HYPE applied to the modelling of hybrid biological systems

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)

24 May 2008

Introduction

- hybrid systems

Introduction

- hybrid systems
  - discrete behaviour

Introduction

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

Introduction

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

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA
  - fine-grained modelling of flows

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA
  - fine-grained modelling of flows
  - hybrid automata

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA
  - fine-grained modelling of flows
  - hybrid automata
- gene regulatory network

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA
  - fine-grained modelling of flows
  - hybrid automata
- gene regulatory network
  - view genes as on or off

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA
  - fine-grained modelling of flows
  - hybrid automata
- gene regulatory network
  - view genes as on or off
  - single gene in the cell with multiple mRNA and proteins

Introduction

- hybrid systems
  - discrete behaviour
  - continuous behaviour, expressed as ODEs
- HYPE
  - process algebra for hybrid systems
  - inspired by PEPA
  - fine-grained modelling of flows
  - hybrid automata
- gene regulatory network
  - view genes as on or off
  - single gene in the cell with multiple mRNA and proteins
  - the Repressilator [Elowitz and Leibler, 2000]

Biological background

- DNA, genes

Biological background

- DNA, genes
  - transcription of mRNA

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins
- proteins

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins
- proteins
  - various roles, structural, enzymes

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins
- proteins
  - various roles, structural, enzymes
- transcription factors

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins
- proteins
  - various roles, structural, enzymes
- transcription factors
  - inhibitory – repressors

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins
- proteins
  - various roles, structural, enzymes
- transcription factors
  - inhibitory – repressors
  - promoting – activators

Biological background

- DNA, genes
  - transcription of mRNA
  - translation to proteins
- proteins
  - various roles, structural, enzymes
- transcription factors
  - inhibitory – repressors
  - promoting – activators
  - feedback loops

The Repressilator

- synthetic network

The Repressilator

- synthetic network
- three genes with three inhibitors

The Repressilator

- synthetic network
- three genes with three inhibitors
- reporter of green flourescent protein (GFP)

The Repressilator

- synthetic network
- three genes with three inhibitors
- reporter of green flourescent protein (GFP)
- negative feedback cycle

The Repressilator

- synthetic network
- three genes with three inhibitors
- reporter of green flourescent protein (GFP)
- negative feedback cycle
  - each gene produces a protein

The Repressilator

- synthetic network
- three genes with three inhibitors
- reporter of green flourescent protein (GFP)
- negative feedback cycle
  - each gene produces a protein
  - protein inhibits transcription of mRNA by another gene

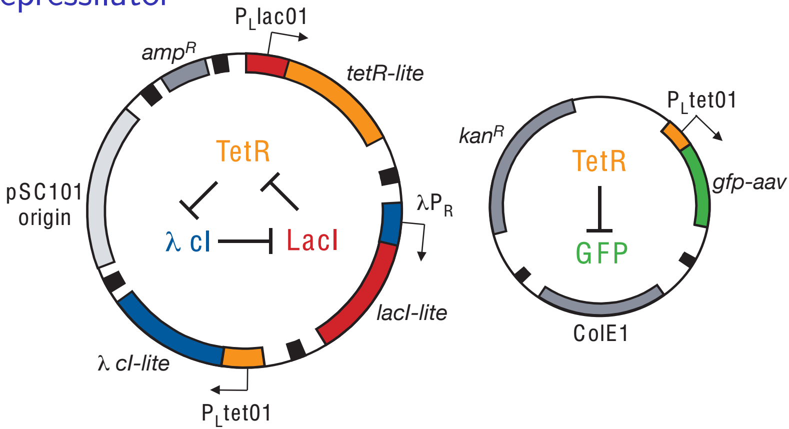
The Repressilator

- synthetic network
- three genes with three inhibitors
- reporter of green flourescent protein (GFP)
- negative feedback cycle
  - each gene produces a protein
  - protein inhibits transcription of mRNA by another gene
  - other gene cannot produce its protein

The Repressilator

- synthetic network
- three genes with three inhibitors
- reporter of green flourescent protein (GFP)
- negative feedback cycle
  - each gene produces a protein
  - protein inhibits transcription of mRNA by another gene
  - other gene cannot produce its protein
- quantities of proteins oscillate over time

The Repressilator



M.B. Elowitz and S. Leibler, A synthetic oscillatory network of transcriptional regulators, *Nature*, 403, 335-338.

The Repressilator

$Gene_A$

The Repressilator

$$Gene_A \xrightarrow[trs_A]{} mRNA_A$$

The Repressilator

$$Gene_A \xrightarrow[trs_A]{} mRNA_A$$
$$\downarrow dm_A$$

The Repressilator

$$Gene_A \underset{trs_A}{\longrightarrow} mRNA_A \underset{trl_A}{\longrightarrow} Pr_A$$
$$\downarrow dm_A$$

The Repressilator

$$Gene_A \underset{trs_A}{\longrightarrow} mRNA_A \underset{trl_A}{\longrightarrow} Pr_A$$
$$\downarrow dm_A \qquad \downarrow dp_A$$

The Repressilator

$$Gene_A \xrightarrow[trs_A]{} mRNA_A \xrightarrow[trl_A]{} Pr_A$$

$$\downarrow dm_A \qquad \downarrow dp_A$$

$$Gene_B \xrightarrow[trs_B]{} mRNA_B \xrightarrow[trl_B]{} Pr_B$$

$$\downarrow dm_B \qquad \downarrow dp_B$$

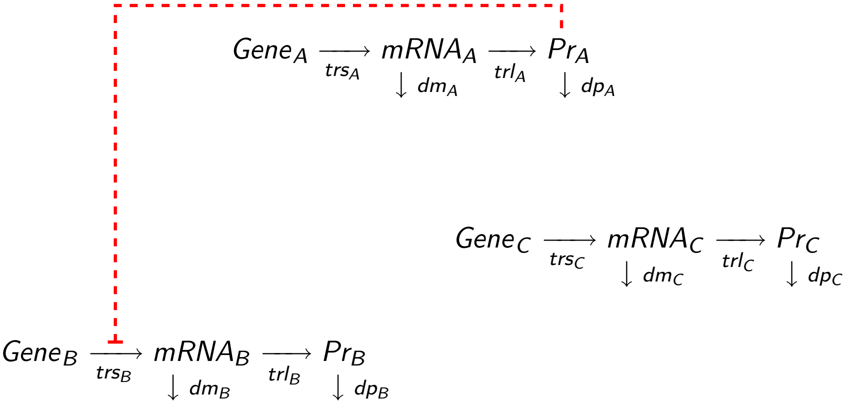
The Repressilator

$$Gene_A \xrightarrow[trs_A]{} mRNA_A \xrightarrow[trl_A]{} Pr_A$$
$$\downarrow dm_A \qquad \downarrow dp_A$$

