Robustness Analysis and Tuning of Synthetic Gene Networks

Grégory Batt
Contraintes research group, INRIA

Oct 20, 2011
Combining Discrete Abstraction and Model Checking for the Analysis of Synthetic Gene Networks

Grégory Batt
Contraintes research group, INRIA
**Escherichia coli** stress responses

- *E. coli* is able to adapt to a variety of stresses in its environment
  
  Model organism for understanding of decision-making processes in single-cell organisms

- *E. coli* is easy to manipulate in the laboratory
  
  Model organism for understanding adaptation of pathogenic bacteria to their host


- Nutritional stress
- Osmotic stress
- Heat shock
- Cold shock
- ...
Nutritional stress response in *E. coli*

- Response of *E. coli* to nutritional stress conditions: transition from exponential phase to stationary phase

- Important developmental decision: profound changes of morphology, metabolism, gene expression,...
Carbon starvation response network

- Genetic regulatory network controlling *E. coli* carbon starvation response

Ropers et al. (2006), *Biosystems*, 84(2):124-52

- No global view of functioning of network available, despite abundant knowledge on network components
  
  Complex dynamics and lack of quantitative information
Synthetic transcriptional cascade in *E. coli*

- **Synthetic biology**: design and construct biological systems with desired behaviors

Hooshangi *et al.*, *PNAS*, 05

Ultrasensitive input/output response
Synthetic transcriptional cascade in *E. coli*

- **Synthetic biology**: design and construct biological systems with desired behaviors

How can the network be tuned?

- Rational design and tuning is difficult
  - Large parameter uncertainties and fluctuating cellular environment
Analysis of gene networks

- Need for mathematical methods and computational tools for
  - verifying dynamical properties of networks
  - finding network modifications such that expected properties are satisfied

- Constraints on gene network analysis
  - genetic regulations are non-linear phenomena (→ non-linear models)
  - networks include large number of genes (→ efficient approach)
  - partial knowledge on network parameters: either qualitative or quantitative with large uncertainties (→ work with parameter constraints)
Overview

I. Introduction: analysis of genetic regulatory networks
II. Gene network models and dynamical property specifications
III. Analysis of piecewise-multiaffine (PMA) models
IV. Discussion and conclusions
Overview

I. Introduction: analysis of genetic regulatory networks

II. Gene network models and dynamical property specifications
   1. Hill-type, piecewise-affine and piecewise-multiaffine models
   2. Dynamical properties expressed in temporal logic

III. Analysis of piecewise-multiaffine (PMA) models

IV. Discussion and conclusions
Gene network models

- Differential equation models

\[
\dot{x}_i = f_i(x) = \sum \kappa_i^j p_i^j(x) - \sum \gamma_i^j d_i^j(x) x_i
\]

- Different regulation functions yield different classes of models

- Hill function ramp function step function

\[
h^+(x_i, \theta_i, \eta_i)
\]

- Hill-type models

\[
r^+(x_i, \theta_i, \theta'_i)
\]

- PMA models

\[
s^+(x_i, \theta_i)
\]

- PA models
Gene network models

- Mutual inhibition example

\[ \dot{x}_a = \kappa_a \ reg_1^-(x_b) \ reg_2^-(x_a) - \gamma_a \ x_a \]

\[ \dot{x}_b = \kappa_b \ reg_3^-(x_a) - \gamma_b \ x_b \]
Gene network models

- Mutual inhibition example

\[
\begin{align*}
\dot{x}_a &= \kappa_a \, \text{reg}_1^- (x_b) \, \text{reg}_2^- (x_a) - \gamma_a \, x_a \\
\dot{x}_b &= \kappa_b \, \text{reg}_3^- (x_a) - \gamma_b \, x_b
\end{align*}
\]

- Hill-type models

\[
\begin{align*}
\dot{x}_a &= \kappa_a \, h^- (x_b, \theta_b, \eta_b) \, h^- (x_a, \theta_a^2, \eta_a^2) - \gamma_a \, x_a \\
\dot{x}_b &= \kappa_b \, h^- (x_a, \theta_a^1, \eta_a^1) - \gamma_b \, x_b
\end{align*}
\]

\(\theta_{1a}, \theta_{2a}, \theta_b\): threshold concentrations, \(\eta_{1a}, \eta_{2a}, \eta_b\): Hill coefficients
Gene network models

