYPE models

- well-behaved – no instantaneous Zeno behaviour
  - no infinite sequence of simultaneous events

Well-behaved HYPE models

- well-behaved – no instantaneous Zeno behaviour
  - no infinite sequence of simultaneous events
- construct I-graph
  - from controller and event conditions
  - identify overlaps between resets and activation conditions
  - identify whether an event can instantaneously follow another

Well-behaved HYPE models

- well-behaved – no instantaneous Zeno behaviour
  - no infinite sequence of simultaneous events
- construct I-graph
  - from controller and event conditions
  - identify overlaps between resets and activation conditions
  - identify whether an event can instantaneously follow another
- theorem: HYPE model with an acyclic I-graph is well-behaved

Well-behaved HYPE models

- well-behaved – no instantaneous Zeno behaviour
  - no infinite sequence of simultaneous events
- construct I-graph
  - from controller and event conditions
  - identify overlaps between resets and activation conditions
  - identify whether an event can instantaneously follow another
- theorem: HYPE model with an acyclic I-graph is well-behaved
- proposition: two well-behaved controllers whose unshared events do not activate events of the other controller have well-behaved cooperation
  - assuming all shared events appear in the cooperation set

Well-behaved HYPE models

- well-behaved – no instantaneous Zeno behaviour
  - no infinite sequence of simultaneous events
- construct I-graph
  - from controller and event conditions
  - identify overlaps between resets and activation conditions
  - identify whether an event can instantaneously follow another
- theorem: HYPE model with an acyclic I-graph is well-behaved
- proposition: two well-behaved controllers whose unshared events do not activate events of the other controller have well-behaved cooperation
  - assuming all shared events appear in the cooperation set
- apply result to train gate controller

Train gate system: well-behaved?

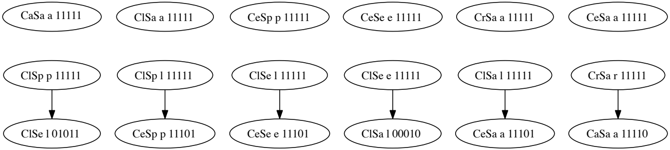
- Con_a is not well-behaved
  - $Con_l \stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r$
  - $\underline{\text{appr}}$ does not inhibit itself
  - $(Con_l, \underline{\text{appr}}, 10001) \rightarrow (Con_l, \underline{\text{appr}}, 10001)$

Train gate system: well-behaved?

- Con_a is not well-behaved
  - $Con_l \stackrel{def}{=} \underline{\text{appr}}.Con_l + \underline{\text{exit}}.Con_r$
  - $\underline{\text{appr}}$ does not inhibit itself
  - $(Con_l, \underline{\text{appr}}, 10001) \rightarrow (Con_l, \underline{\text{appr}}, 10001)$
- I-graph of $Con_a \underset{*}{\bowtie} Seq_a$

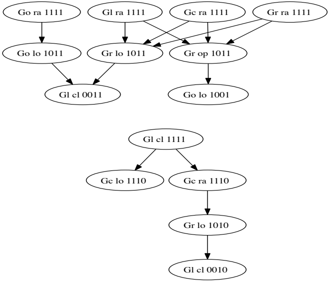
Train gate system: well-behaved?

- Con_a is not well-behaved
  - $Con_l \stackrel{def}{=} \underline{\mathrm{appr}}.Con_l + \underline{\mathrm{exit}}.Con_r$
  - $\underline{\mathrm{appr}}$ does not inhibit itself
  - $(Con_l, \underline{\mathrm{appr}}, 10001) \rightarrow (Con_l, \underline{\mathrm{appr}}, 10001)$
- I-graph of $Con_a \underset{*}{\bowtie} Seq_a$