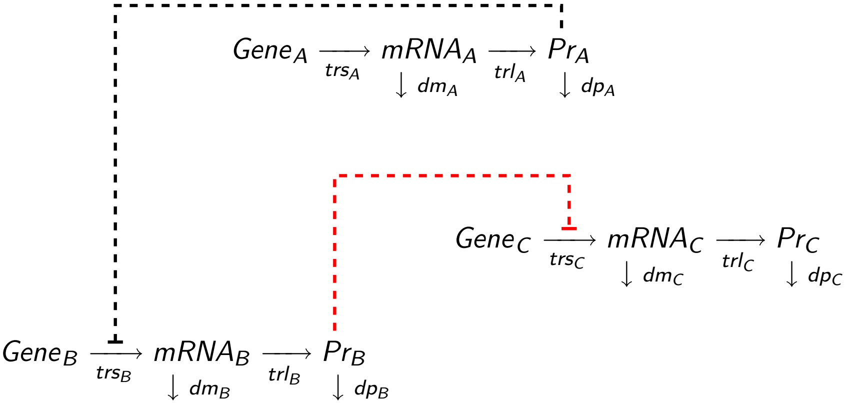
$$Gene_C \xrightarrow[trs_C]{} mRNA_C \xrightarrow[trl_C]{} Pr_C$$
$$\downarrow dm_C \qquad \downarrow dp_C$$

$$Gene_B \xrightarrow[trs_B]{} mRNA_B \xrightarrow[trl_B]{} Pr_B$$
$$\downarrow dm_B \qquad \downarrow dp_B$$

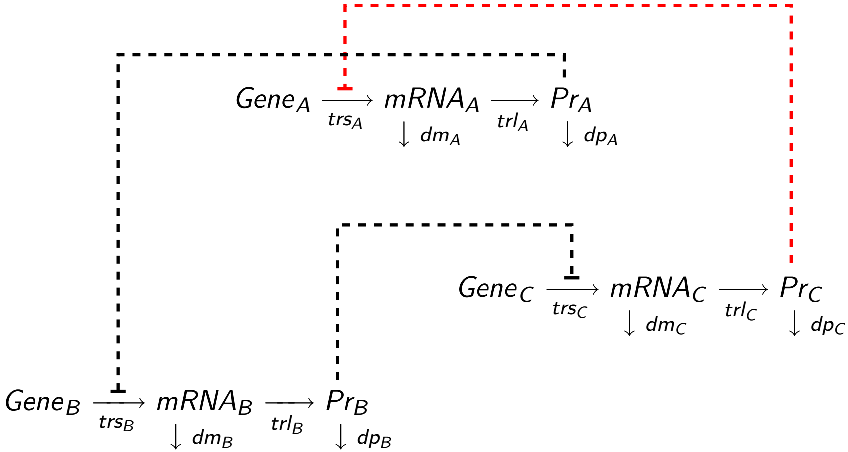
The Repressilator



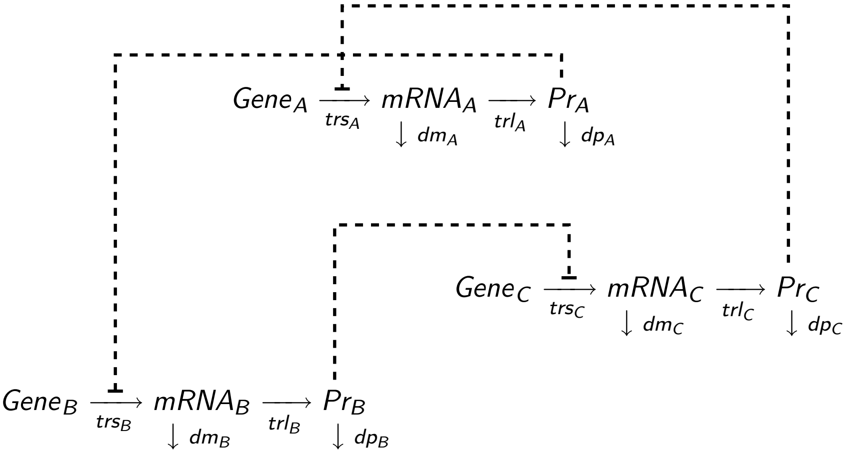
The Repressilator



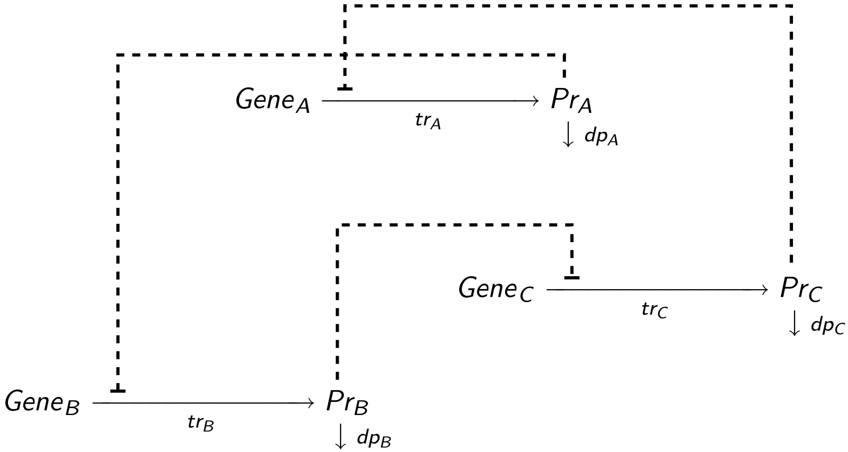
The Repressilator



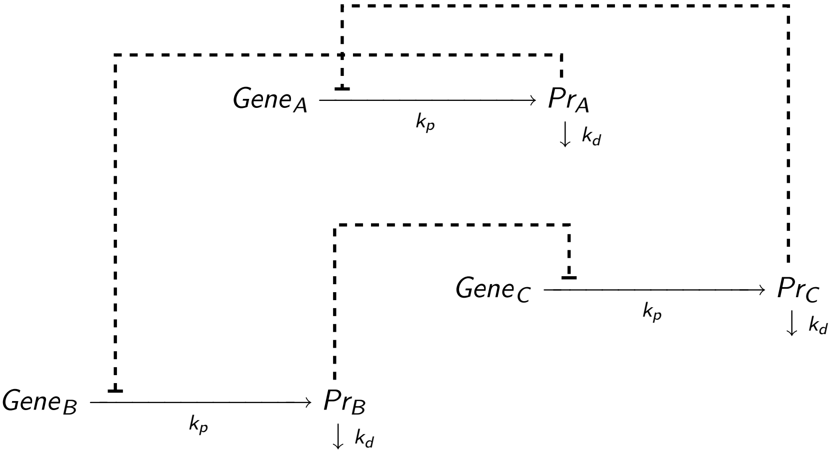
The Repressilator



The Repressilator



The Repressilator



HYPE syntax

HYPE syntax

- two types of action

HYPE syntax

- two types of action
- events — instantaneous, discrete changes

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

HYPE syntax

- two types of action
- events — instantaneous, discrete changes

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

- activities — influences on continuous aspects, flows

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

influence name (ι), rate (r), influence type ($I(\vec{X})$) with $[\![I(\vec{X})]\!] = f(\vec{X})$

where $\vec{X}$ is a formal parameter.