- Mutual inhibition example

\[ \dot{x}_a = \kappa_a \, \text{reg}_1^-(x_b) \, \text{reg}_2^-(x_a) - \gamma_a \, x_a \]
\[ \dot{x}_b = \kappa_b \, \text{reg}_3^-(x_a) - \gamma_b \, x_b \]

- PMA models

\[ \dot{x}_a = \kappa_a \, r^-(x_b, \theta_b^1, \theta_b^2) \, r^-(x_a, \theta_a^3, \theta_a^4) - \gamma_a \, x \]
\[ \dot{x}_b = \kappa_b \, r^-(x_a, \theta_a^1, \theta_a^2) - \gamma_b \, x_b \]
Gene network models

- Mutual inhibition example

- PA models

\[ \dot{x}_a = \kappa_a \, s^{-}(x_b, \theta_b) \, s^{-}(x_a, \theta_a^2) - \gamma_a \, x_a \]

\[ \dot{x}_b = \kappa_b \, s^{-}(x_a, \theta_a^1) - \gamma_b \, x_b \]

\[ \dot{x}_a = \kappa_a \, \text{reg}_1^{-}(x_b) \, \text{reg}_2^{-}(x_a) - \gamma_a \, x_a \]

\[ \dot{x}_b = \kappa_b \, \text{reg}_3^{-}(x_a) - \gamma_b \, x_b \]

\[ \theta_{a1}, \theta_{a2}, \theta_b: \text{threshold concentrations} \]
Assumptions for gene network models

- Genetic regulatory network are heterogeneous in nature
  - Different types of biochemical processes (gene expression, enzymatic reactions, protein complex formation, …)
  - Different characteristic time-scales associated with these processes (ms to min)
  - Different types of data, specific for each process (transcriptomics, proteomics, metabolomics, interactomics, …)

- Assumptions underlying genetic regulatory networks:
  - Reference time-scale is that of protein synthesis/degradation
  - Observations are measurements of gene products (mRNA, protein)
  - Fast reactions (w.r.t. reference time-scale) are in quasi-steady state

Heinrich and Schuster (1996), The Regulation of Cellular Systems, Chapman & Hall
Specifications of dynamical properties

- Dynamical properties expressed in temporal logic (CTL or LTL)
  - set of atomic proposition \( \Pi : \ x_i < \lambda_i, \ x_i > \lambda_i, \ \dot{x}_i < 0, \ \dot{x}_i = 0, \ \dot{x}_i > 0 \)
  - usual logical operators \( \neg f, \ f_1 \land f_2, \ f_1 \lor f_2, \ f_1 \rightarrow f_2, \ \ldots \)
  - temporal operators \( X f, \ F f, \ G f, \ f_1 \mathcal{U} f_2, \ \ldots \)
  - path operators \( \mathcal{E} f \) and \( \mathcal{A} f \)
  - some restrictions apply on combination of path and temporal operators
Specifications of dynamical properties

- Dynamical properties expressed in temporal logic (CTL or LTL)
  - set of atomic proposition $\Pi$: $x_i < \lambda_i$, $x_i > \lambda_i$, $\dot{x}_i < 0$, $\dot{x}_i = 0$, $\dot{x}_i > 0$
  - usual logical operators $\neg f$, $f_1 \land f_2$, $f_1 \lor f_2$, $f_1 \rightarrow f_2$, ...
  - temporal operators $X f$, $F f$, $G f$, $f_1 U f_2$, ...
  - path operators $E f$ and $A f$
  - some restrictions apply on combination of path and temporal operators

\[ \begin{align*}
\phi_1 = u_{aTc} < 100 & \rightarrow FG(2.5 \times 10^2 < x_{eyfp} < 5 \times 10^2) \\
\land 100 < u_{aTc} < 200 & \rightarrow FG(2.5 \times 10^2 < x_{eyfp} < 10^6) \\
\land u_{aTc} > 200 & \rightarrow FG(5 \times 10^5 < x_{eyfp} < 10^6).
\end{align*} \]
Overview

I. Introduction: analysis of genetic regulatory networks

II. Gene network models and dynamical property specifications
   1. Hill-type, piecewise-affine and piecewise-multiaffine models
   2. Dynamical properties expressed in temporal logic

III. Analysis of piecewise-multiaffine (PMA) models

IV. Discussion and conclusions
Overview

I. Introduction: analysis of genetic regulatory networks

II. Gene network models and dynamical property specifications

III. Analysis of piecewise-multiaffine (PMA) models
   1. State-space partition and discrete abstraction
   2. Verification for fixed parameters
   3. Parameter equivalence classes and verification for sets of parameters
   4. Application to tuning the transcriptional cascade