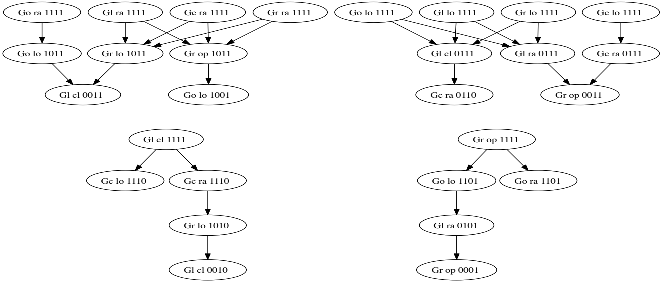
Train gate system: well-behaved

- I-graph of gate controller



Train gate system: well-behaved

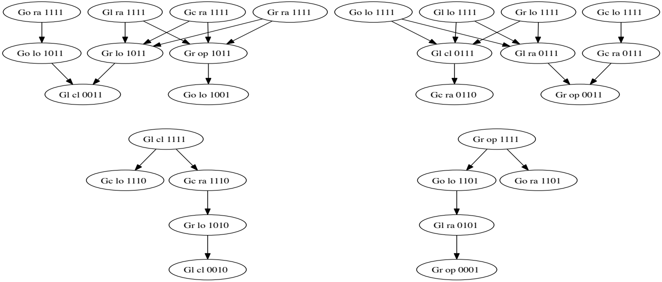
- I-graph of gate controller



- $Con_a \underset{*}{\bowtie} Seq_a$ is well-behaved and GC_o is well-behaved

Train gate system: well-behaved

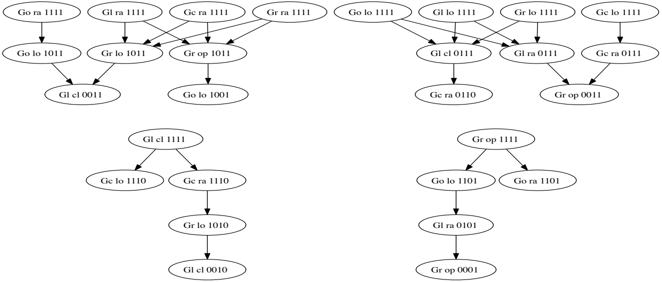
- I-graph of gate controller



- $Con_a \underset{*}{\bowtie} Seq_a$ is well-behaved and GC_o is well-behaved
- no event activates another event

Train gate system: well-behaved

- I-graph of gate controller



- $Con_a \bowtie_{*} Seq_a$ is well-behaved and GC_o is well-behaved
- no event activates another event
- hence $Con_a \bowtie_{*} Seq_a \bowtie_{*} GC_o$ is well-behaved

Conclusion

- HYPE
  - process algebra for hybrid systems, describes flows
  - operational semantics provide ODEs, then map to hybrid automata

Conclusion

- HYPE
  - process algebra for hybrid systems, describes flows
  - operational semantics provide ODEs, then map to hybrid automata
- illustrated through a railway gate system
  - separation of continuous and discrete aspects
  - compositionality to modify behaviour
  - add failure event based on conditions from different subsystems

Conclusion

- HYPE
  - process algebra for hybrid systems, describes flows
  - operational semantics provide ODEs, then map to hybrid automata
- illustrated through a railway gate system
  - separation of continuous and discrete aspects
  - compositionality to modify behaviour
  - add failure event based on conditions from different subsystems
- exclusion of infinite behaviour at a time instant
  - instantaneous Zeno behaviour
  - construct I-graph from controller and event conditions
  - check for acyclicity
  - abstract from continuous behaviour

Thank you

Instantaneous Zeno behaviour

- Zeno behaviour
  - infinite number of events in a finite time period

Instantaneous Zeno behaviour

- Zeno behaviour
  - infinite number of events in a finite time period
- instantaneous Zeno behaviour
  - infinite number of events in a time instant
  - not representative of reality
  - not permitted in piecewise deterministic Markov processes

Instantaneous Zeno behaviour

- Zeno behaviour
  - infinite number of events in a finite time period
- instantaneous Zeno behaviour
  - infinite number of events in a time instant
  - not representative of reality
  - not permitted in piecewise deterministic Markov processes
- how to check for this behaviour
  - without investigating the full behaviour of a model
  - controller determines event ordering
  - event conditions affect possible values
  - combine these and ignore continuous behaviour

Instantaneous Zeno behaviour

- Zeno behaviour
  - infinite number of events in a finite time period
- instantaneous Zeno behaviour
  - infinite number of events in a time instant
  - not representative of reality
  - not permitted in piecewise deterministic Markov processes
- how to check for this behaviour
  - without investigating the full behaviour of a model
  - controller determines event ordering
  - event conditions affect possible values
  - combine these and ignore continuous behaviour
- basic idea: how does the activation condition and reset of one event affect whether another event can occur immediately

Instantaneous activation graph

- nodes: $(\mathit{CeGlcSp}, \underline{\text{closed}}, 1110111)$
  - controller state
  - possible next event
  - vector indicating what events are enabled or inhibited

Instantaneous activation graph

- nodes: $(\mathit{CeGlcSp}, \underline{\text{closed}}, 1110111)$
  - controller state
  - possible next event
  - vector indicating what events are enabled or inhibited
- edges: $(\mathit{CeGlSp}, \underline{\text{closed}}, 1110111) \to (\mathit{CeGcSp}, \underline{\text{pass}}, 1110110)$
  - transition in controller, $\mathit{CeGlSp} \xrightarrow{\underline{\text{closed}}} \mathit{CeGcSp}$
  - current event is enabled, following event is enabled
  - new vector is update of previous vector taking in account what has been enabled and inhibited by current event

Instantaneous activation graph

- nodes: $(CeGlcSp, \underline{\text{closed}}, 1110111)$
  - controller state
  - possible next event
  - vector indicating what events are enabled or inhibited
- edges: $(CeGlSp, \underline{\text{closed}}, 1110111) \rightarrow (CeGcSp, \underline{\text{pass}}, 1110110)$
  - transition in controller, $CeGlSp \xrightarrow{\text{closed}} CeGcSp$
  - current event is enabled, following event is enabled
  - new vector is update of previous vector taking in account what has been enabled and inhibited by current event
- construct graph from every valid controller state and event pair together with vector consisting of ones
  - abstracts from initial conditions