HYPE syntax

- two types of action
- events — instantaneous, discrete changes

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

- activities — influences on continuous aspects, flows

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

influence name (ι) — rate (r) — influence type ($I(\vec{X})$) with $[\![I(\vec{X})]\!] = f(\vec{X})$

where $\vec{X}$ is a formal parameter.

HYPE syntax

- two types of action
- events — instantaneous, discrete changes

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

- activities — influences on continuous aspects, flows

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

influence name (ι), rate (r), influence type ($I(\vec{X})$) with $[\![I(\vec{X})]\!] = f(\vec{X})$

where $\vec{X}$ is a formal parameter.

HYPE syntax

- two types of action
- events — instantaneous, discrete changes

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

- activities — influences on continuous aspects, flows

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

influence name (ι), rate (r), influence type ($I(\vec{X})$) with $[\![I(\vec{X})]\!] = f(\vec{X})$

where $\vec{X}$ is a formal parameter.

HYPE syntax

- two types of action
- events — instantaneous, discrete changes

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

- activities — influences on continuous aspects, flows

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

ι: influence name; r: rate; $I(\vec{X})$: influence type with $[\![I(\vec{X})]\!] = f(\vec{X})$

where $\vec{X}$ is a formal parameter.

- not stochastic

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\mathrm{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\mathrm{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\mathrm{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\mathrm{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

- composed

$$Gene_A(X) \stackrel{def}{=} (G_A^{degr}(X) \underset{\{\underline{\mathrm{init}}\}}{\bowtie} G_A^{prod})$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

- composed

$$Gene_A(X) \stackrel{def}{=} (G_A^{degr}(X) \underset{\{\underline{\text{init}}\}}{\bowtie} G_A^{prod})$$

The Repressilator in HYPE

- degradation

$$G_A^{degr}(X) \stackrel{def}{=} \underline{\text{init}} : (d_A, -k_d, linear(X)).G_A^{degr}(X)$$

- production

$$\begin{aligned} G_A^{prod} &\stackrel{def}{=} \underline{\text{init}} : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{express}}_A : (p_A, k_p, const).G_A^{prod} \\ &+ \underline{\text{inhibit}}_A : (p_A, 0, const).G_A^{prod} \end{aligned}$$

- composed

$$Gene_A(X) \stackrel{def}{=} (G_A^{degr}(X) \underset{\{\underline{\text{init}}\}}{\bowtie} G_A^{prod})$$

- "controller"

$$Con_A \stackrel{def}{=} \underline{\text{inhibit}}_A.\underline{\text{express}}_A.Con_A$$

The Repressilator in HYPE (cont.)

$$Rep \stackrel{def}{=} (Gene_A(A_B) \underset{\underline{\text{init}}}{\bowtie} Gene_B(B_C) \underset{\underline{\text{init}}}{\bowtie} Gene_C(C_A)) \underset{L}{\bowtie} \underline{\text{init}}.(Con_A \parallel Con_B \parallel Con_C)$$

$$L = \{\underline{\text{init}}, \underline{\text{inhibit}}_A, \underline{\text{inhibit}}_B, \underline{\text{inhibit}}_C, \underline{\text{express}}_A, \underline{\text{express}}_B, \underline{\text{express}}_C\}$$

The Repressilator in HYPE (cont.)

$$Rep \stackrel{def}{=} (Gene_A(A_B) \underset{\underline{\mathrm{init}}}{\bowtie} Gene_B(B_C) \underset{\underline{\mathrm{init}}}{\bowtie} Gene_C(C_A)) \underset{L}{\bowtie} \underline{\mathrm{init}}.(Con_A \parallel Con_B \parallel Con_C)$$

$$L = \{\underline{\mathrm{init}}, \underline{\mathrm{inhibit}}_A, \underline{\mathrm{inhibit}}_B, \underline{\mathrm{inhibit}}_C, \underline{\mathrm{express}}_A, \underline{\mathrm{express}}_B, \underline{\mathrm{express}}_C\}$$

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

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

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

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

$$iv(d_A) = A_B \qquad iv(p_A) = A_B$$

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

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

$$iv(d_A) = A_B \qquad iv(p_A) = A_B$$
$$iv(d_B) = B_C \qquad iv(p_B) = B_C$$
$$iv(d_C) = C_A \qquad iv(p_C) = C_A$$

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

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

$$\begin{aligned} iv(d_A) &= A_B & \qquad iv(p_A) &= A_B \\ iv(d_B) &= B_C & \qquad iv(p_B) &= B_C \\ iv(d_C) &= C_A & \qquad iv(p_C) &= C_A \end{aligned}$$

$$ec(\underline{\text{init}}) = (true, (A_B = c_A \wedge B_C = c_B \wedge C_A = c_C))$$

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

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

$$iv(d_A) = A_B \qquad iv(p_A) = A_B$$
$$iv(d_B) = B_C \qquad iv(p_B) = B_C$$
$$iv(d_C) = C_A \qquad iv(p_C) = C_A$$

$$ec(\underline{\text{init}}) = (true, (A_B = c_A \wedge B_C = c_B \wedge C_A = c_C))$$
$$ec(\underline{\text{inhibit}}_A) = (C_A > p, T) \qquad ec(\underline{\text{express}}_A) = (C_A \leq p, T)$$

The Repressilator in HYPE (cont.)

$$\mathcal{V} = \{A_B, B_C, C_A\}$$

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

$$\begin{aligned} iv(d_A) &= A_B & \quad iv(p_A) &= A_B \\ iv(d_B) &= B_C & \quad iv(p_B) &= B_C \\ iv(d_C) &= C_A & \quad iv(p_C) &= C_A \end{aligned}$$

$$ec(\underline{\text{init}}) = (true, (A_B = c_A \wedge B_C = c_B \wedge C_A = c_C))$$

$$\begin{aligned} ec(\underline{\text{inhibit}}_A) &= (C_A > p, T) & \quad ec(\underline{\text{express}}_A) &= (C_A \leq p, T) \\ ec(\underline{\text{inhibit}}_B) &= (A_B > p, T) & \quad ec(\underline{\text{express}}_B) &= (A_B \leq p, T) \\ ec(\underline{\text{inhibit}}_C) &= (B_C > p, T) & \quad ec(\underline{\text{express}}_C) &= (B_C \leq p, T) \end{aligned}$$

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

Operational semantics

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

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

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:

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

Constant:

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

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

The Repressilator – transition derivation

$$\dfrac{\dfrac{\langle G_A^{degr}(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B), \tau_1 \rangle \qquad \langle G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{prod}, \tau_2 \rangle}{\langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau_3 \rangle}}{\langle Gene_A(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle Gene_A(A_B), \tau_3 \rangle}$$

$$\begin{aligned}
\tau &= \{ d_A \mapsto *,\ p_A \mapsto *, \ldots \} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{ d_A \mapsto (-1, I(A_B)),\ p_A \mapsto *, \ldots \} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{ d_A \mapsto *,\ p_A \mapsto (k_p, c), \ldots \} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{ d_A \mapsto (-1, I(A_B)),\ p_A \mapsto (k_p, c), \ldots \}
\end{aligned}$$