IV. Discussion and conclusions
Analysis of PMA models

- Analysis of the dynamics in state space: \( \dot{x} = f(x), \quad x \in \mathcal{X} \)

\[
\begin{align*}
\dot{x}_a &= \kappa_a \, r^-(x_b, \theta_b^1, \theta_b^2) \, r^-(x_a, \theta_a^3, \theta_a^4) - \gamma_a \, x_a \\
\dot{x}_b &= \kappa_b \, r^-(x_a, \theta_a^1, \theta_a^2) - \gamma_b \, x_b
\end{align*}
\]

Plahte et al., Dyn. Stabil. Syst, 94
Analysis of PMA models

- Partition of phase space into rectangles
Embedding transition system

- PMA system, $\Sigma = (f, \Pi)$, associated with embedding transition system, $T_X(p) = (X_R, \rightarrow_X, p, \models_X)$, where
Embedding transition system

- PMA system, $\Sigma = (f, \Pi)$, associated with embedding transition system, $T\chi(p) = (\mathcal{X}_R, \rightarrow, \models)$, where
  - $\mathcal{X}_R$ continuous state space
Embedding transition system

- **PMA system,** $\Sigma = (f, \Pi)$ associated with embedding transition system, $T_{x(p)} = (x_{\mathcal{R}}, \rightarrow_{x, p}, \models x)$, where
  - $x_{\mathcal{R}}$ continuous state space
  - $\rightarrow_{x, p}$ transition relation
Embedding transition system

- PMA system, $\Sigma = (f, \Pi)$ associated with embedding transition system, $T\chi(p) = (\mathcal{X}_R, \rightarrow\chi, p, \models\chi)$, where
  - $\mathcal{X}_R$ continuous state space
  - $\rightarrow\chi, p$ transition relation
  - $\models\chi$ satisfaction relation

\[
\begin{align*}
x^1 & \models\chi x_a < \theta^1_a, & x^1 & \models\chi x_b < \theta^1_b, \\
x^4 & \models\chi x_a < \theta^1_a, & x^4 & \models\chi x_b > \theta^1_b
\end{align*}
\]
Embedding transition system

- **PMA system**, $\Sigma = (f, \Pi)$ associated with embedding transition system, $T_\chi(p) = (\mathcal{X}_R, \rightarrow_{\mathcal{X},p}, \models_{\mathcal{X}})$, where
  - $\mathcal{X}_R$ continuous state space
  - $\rightarrow_{\mathcal{X},p}$ transition relation
  - $\models_{\mathcal{X}}$ satisfaction relation

$T_\chi(p)$ captures almost all solution trajectories of $\Sigma$

\[ x^1 \models_{\mathcal{X}} x_a < \theta^1_a, \quad x^1 \models_{\mathcal{X}} x_b < \theta^1_b, \]
\[ x^4 \models_{\mathcal{X}} x_a < \theta^1_a, \quad x^4 \models_{\mathcal{X}} x_b > \theta^1_b \]
Discrete abstraction

- **Discrete transition system**, $T_\mathcal{R}(p) = (\mathcal{R}, \rightarrow_\mathcal{R}, p, \models_\mathcal{R})$, where
Discrete abstraction

- **Discrete transition system**, $T_{\mathcal{R}}(p) = (\mathcal{R}, \rightarrow_{\mathcal{R},p}, \models_{\mathcal{R}})$, where
  - $\mathcal{R}$ finite set of rectangles
Discrete abstraction

- **Discrete transition system**, \( T_{\mathcal{R}}(p) = (\mathcal{R}, \rightarrow_{\mathcal{R},p}, \models_{\mathcal{R}}) \), where
  - \( \mathcal{R} \) finite set of rectangles
  - \( \rightarrow_{\mathcal{R},p} \) quotient transition relation

\[ R^1 \rightarrow_{\mathcal{R},p} R^1, \quad R^1 \rightarrow_{\mathcal{R},p} R^6, \quad R^6 \rightarrow_{\mathcal{R},p} R^{11} \]
Discrete abstraction

- **Discrete transition system**, $T_\mathcal{R}(p) = (\mathcal{R}, \rightarrow_{\mathcal{R},p}, \models_{\mathcal{R}})$, where
  - $\mathcal{R}$ finite set of rectangles
  - $\rightarrow_{\mathcal{R},p}$ quotient transition relation
  - $\models_{\mathcal{R}}$ quotient satisfaction relation

\[ R^1 \models_{\mathcal{R}} x_a < \theta^1_a, \quad R^1 \models_{\mathcal{R}} x_b < \theta^1_b, \quad R^{11} \models_{\mathcal{R}} x_a < \theta^1_a \]
Discrete abstraction

- **Discrete transition system**, $T_{\mathcal{R}}(p) = (\mathcal{R}, \to_{\mathcal{R},p}, \models_{\mathcal{R}})$, where
  - $\mathcal{R}$ finite set of rectangles
  - $\to_{\mathcal{R},p}$ quotient transition relation
  - $\models_{\mathcal{R}}$ quotient satisfaction relation
Discrete abstraction

- **Discrete transition system**, $T_{\mathcal{R}}(p) = (\mathcal{R}, \rightarrow_{\mathcal{R},p}, \models_{\mathcal{R}})$, where
  - $\mathcal{R}$ finite set of rectangles
  - $\rightarrow_{\mathcal{R},p}$ quotient transition relation
  - $\models_{\mathcal{R}}$ quotient satisfaction relation

- Quotient transition system $T_{\mathcal{R}}(p)$ is a **simulation** of $T_{\chi}(p)$
Computation for fixed parameters

- Multi-affine functions on rectangular regions

In every rectangular region, the flow is a convex combination of its values at the vertices

Computation for fixed parameters

- Multiaffine functions on rectangular regions

In every rectangular region, the flow is a convex combination of its values at the vertices


- There is a transition between two adjacent rectangles iff for some common vertex, the flow and relative position agree

\[ (R, R') \in \rightarrow_{R,p} \text{ iff } \exists v \in V_R \cap V_{R'} \text{ such that } f_i(v, p)(c'_i - c_i) > 0 \]

- \( T_R(p) \) can be **computed** by evaluating \( f \) at all vertices

Batt *et al*, HSCC’07
Uncertain PMA systems

- **Equivalence relation** on parameters
  
  Equivalent parameters correspond to the same discrete abstraction $T_{\mathcal{K}}(p)$
Uncertain PMA systems

- **Equivalence relation** on parameters
  - Equivalent parameters correspond to the same discrete abstraction \( T_\mathcal{R}(p) \)
- **Discrete abstraction** depends on parameter values \( p \) (\( \kappa \)'s and \( \gamma \)'s)
  - Transitions depend on **signs** of \( f_i(v, p) \)
    \[
    (R, R') \in \rightarrow_{\mathcal{R},p} \iff \exists v \in \mathcal{V}_R \cap \mathcal{V}_{R'} \text{ such that } f_i(v, p)(c'_i - c_i) > 0
    \]
  - \( f_i(v, p) \) is an **affine** expression in \( p \)
    \[
    f_i(v, p) = \sum_{j \in P_i} \kappa_i^j p_i^j(v) - \sum_{j \in D_i} \gamma_i^j d_i^j(v) v_i = a^T p + b
    \]
Uncertain PMA systems

- **Equivalence relation** on parameters
  Equivalent parameters correspond to the same discrete abstraction $T_R(p)$

- **Discrete abstraction depends on parameter values** $p$ ($\kappa$'s and $\gamma$'s)

\[
\begin{align*}
\dot{x}_a &= \kappa_a \, r^{-}(x_b, \theta^1_b, \theta^2_b) \, r^{-}(x_a, \theta^3_a, \theta^4_a) - \gamma_a \, x_a \\
\dot{x}_b &= \kappa_b \, r^{-}(x_a, \theta^1_a, \theta^2_a) - \gamma_b \, x_b
\end{align*}
\]

\[
\begin{align*}
f_a(8, 0, \kappa_a, \kappa_b) &= \kappa_a - 8 \\
f_a(8, 8, \kappa_a, \kappa_b) &= \kappa_a - 8
\end{align*}
\]
Uncertain PMA systems

- **Equivalence relation on parameters**
  Equivalent parameters correspond to the same discrete abstraction $T_{\mathcal{R}}(p)$