The Repressilator – transition derivation

$$\frac{\dfrac{\langle G_A^{degr}(A_B), \tau\rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B), \tau_1\rangle \qquad \langle G_A^{prod}, \tau\rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{prod}, \tau_2\rangle}{\langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau\rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau_3\rangle}}{\langle Gene_A(A_B), \tau\rangle \xrightarrow{\underline{\text{init}}} \langle Gene_A(A_B), \tau_3\rangle}$$

$$\tau = \{d_A \mapsto *,\ p_A \mapsto *, \ldots\}$$

$$\tau_1 = \tau[d_A \mapsto (-1, I(A_B))] = \{d_A \mapsto (-1, I(A_B)),\ p_A \mapsto *, \ldots\}$$

$$\tau_2 = \tau[p_A \mapsto (k_p, c)] = \{d_A \mapsto *,\ p_A \mapsto (k_p, c), \ldots\}$$

$$\tau_3 = \Gamma(\tau, \tau_1, \tau_2) = \{d_A \mapsto (-1, I(A_B)),\ p_A \mapsto (k_p, c), \ldots\}$$

The Repressilator – transition derivation

$$\dfrac{\dfrac{\langle G_A^{degr}(A_B), \tau\rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B), \tau_1\rangle \qquad \langle G_A^{prod}, \tau\rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{prod}, \tau_2\rangle}{\langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau\rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau_3\rangle}}{\langle Gene_A(A_B), \tau\rangle \xrightarrow{\underline{\text{init}}} \langle Gene_A(A_B), \tau_3\rangle}$$

$$\begin{aligned}
\tau &= \{d_A \mapsto *,\ p_A \mapsto *, \ldots\} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{d_A \mapsto (-1, I(A_B)),\ p_A \mapsto *, \ldots\} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{d_A \mapsto *,\ p_A \mapsto (k_p, c), \ldots\} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{d_A \mapsto (-1, I(A_B)),\ p_A \mapsto (k_p, c), \ldots\}
\end{aligned}$$

The Repressilator – transition derivation

$$\dfrac{\dfrac{\langle G_A^{degr}(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B), \tau_1 \rangle \qquad \langle G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{prod}, \tau_2 \rangle}{\langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau_3 \rangle}}{\langle Gene_A(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle Gene_A(A_B), \tau_3 \rangle}$$

$$\begin{aligned}
\tau &= \{ d_A \mapsto *, \ p_A \mapsto *, \ldots \} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{ d_A \mapsto (-1, I(A_B)), \ p_A \mapsto *, \ldots \} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{ d_A \mapsto *, \ p_A \mapsto (k_p, c), \ldots \} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{ d_A \mapsto (-1, I(A_B)), \ p_A \mapsto (k_p, c), \ldots \}
\end{aligned}$$

The Repressilator – transition derivation

$$\frac{\dfrac{\langle G_A^{degr}(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B), \tau_1 \rangle \qquad \langle G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{prod}, \tau_2 \rangle}{\langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B) \underset{\underline{\text{init}}}{\bowtie} G_A^{prod}, \tau_3 \rangle}}{\langle Gene_A(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle Gene_A(A_B), \tau_3 \rangle}$$

$$\begin{aligned}
\tau &= \{ d_A \mapsto *,\ p_A \mapsto *, \ldots \} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{ d_A \mapsto (-1, I(A_B)),\ p_A \mapsto *, \ldots \} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{ d_A \mapsto *,\ p_A \mapsto (k_p, c), \ldots \} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{ d_A \mapsto (-1, I(A_B)),\ p_A \mapsto (k_p, c), \ldots \}
\end{aligned}$$

The Repressilator – transition derivation

$$\frac{\dfrac{\langle G_A^{degr}(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B), \tau_1 \rangle \qquad \langle G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{prod}, \tau_2 \rangle}{\langle G_A^{degr}(A_B) \bowtie_{\underline{\text{init}}} G_A^{prod}, \tau \rangle \xrightarrow{\underline{\text{init}}} \langle G_A^{degr}(A_B) \bowtie_{\underline{\text{init}}} G_A^{prod}, \tau_3 \rangle}}{\langle Gene_A(A_B), \tau \rangle \xrightarrow{\underline{\text{init}}} \langle Gene_A(A_B), \tau_3 \rangle}$$

$$\begin{aligned}
\tau &= \{ d_A \mapsto *, \; p_A \mapsto *, \ldots \} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{ d_A \mapsto (-1, I(A_B)), \; p_A \mapsto *, \ldots \} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{ d_A \mapsto *, \; p_A \mapsto (k_p, c), \ldots \} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{ d_A \mapsto (-1, I(A_B)), \; p_A \mapsto (k_p, c), \ldots \}
\end{aligned}$$

The Repressilator – transition derivation

$$\dfrac{\dfrac{\langle G_A^{degr}(A_B), \tau \rangle \xrightarrow{\text{init}} \langle G_A^{degr}(A_B), \tau_1 \rangle \qquad \langle G_A^{prod}, \tau \rangle \xrightarrow{\text{init}} \langle G_A^{prod}, \tau_2 \rangle}{\langle G_A^{degr}(A_B) \underset{\text{init}}{\bowtie} G_A^{prod}, \tau \rangle \xrightarrow{\text{init}} \langle G_A^{degr}(A_B) \underset{\text{init}}{\bowtie} G_A^{prod}, \tau_3 \rangle}}{\langle Gene_A(A_B), \tau \rangle \xrightarrow{\text{init}} \langle Gene_A(A_B), \tau_3 \rangle}$$

$$\begin{aligned}
\tau &= \{ d_A \mapsto *,\ p_A \mapsto *, \ldots \} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{ d_A \mapsto (-1, I(A_B)),\ p_A \mapsto *, \ldots \} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{ d_A \mapsto *,\ p_A \mapsto (k_p, c), \ldots \} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{ d_A \mapsto (-1, I(A_B)),\ p_A \mapsto (k_p, c), \ldots \}
\end{aligned}$$

The Repressilator – transition derivation

$$\dfrac{\dfrac{\langle G_A^{degr}(A_B), \tau\rangle \xrightarrow{\text{init}} \langle G_A^{degr}(A_B), \tau_1\rangle \qquad \langle G_A^{prod}, \tau\rangle \xrightarrow{\text{init}} \langle G_A^{prod}, \tau_2\rangle}{\langle G_A^{degr}(A_B) \underset{\text{init}}{\bowtie} G_A^{prod}, \tau\rangle \xrightarrow{\text{init}} \langle G_A^{degr}(A_B) \underset{\text{init}}{\bowtie} G_A^{prod}, \tau_3\rangle}}{\langle Gene_A(A_B), \tau\rangle \xrightarrow{\text{init}} \langle Gene_A(A_B), \tau_3\rangle}$$

$$\begin{aligned}
\tau &= \{d_A \mapsto *,\ p_A \mapsto *, \ldots\} \\
\tau_1 &= \tau[d_A \mapsto (-1, I(A_B))] = \{d_A \mapsto (-1, I(A_B)),\ p_A \mapsto *, \ldots\} \\
\tau_2 &= \tau[p_A \mapsto (k_p, c)] = \{d_A \mapsto *,\ p_A \mapsto (k_p, c), \ldots\} \\
\tau_3 &= \Gamma(\tau, \tau_1, \tau_2) = \{d_A \mapsto (-1, I(A_B)),\ p_A \mapsto (k_p, c), \ldots\}
\end{aligned}$$