- **Discrete abstraction depends on parameter values** $p$ ($\kappa$'s and $\gamma$'s)

\[
\begin{align*}
\dot{x}_a &= \kappa_a \ r^{-}(x_b, \theta^1_b, \theta^2_b) \ r^{-}(x_a, \theta^3_a, \theta^4_a) - \gamma_a \ x_a \\
\dot{x}_b &= \kappa_b \ r^{-}(x_a, \theta^1_a, \theta^2_a) - \gamma_b \ x_b
\end{align*}
\]

\[
\begin{align*}
f_a(8, 0, \kappa_a, \kappa_b) &= \kappa_a - 8 \\
f_a(8, 8, \kappa_a, \kappa_b) &= \kappa_a - 8 \\
f_b(0, 8, \kappa_a, \kappa_b) &= \kappa_b - 16 \\
f_b(8, 8, \kappa_a, \kappa_b) &= \kappa_b - 16
\end{align*}
\]
Uncertain PMA systems

- **Equivalence relation** on parameters
  
  Equivalent parameters correspond to the same discrete abstraction $T_{\mathcal{R}}(p)$

- **Discrete abstraction** depends on parameter values $p$ ($\kappa'$s and $\gamma'$s)

- **Polyhedral partition** of parameter space by affine constraints

![Diagram]

- Parameters in a same region $P \in \mathcal{P}$ are all equivalents

  \[
  f_a(8, 0, \kappa_a, \kappa_b) = \kappa_a - 8 \\
  f_a(8, 8, \kappa_a, \kappa_b) = \kappa_a - 8 \\
  f_b(0, 8, \kappa_a, \kappa_b) = \kappa_b - 16 \\
  f_b(8, 8, \kappa_a, \kappa_b) = \kappa_b - 16 \\
  \ldots
  \]
Uncertain PMA systems

- **Valid parameters**: parameters for which property $\phi$ is true
Uncertain PMA systems

- **Valid parameters**: parameters for which property $\phi$ is true
- **Finding sets of valid parameters**

1. compute partition $\mathcal{P}$
2. for every region $P \in \mathcal{P}$
3. pick one parameter $p \in P$
4. compute $T_R(p)$
5. if $T_R(p) \models \phi$ then $\phi$ is true for every $p \in P$
6. end for
Uncertain PMA systems

- **Valid parameters**: parameters for which property $\phi$ is true
- **Finding sets of valid parameters**

1. compute partition $\mathcal{P}$
2. for every region $P \in \mathcal{P}$
3. pick one parameter $p \in P$
4. compute $T_R(p)$
5. if $T_R(p) \models \phi$ then $\phi$ is true for every $p \in P$
6. end for

bistability property:

\[
\phi_2 = (x_a < \theta_a^1 \land x_b > \theta_b^2 \rightarrow G (x_a < \theta_a^1 \land x_b > \theta_b^2))
\land (x_b < \theta_b^1 \land x_a > \theta_a^3 \rightarrow G (x_b < \theta_b^1 \land x_a > \theta_a^3))
\]
Uncertain PMA systems

- **Valid parameters:** parameters for which property $\phi$ is true
- **Finding sets of valid parameters**

1. compute partition $\mathcal{P}$
2. for every region $P \in \mathcal{P}$
3. pick one parameter $p \in P$
4. compute $T_R(p)$
5. if $T_R(p) \models \phi$ then $\phi$ is true for every $p \in P$
6. end for

bistability property:

$$\phi_2 = (x_a < \theta^1_a \land x_b > \theta^2_b \rightarrow G (x_a < \theta^1_a \land x_b > \theta^2_b)) \land (x_b < \theta^1_b \land x_a > \theta^3_a \rightarrow G (x_b < \theta^1_b \land x_a > \theta^3_a))$$

- **Inefficient approach:** number of regions grows exponentially with number of rectangles in state space
Uncertain PMA systems

- Reasoning for polyhedral parameter sets: $T_R^\exists(P)$ and $T_R^\forall(P)$
Uncertain PMA systems

- Reasoning for polyhedral parameter sets: $T^\exists_R(P)$ and $T^\forall_R(P)$
  - $T^\exists_R(P)$ and $T^\forall_R(P)$ correspond to over- and under-approximations of $T_R(p)$ when $p$ varies in $P$

  $$(R, R') \in \rightarrow^\exists_{R,P} \iff \exists p \in P \text{ such that } (R, R') \in \rightarrow_{R,p} \text{ in } T_R(p),$$

  $$(R, R') \in \rightarrow^\forall_{R,P} \iff \forall p \in P, (R, R') \in \rightarrow_{R,p} \text{ in } T_R(p).$$
Uncertain PMA systems