The Repressilator – labelled transition system

8 configurations represent all possibilities with 8 distinct states

$$
\begin{aligned}
\sigma_0 &= D \cup \{p_A \mapsto (0, c),\ p_B \mapsto (0, c),\ p_C \rightarrow (0, c)\} \\
\sigma_1 &= D \cup \{p_A \mapsto (0, c),\ p_B \mapsto (0, c),\ p_C \mapsto (k_p, c)\} \\
\sigma_2 &= D \cup \{p_A \mapsto (0, c),\ p_B \mapsto (k_p, c),\ p_C \mapsto (0, c)\} \\
\sigma_3 &= D \cup \{p_A \mapsto (0, c),\ p_B \mapsto (k_p, c),\ p_C \mapsto (k_p, c)\} \\
\sigma_4 &= D \cup \{p_A \mapsto (k_p, c),\ p_B \mapsto (0, c),\ p_C \mapsto (0, c)\} \\
\sigma_5 &= D \cup \{p_A \mapsto (k_p, c),\ p_B \mapsto (0, c),\ p_C \mapsto (k_p, c)\} \\
\sigma_6 &= D \cup \{p_A \mapsto (k_p, c),\ p_B \mapsto (k_p, c),\ p_C \mapsto (0, c)\} \\
\sigma_7 &= D \cup \{p_A \mapsto (k_p, c),\ p_B \mapsto (k_p, c),\ p_C \mapsto (k_p, c)\}
\end{aligned}
$$

where

$$D = \{d_A \mapsto (-1, I(A_B)), d_B \mapsto (-1, I(B_C)), d_C \mapsto (-1, I(C_A))\}$$

Hybrid semantics

- extract ODEs from each state σ in the lts of CS

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

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

Hybrid semantics

- extract ODEs from each state σ in the lts of CS

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

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

  - for any influence name associated with V

Hybrid semantics

- extract ODEs from each state σ in the lts of CS

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

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

  - for any influence name associated with V
  - determine from σ its rate and influence type

Hybrid semantics

- extract ODEs from each state σ in the lts of CS

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

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

  - for any influence name associated with V
  - determine from σ its rate and influence type
  - multiply its rate and influence function together

Hybrid semantics

- extract ODEs from each state σ in the lts of CS

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

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

  - 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

The Repressilator – ODEs

$$\begin{aligned}\sigma_4 \;=\; \{ & d_A \mapsto (-1, linear(A_B)),\ p_A \mapsto (k_p, const), \\ & d_B \mapsto (-1, linear(B_C)),\ p_B \mapsto (0, const), \\ & d_C \mapsto (-1, linear(C_A)),\ p_C \mapsto (0, const)\}\end{aligned}$$

The ODEs for this state are

$$\frac{dA_B}{dt} = -k_d A_B + k_p, \quad \frac{dB_C}{dt} = -k_d B_C, \quad \frac{dC_A}{dt} = -k_d C_A$$

The Repressilator – ODEs

$$\begin{aligned}\sigma_4 \;=\; & \{d_A \mapsto (-1, \mathit{linear}(A_B)),\; p_A \mapsto (k_p, \mathit{const}),\\ & d_B \mapsto (-1, \mathit{linear}(B_C)),\; p_B \mapsto (0, \mathit{const}),\\ & d_C \mapsto (-1, \mathit{linear}(C_A)),\; p_C \mapsto (0, \mathit{const})\}\end{aligned}$$

The ODEs for this state are

$$\frac{dA_B}{dt} = -k_d A_B + k_p, \quad \frac{dB_C}{dt} = -k_d B_C, \quad \frac{dC_A}{dt} = -k_d C_A$$

Use ABC for this set of ODEs

The Repressilator – ODEs

$$\begin{aligned}\sigma_4 \;=\; \{& d_A \mapsto (-1, \mathit{linear}(A_B)),\ p_A \mapsto (k_p, \mathit{const}),\\ & d_B \mapsto (-1, \mathit{linear}(B_C)),\ p_B \mapsto (0, \mathit{const}),\\ & d_C \mapsto (-1, \mathit{linear}(C_A)),\ p_C \mapsto (0, \mathit{const})\}\end{aligned}$$

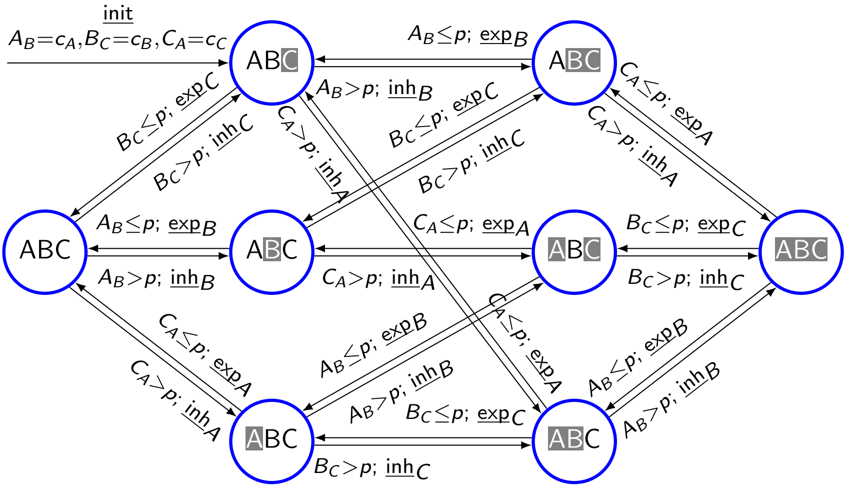
The ODEs for this state are

$$\frac{dA_B}{dt} = -k_d A_B + k_p, \quad \frac{dB_C}{dt} = -k_d B_C, \quad \frac{dC_A}{dt} = -k_d C_A$$

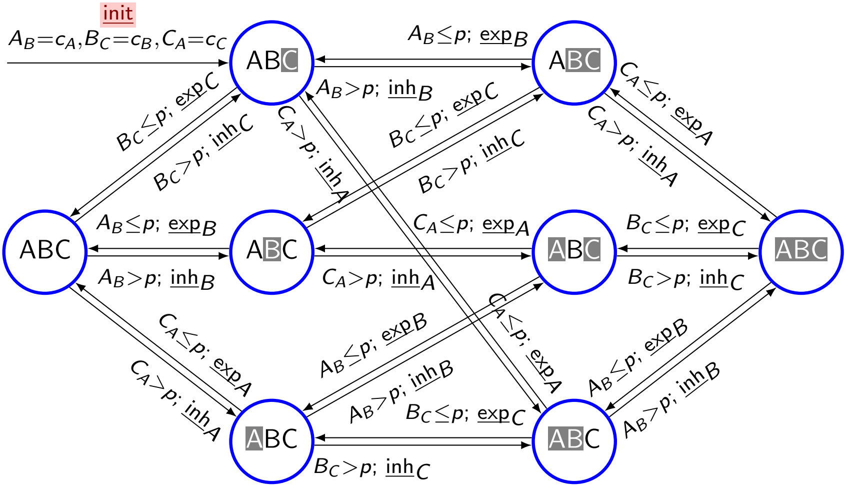
Use ABC for this set of ODEs

Likewise the ODEs for σ_3 would be ABC and for σ_7, ABC

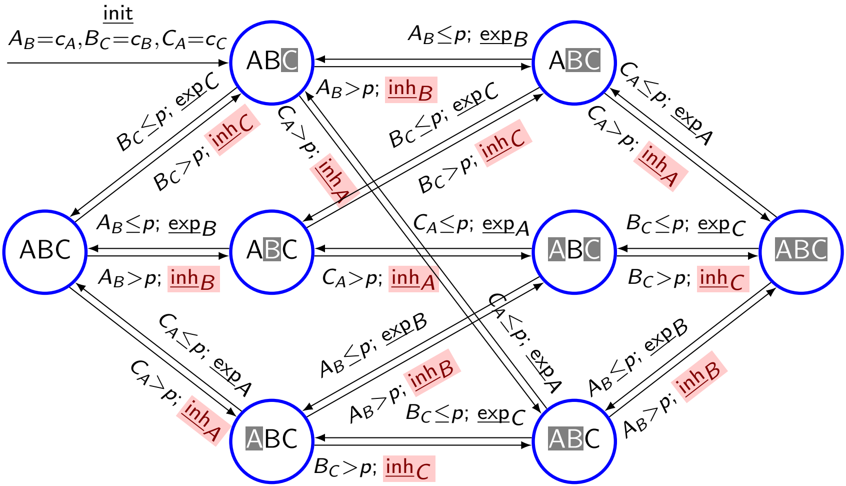
The Repressilator – hybrid automaton



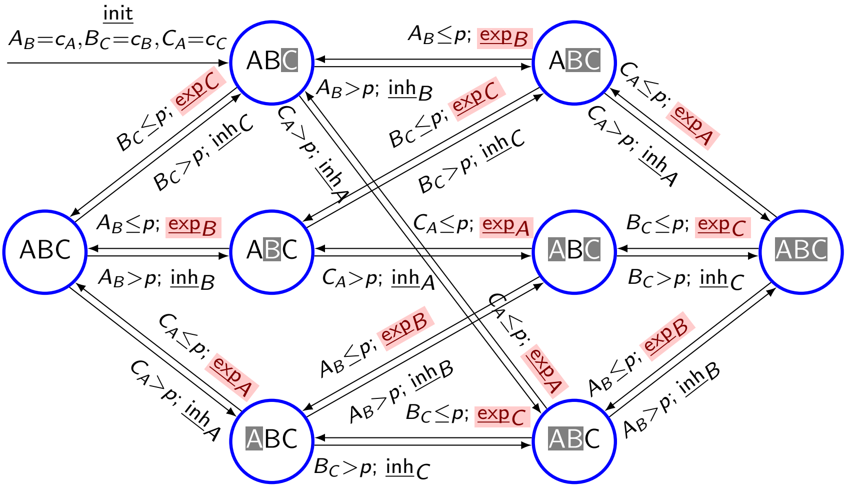
The Repressilator – hybrid automaton



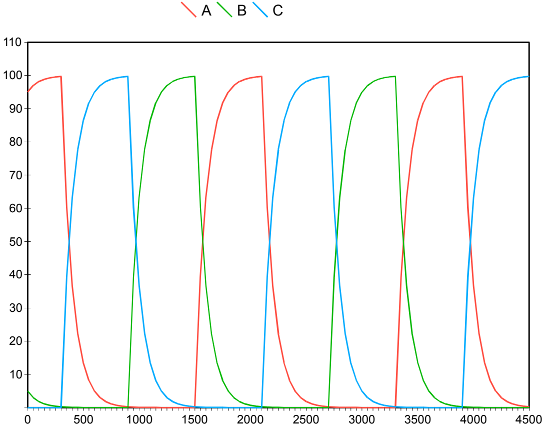
The Repressilator – hybrid automaton



The Repressilator – hybrid automaton



The Repressilator – protein levels over time



k_p=1.00 k_d=0.01

The Repressilator in ACP^{srt}_{hs}

- different style of modelling
- explicit ODEs

$$\begin{aligned}
\text{GeneA} &\stackrel{def}{=} (dA_B = c_A) \curvearrowright \text{OnA} \\
\text{OnA} &\stackrel{def}{=} (\,dA_B/dt = -k_d A_B + k_p\,) \curvearrowright \sigma^*_{\text{rel}} \\
&\quad ((C_A > p) :\!\rightarrow ((A^\bullet_B = {}^\bullet A_B) \curvearrowright \textit{inhibit} \cdot \text{OffA})) \\
\text{OffA} &\stackrel{def}{=} (\,dA_B/dt = -k_d A_B\,) \curvearrowright \sigma^*_{\text{rel}} \\
&\quad ((C_A \leq p) :\!\rightarrow ((A^\bullet_B = {}^\bullet A_B) \curvearrowright \textit{express} \cdot \text{OnA}))
\end{aligned}$$

$$\text{Rep} \stackrel{def}{=} \text{GeneA} \parallel \text{GeneB} \parallel \text{GeneC}$$

The Repressilator in ACP^{srt}_{hs}

- different style of modelling
- explicit ODEs

$$\mathrm{GeneA} \stackrel{def}{=} (dA_B = c_A) \curlywedge \mathrm{OnA}$$

$$\mathrm{OnA} \stackrel{def}{=} (dA_B/dt = -k_d A_B + k_p) \curvearrowright \sigma^*_{\mathrm{rel}}$$
$$((C_A > p) :\rightarrow ((A^\bullet_B = {}^\bullet A_B) \curvearrowright \mathit{inhibit} \cdot \mathrm{OffA}))$$

$$\mathrm{OffA} \stackrel{def}{=} (dA_B/dt = -k_d A_B) \curvearrowright \sigma^*_{\mathrm{rel}}$$
$$((C_A \leq p) :\rightarrow ((A^\bullet_B = {}^\bullet A_B) \curvearrowright \mathit{express} \cdot \mathrm{OnA}))$$

$$\mathrm{Rep} \stackrel{def}{=} \mathrm{GeneA} \parallel \mathrm{GeneB} \parallel \mathrm{GeneC}$$

Further work and conclusions

- further work

Further work and conclusions

- further work
  - less abstract model including mRNA and “leakiness”

Further work and conclusions

- further work
  - less abstract model including mRNA and "leakiness"
  - more realistic parameters

Further work and conclusions

- further work
  - less abstract model including mRNA and “leakiness”
  - more realistic parameters
  - equivalence between more abstract and more concrete models