- Reasoning for polyhedral parameter sets: $T^3_R(P)$ and $T^\forall_R(P)$
  - $T^3_R(P)$ and $T^\forall_R(P)$ correspond to over- and under-approximations of $T_R(p)$ when $p$ varies in $P$
    
    \[(R, R') \in \rightarrow^3_R, P \iff \exists p \in P \text{ such that } (R, R') \in \rightarrow_{R, p} \text{ in } T_R(p),\]

    \[(R, R') \in \rightarrow^\forall_R, P \iff \forall p \in P, (R, R') \in \rightarrow_{R, p} \text{ in } T_R(p).\]

  - $T^3_R(P)$ and $T^\forall_R(P)$ can be used to prove properties for parameter sets
    
    if $T^3_R(P) \models \phi$, then $\forall p \in P, T_X(p) \models \phi$
    
    $\phi$ true for all $p \in P$

    if $T^\forall_R(P) \not\models \phi$, then $\forall p \in P, T_R(p) \not\models \phi$
    
    no conclusion possible
Uncertain PMA systems

- Reasoning for polyhedral parameter sets: $T_{R}^{3}(P)$ and $T_{R}^{\forall}(P)$
  - $T_{R}^{3}(P)$ and $T_{R}^{\forall}(P)$ correspond to over- and under-approximations of $T_{R}(p)$ when $p$ varies in $P$
    
    $\left(R, R'\right) \in \rightarrow_{R,P}^{3}$ iff $\exists p \in P$ such that $\left(R, R'\right) \in \rightarrow_{R,p}$ in $T_{R}(p)$,
    
    $\left(R, R'\right) \in \rightarrow_{R,P}^{\forall}$ iff $\forall p \in P$, $\left(R, R'\right) \in \rightarrow_{R,p}$ in $T_{R}(p)$.

  - $T_{R}^{3}(P)$ and $T_{R}^{\forall}(P)$ can be used to prove properties for parameter sets
    
    if $T_{R}^{3}(P) \models \phi$, then $\forall p \in P$, $T_{X}(p) \models \phi$
    
    if $T_{R}^{\forall}(P) \not\models \phi$, then $\forall p \in P$, $T_{R}(p) \not\models \phi$
    
    no conclusion possible

  - $T_{R}^{3}(P)$ and $T_{R}^{\forall}(P)$ can be computed using polyhedral operations
Uncertain PMA systems

- Hierarchical exploration of parameter space
  Model checking while constructing the partition
Uncertain PMA systems

- Hierarchical exploration of parameter space
Model checking liveness properties

- Verification of liveness properties generally fails
  - Liveness properties state that something will eventually happen
  - Fails because quantitative aspects of time abstracted away

- Need to enforce progress of time in discrete abstraction
  - Rule out spurious counter examples

"Eventually system remains in $R^3$"

$$FG (x_a > \theta_a^2 \land x_b < \theta_b)$$
An execution is time diverging if it is an abstraction of at least one solution on \([0, \infty[\)
Transient regions

- A region is transient if it is left by all solutions in finite time

**Definition 6.** Let \( p \in \mathcal{P} \) and \( U \subseteq \mathcal{X} \) be a union of rectangles \( R \in \mathcal{R} \). \( U \) is transient for parameter \( p \) if for every solution \( \xi \) of (1) such that \( \xi(0) \in U \), there exists \( \tau > 0 \) such that \( \xi(\tau) \notin \overline{U} \).
Ruling out time converging executions

Any execution remains eventually always in a SCC

Proposition 1. Let \( p \in \mathcal{P} \). If an execution \( e_\mathcal{R} \) of \( \mathcal{T}_\mathcal{R}(p) \) is time-diverging, then \( SCC(e_\mathcal{R}) \) is not transient for \( p \).

Batt et al, TACAS’07
Ruling out time converging executions

- Any execution remains eventually always in a SCC

**Proposition 1.** Let $p \in \mathcal{P}$. If an execution $e_R$ of $T_R(p)$ is time-diverging, then $SCC(e_R)$ is not transient for $p$.