Further work and conclusions

- further work
  - less abstract model including mRNA and “leakiness”
  - more realistic parameters
  - equivalence between more abstract and more concrete models
  - more biological modelling

Further work and conclusions

- further work
  - less abstract model including mRNA and "leakiness"
  - more realistic parameters
  - equivalence between more abstract and more concrete models
  - more biological modelling
- conclusions

Further work and conclusions

- further work
  - less abstract model including mRNA and "leakiness"
  - more realistic parameters
  - equivalence between more abstract and more concrete models
  - more biological modelling
- conclusions
  - HYPE for modelling hybrid biological systems

Further work and conclusions

- further work
  - less abstract model including mRNA and "leakiness"
  - more realistic parameters
  - equivalence between more abstract and more concrete models
  - more biological modelling
- conclusions
  - HYPE for modelling hybrid biological systems
  - use of flows to obtain ODEs

Further work and conclusions

- further work
  - less abstract model including mRNA and “leakiness”
  - more realistic parameters
  - equivalence between more abstract and more concrete models
  - more biological modelling
- conclusions
  - HYPE for modelling hybrid biological systems
  - use of flows to obtain ODEs
  - separation of modelling concerns

Thank you

This research was funded by the EPSRC SIGNAL Project

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

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

- activities: influences on continuous aspect of system, flows

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

HYPE syntax

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

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

- activities: influences on continuous aspect of system, flows

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

  - influence name $\iota \in IN$

HYPE syntax

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

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

- activities: influences on continuous aspect of system, flows

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

  - influence name $\iota \in IN$
  - rate $r \in \mathbb{R}$

HYPE syntax

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

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

- activities: influences on continuous aspect of system, flows

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

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

HYPE syntax

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

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

- activities: influences on continuous aspect of system, flows

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

  - influence name $\iota \in IN$
  - rate $r \in \mathbb{R}$
  - influence type $I(\vec{X})$ with $[\![I(\vec{X})]\!] = f(\vec{X})$
  - parameterised by formal variables $\vec{X}$

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{\mathsf{a}} : \alpha . C_s \mid S + S \qquad \underline{\mathsf{a}} \in \mathcal{E}, \alpha \in \mathcal{A}$

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{a}:\alpha.C_s \mid S + S \qquad \underline{a} \in \mathcal{E}, \alpha \in \mathcal{A}$
  - subcomponent names: $C_s(\vec{X}) \stackrel{def}{=} S$

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{a} : \alpha . C_s \mid S + S \qquad \underline{a} \in \mathcal{E}, \alpha \in \mathcal{A}$
  - subcomponent names: $C_s(\vec{X}) \stackrel{def}{=} S$
- components: $P ::= C(\vec{X}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{a} : \alpha . C_s \mid S + S \qquad \underline{a} \in \mathcal{E}, \alpha \in \mathcal{A}$
  - subcomponent names: $C_s(\vec{X}) \stackrel{def}{=} S$
- components: $P ::= C(\vec{X}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
  - component names: $C(\vec{X}) \stackrel{def}{=} P$ or subcomponent name

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{a} : \alpha . C_s \mid S + S \qquad \underline{a} \in \mathcal{E}, \alpha \in \mathcal{A}$
  - subcomponent names: $C_s(\vec{X}) \stackrel{def}{=} S$
- components: $P ::= C(\vec{X}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
  - component names: $C(\vec{X}) \stackrel{def}{=} P$ or subcomponent name
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \qquad L \subseteq \mathcal{E}$

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{a} : \alpha . C_s \mid S + S \qquad \underline{a} \in \mathcal{E}, \alpha \in \mathcal{A}$
  - subcomponent names: $C_s(\vec{X}) \stackrel{def}{=} S$
- components: $P ::= C(\vec{X}) \mid P \underset{L}{\bowtie} P \qquad L \subseteq \mathcal{E}$
  - component names: $C(\vec{X}) \stackrel{def}{=} P$ or subcomponent name
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \qquad L \subseteq \mathcal{E}$
- controller: $M ::= \underline{a}.M \mid 0 \mid M + M \qquad \underline{a} \in \mathcal{E}$

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

HYPE syntax (cont.)

- subcomponents: $S ::= \underline{a} : \alpha . C_s \mid S + S \qquad \underline{a} \in \mathcal{E}, \alpha \in \mathcal{A}$
  - subcomponent names: $C_s(\vec{X}) \stackrel{def}{=} S$
- components: $P ::= C(\vec{X}) \mid P \bowtie_L P \qquad L \subseteq \mathcal{E}$
  - component names: $C(\vec{X}) \stackrel{def}{=} P$ or subcomponent name
- uncontrolled system: $\Sigma ::= C(\vec{V}) \mid \Sigma \bowtie_L \Sigma \qquad L \subseteq \mathcal{E}$
- controller: $M ::= \underline{a}.M \mid 0 \mid M + M \qquad \underline{a} \in \mathcal{E}$
  $Con ::= M \mid Con \bowtie_L Con \qquad L \subseteq \mathcal{E}$
- controlled system: $ConSys ::= \Sigma \bowtie_L \underline{\text{init}}.Con \qquad L \subseteq \mathcal{E}$

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$

HYPE syntax (cont.)

- HYPE model: $(\mathit{ConSys}, \mathcal{V}, \mathcal{X}, \mathit{IN}, \mathit{IT}, \mathcal{E}, \mathcal{A}, \mathit{ec}, \mathit{iv}, \mathit{EC}, \mathit{ID})$
  - IN influence names, IT influence types

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $iv : IN \rightarrow \mathcal{V}$, association of influence names with variables

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $iv : IN \rightarrow \mathcal{V}$, association of influence names with variables
  - ID influence descriptions, $[\![I(\vec{X})]\!] = f(\vec{X})$,

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $iv : IN \rightarrow \mathcal{V}$, association of influence names with variables
  - ID influence descriptions, $[\![I(\vec{X})]\!] = f(\vec{X})$,
- well-defined HYPE model

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $iv : IN \rightarrow \mathcal{V}$, association of influence names with variables
  - ID influence descriptions, $[\![I(\vec{X})]\!] = f(\vec{X})$,
- well-defined HYPE model
  - subcomponents:

$$\mathcal{C}_s(\vec{X}) \stackrel{def}{=} \underline{a}_1 : \alpha_1.\mathcal{C}_s(\vec{X}) + \ldots + \underline{a}_n : \alpha_n.\mathcal{C}_s(\vec{X}) \quad \underline{a}_i \neq \underline{a}_j$$

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $iv : IN \rightarrow \mathcal{V}$, association of influence names with variables
  - ID influence descriptions, $[\![I(\vec{X})]\!] = f(\vec{X})$,
- well-defined HYPE model
  - subcomponents:
    $$\mathcal{C}_s(\vec{X}) \stackrel{def}{=} \underline{a}_1 : \alpha_1.\mathcal{C}_s(\vec{X}) + \ldots + \underline{a}_n : \alpha_n.\mathcal{C}_s(\vec{X}) \quad \underline{a}_i \neq \underline{a}_j$$
  - $\underline{\text{init}} : (\iota, \_, \_)$ appears exactly once
  - $\underline{a} : (\iota, \_, \_)$ appears at most once

HYPE syntax (cont.)