**Approach** (assuming transient regions can be computed)
- compute discrete abstraction and its SCCs
- test whether SCC is transient and label every state in transient SCCs by 'transient'
- test $\varphi'$: $\neg FG 'transient' \rightarrow \varphi$

Batt et al, TACAS'07

\[ \neg F G 'transient' \rightarrow F G (x_a > \theta_a^2 \land x_b < \theta_b) \]
Computation of transient regions

- Based on convexity properties of multiaffine functions

Proposition 3. Let \( p \in \mathcal{P} \) and \( U \subseteq \mathcal{X} \) be a union of rectangles \( R \in \mathcal{R} \). If

\[
0 \notin \text{hull}(\{f(v, p) \mid v \in \mathcal{V}_R, R \subseteq U\}),
\]

then \( U \) is transient for parameter \( p \).
Computation of transient regions

- Based on convexity properties of multiaffine functions

Proposition 3. Let \( p \in \mathcal{P} \) and \( U \subseteq X \) be a union of rectangles \( R \in \mathcal{R} \). If

\[
0 \notin \text{hull}(\{ f(v, p) \mid v \in \mathcal{V}_R, R \subseteq U \}),
\]

then \( U \) is transient for parameter \( p \).

- Generalized to reason with sets of parameters

Proposition 2. Let \( P \subseteq \mathcal{P} \).

(a) If an execution \( e_R \) of \( T^\exists_R(P) \) is time-diverging, then for some \( p \in P \), \( \text{SCC}(e_R) \) is not transient for \( p \).

(b) If an execution \( e_R \) of \( T^\forall_R(P) \) is time-diverging, then for all \( p \in P \), \( \text{SCC}(e_R) \) is not transient for \( p \).

Proposition 4. Let \( P \subseteq \mathcal{P} \) be a polytope and \( U \subseteq X \) be a union of rectangles \( R \in \mathcal{R} \). If \( 0 \notin \text{hull}(\{ f(v, w) \mid v \in \mathcal{V}_R, R \subseteq U, w \in \mathcal{V}_P \}) \), then \( U \) is transient for all parameters \( p \in P \).
Computation of transient regions

- Based on convexity properties of multi-affine functions

  \[ \text{Proposition 3. Let } p \in \mathcal{P} \text{ and } U \subseteq \mathcal{X} \text{ be a union of rectangles } R \in \mathcal{R}. \text{ If } 0 \notin \text{hull}(\{ f(v, p) \mid v \in \mathcal{V}_R, R \subseteq U \}), \]

  \text{then } U \text{ is transient for parameter } p. \]

- Generalized to reason with sets of parameters

- Testing whether SCCs are transient can be decided by solving linear optimization problems

  \text{Batt et al, TACAS'07}

- Approach implemented in Matlab tool \textbf{RoVerGeNe 3.0}

  Exploits tools for polyhedral and graph operations, and model checker

  \text{http://iasi.bu.edu/~batt/rovergene/rovergene.htm}
Summary

- **Discrete abstraction**
  - $T_R(p)$ provides **finite** description of the dynamics of system $\Sigma$ in state space.
  - $T_R(p)$ is a **conservative approximation** of $\Sigma$.
  - $T_R(p)$ depends on the signs of $f$ at the vertices of state partition.

- **Verification under parameter uncertainty**
  - **Equivalence relation** on parameters.
  - **Polyhedral partition** of parameter space into regions of equivalent parameters.
  - Use of $T^{\exists}_R(P)$ and $T^{\forall}_R(P)$ for reasoning on **parameter sets**.
  - $T^{\exists}_R(P)$ and $T^{\forall}_R(P)$ can be **computed** using polyhedral operations and model checked.
  - Enforcing **progress of time** for liveness checking: transient regions.
Transcriptional cascade: modeling

- Tuning a transcriptional cascade

Hooshangi et al., PNAS, 05
Transcriptional cascade: modeling

- **Differential equation model**

\[
\begin{align*}
\dot{x}_{tetR} &= \kappa_{tetR} - \gamma_{tetR} x_{tetR}, \\
\dot{x}_{lacI} &= \kappa_{0}^{lacI} + \kappa_{lacI} (1 - r^{+}(x_{tetR}, \theta_{tetR}^{1}, \theta_{tetR}^{2}) r^{-}(u_{aTc}, \theta_{aTc}^{1}, \theta_{aTc}^{2})) - \gamma_{lacI} x_{lacI}, \\
\dot{x}_{cI} &= \kappa_{0}^{cI} + \kappa_{cI} r^{-}(x_{lacI}, \theta_{lacI}^{1}, \theta_{lacI}^{2}) - \gamma_{cI} x_{cI}, \\
\dot{x}_{eyfp} &= \kappa_{0}^{eyfp} + \kappa_{eyfp} r^{-}(x_{cI}, \theta_{cI}^{1}, \theta_{cI}^{2}) - \gamma_{eyfp} x_{eyfp},
\end{align*}
\]