- HYPE model: $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$
  - IN influence names, IT influence types
  - $ec : \mathcal{E} \rightarrow EC$, association of events with event conditions
  - EC, event conditions, (activation condition, reset)
  - $iv : IN \rightarrow \mathcal{V}$, association of influence names with variables
  - ID influence descriptions, $[\![I(\vec{X})]\!] = f(\vec{X})$,
- well-defined HYPE model
  - subcomponents:
    $$\mathcal{C}_s(\vec{X}) \stackrel{def}{=} \underline{a}_1 : \alpha_1.\mathcal{C}_s(\vec{X}) + \ldots + \underline{a}_n : \alpha_n.\mathcal{C}_s(\vec{X}) \quad \underline{a}_i \neq \underline{a}_j$$
  - $\underline{\text{init}} : (\iota, \_, \_)$ appears exactly once
  - $\underline{a} : (\iota, \_, \_)$ appears at most once
  - synchronisation on shared events

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$
- (control) modes: $v \in V$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$
- (control) modes: $v \in V$
  - associated ODEs: $\dot{\mathbf{X}} = \mathit{flow}(v)$
  - initial conditions: $\mathit{init}(v)$
  - invariants: $\mathit{inv}(v)$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$
- (control) modes: $v \in V$
  - associated ODEs: $\dot{\mathbf{X}} = \mathit{flow}(v)$
  - initial conditions: $\mathit{init}(v)$
  - invariants: $\mathit{inv}(v)$
- (control) switches: $e \in E$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$
- (control) modes: $v \in V$
  - associated ODEs: $\dot{\mathbf{X}} = \mathit{flow}(v)$
  - initial conditions: $\mathit{init}(v)$
  - invariants: $\mathit{inv}(v)$
- (control) switches: $e \in E$
  - events: $\mathit{event}(e) \in \mathcal{E}$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$
- (control) modes: $v \in V$
  - associated ODEs: $\dot{\mathbf{X}} = \mathit{flow}(v)$
  - initial conditions: $\mathit{init}(v)$
  - invariants: $\mathit{inv}(v)$
- (control) switches: $e \in E$
  - events: $\mathit{event}(e) \in \mathcal{E}$
  - predicate on $\mathbf{X}$: $\mathit{jump}(e)$
  - predicate on $\mathbf{X} \cup \mathbf{X}'$: $\mathit{reset}(e)$

Hybrid automata

- $(V, E, \mathbf{X}, \mathcal{E}, \mathit{flow}, \mathit{init}, \mathit{inv}, \mathit{event}, \mathit{jump}, \mathit{reset}, \mathit{urgent})$
- $\mathbf{X} = \{X_1, \ldots, X_n\}$, $\dot{X}_j$, X'_j
- control graph: $G = (V, E)$
- (control) modes: $v \in V$
  - associated ODEs: $\dot{\mathbf{X}} = \mathit{flow}(v)$
  - initial conditions: $\mathit{init}(v)$
  - invariants: $\mathit{inv}(v)$
- (control) switches: $e \in E$
  - events: $\mathit{event}(e) \in \mathcal{E}$
  - predicate on $\mathbf{X}$: $\mathit{jump}(e)$
  - predicate on $\mathbf{X} \cup \mathbf{X}'$: $\mathit{reset}(e)$
  - boolean: $\mathit{urgent}(e)$

HYPE model to hybrid automaton

- modes V: set of reachable configurations

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations
- variables $\mathbf{X}$: variables $\mathcal{V}$

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations
- variables $\mathbf{X}$: variables $\mathcal{V}$
- if $v_j = \langle P_j, \sigma_j \rangle$ then
  $$\mathit{flow}(v_j)[X_i] \;=\; \sum\{ r[\![I(\vec{W})]\!] \mid iv(\iota) = X_i \text{ and } \sigma_j(\iota) = (r, I(\vec{W}))\}$$

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations
- variables $\mathbf{X}$: variables $\mathcal{V}$
- if $v_j = \langle P_j, \sigma_j \rangle$ then

  $$\mathit{flow}(v_j)[X_i] \;=\; \sum\{r[\![I(\vec{W})]\!] \mid iv(\iota) = X_i \text{ and } \sigma_j(\iota) = (r, I(\vec{W}))\}$$

- $\mathit{inv}(v) = \mathit{true}$

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations
- variables $\mathbf{X}$: variables $\mathcal{V}$
- if $v_j = \langle P_j, \sigma_j \rangle$ then

  $$\mathit{flow}(v_j)[X_i] \;=\; \sum\{r[\![I(\vec{W})]\!] \mid iv(\iota) = X_i \text{ and } \sigma_j(\iota) = (r, I(\vec{W}))\}$$
- $\mathit{inv}(v) = \mathit{true}$
- let e be an edge associated with $\underline{a}$ and let $ec(\underline{a}) = (\mathit{act}_{\underline{a}}, \mathit{res}_{\underline{a}})$

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations
- variables $\mathbf{X}$: variables $\mathcal{V}$
- if $v_j = \langle P_j, \sigma_j \rangle$ then

  $$flow(v_j)[X_i] \;=\; \sum\{r[\![I(\vec{W})]\!] \mid iv(\iota) = X_i \text{ and } \sigma_j(\iota) = (r, I(\vec{W}))\}$$
- $inv(v) = true$
- let e be an edge associated with $\underline{a}$ and let $ec(\underline{a}) = (act_{\underline{a}}, res_{\underline{a}})$
  - $event(e) = \underline{a}$ and $reset(e) = res_{\underline{a}}$
  - if $act_{\underline{a}} \neq \bot$ then $jump(e) = act_{\underline{a}}$ and $urgent(e) = true$
    else $jump(e) = true$ and $urgent(e) = false$

HYPE model to hybrid automaton

- modes V: set of reachable configurations
- edges E: transitions between configurations
- variables $\mathbf{X}$: variables $\mathcal{V}$
- if $v_j = \langle P_j, \sigma_j \rangle$ then

$$flow(v_j)[X_i] \;=\; \sum\{r[\![I(\vec{W})]\!] \mid iv(\iota) = X_i \text{ and } \sigma_j(\iota) = (r, I(\vec{W}))\}$$

- $inv(v) = true$
- let e be an edge associated with $\underline{a}$ and let $ec(\underline{a}) = (act_{\underline{a}}, res_{\underline{a}})$
  - $event(e) = \underline{a}$ and $reset(e) = res_{\underline{a}}$
  - if $act_{\underline{a}} \neq \bot$ then $jump(e) = act_{\underline{a}}$ and $urgent(e) = true$
    else $jump(e) = true$ and $urgent(e) = false$
- $init(v) = \begin{cases} res_{\underline{init}} & \text{if } v = \langle P, \sigma \rangle \text{ with primes removed} \\ false & otherwise \end{cases}$