- **Parameter identification**

<table>
<thead>
<tr>
<th>variable</th>
<th>$\kappa_{0}^{i}$</th>
<th>$\kappa_{i}$</th>
<th>$\gamma_{i}$</th>
<th>$\theta_{i}^{1}$</th>
<th>$\theta_{i}^{2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_{aTc}$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>80</td>
<td>4000</td>
</tr>
<tr>
<td>$x_{tetR}$</td>
<td>-</td>
<td>260</td>
<td>0.013</td>
<td>4500</td>
<td>5500</td>
</tr>
<tr>
<td>$x_{lacI}$</td>
<td>2.405</td>
<td>875.6</td>
<td>0.013</td>
<td>500</td>
<td>4500</td>
</tr>
<tr>
<td>$x_{cI}$</td>
<td>3.9</td>
<td>386</td>
<td>0.013</td>
<td>600</td>
<td>23000</td>
</tr>
<tr>
<td>$x_{eyfp}$</td>
<td>4.58</td>
<td>4048</td>
<td>0.013</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Verification: does the system satisfy its specifications?
Use identified parameter values. Answer: ‘No’

\[ \phi_1 = \begin{align*}
&u_{aTc} < 100 \rightarrow FG(2.5 \times 10^2 < x_{eyfp} < 5 \times 10^2) \\
&\land 100 < u_{aTc} < 200 \rightarrow FG(2.5 \times 10^2 < x_{eyfp} < 10^6) \\
&\land u_{aTc} > 200 \rightarrow FG(5 \times 10^5 < x_{eyfp} < 10^6).
\]
Transcriptional cascade: analysis

- **Verification**: does the system satisfy its specifications?
  Use identified parameter values. Answer: ‘No’

- **Tuning**: search for valid parameter sets
  \[ \kappa_{\text{lacI}} \in [10, 4000], \kappa_{cI} \in [10, 8000], \text{ and } \kappa_{\text{eyfp}} \in [100, 20000] \]
  Answer: 15 sets found (<4 h., 1500 rectangles, 18 parameter constraints)

Comparison with numerical simulation results in parameter space and for input/output behavior

Batt et al., *Bioinformatics*, 07
Transcriptional cascade: analysis

- **Verification**: does the system satisfy its specifications?
  
  Use identified parameter values. Answer: ‘No’

- **Tuning**: search for valid parameter sets
  
  \[ \kappa_{lacI} \in [10, 4000], \kappa_{cI} \in [10, 8000], \text{ and } \kappa_{eyfp} \in [100, 20000] \]
  
  Answer: 15 sets found (<4 h., 1500 rectangles, 18 parameter constraints)

- **Robustness**: test robustness of proposed tuning
  
  Assume \( \kappa_{lacI} = 2591, \kappa_{cI} = 550 \text{ and } \kappa_{eyfp} = 8000 \)
  
  Is property true if all parameters vary in a ±10% interval? in a ±20%?
  
  (threshold parameters excluded)

  Answer: ‘Yes’ for ±10% variation

  (<4 h.) ‘No’ for ±20% variation

Batt et al, Bioinformatics, 07
Overview

I. Introduction: analysis of genetic regulatory networks

II. Gene network models and dynamical property specifications

III. Analysis of piecewise-multiaffine (PMA) models
   1. State-space partition and discrete abstraction
   2. Verification for fixed parameters
   3. Parameter equivalence classes and verification for sets of parameters
   4. Application to tuning the transcriptional cascade

IV. Discussion and conclusions
Overview

I. Introduction: analysis of genetic regulatory networks
II. Gene network models and dynamical property specifications
III. Analysis of piecewise-multiaffine (PMA) models
IV. Discussion and conclusions
Conclusion

- Analysis of partially-known models of gene networks
  - well-adapted to experimental data
  - verification of robust properties
  - parameter constraint synthesis

- Use of piecewise-multiaffine models
  - complex dynamics described by set of locally-simple differential equations
  - tailored combination of discrete abstraction and model checking

- Approaches can answer efficiently non-trivial questions on networks of biological interest
Discussion

- **Related work: analysis of uncertain biological networks**
  - Symbolic reachability analysis of PA models using discrete abstractions
    - Ghosh and Tomlin, *Systems Biology*, 04
  - Iterative search in dense parameter space of ODE models using model checking
  - Exhaustive exploration of finite parameter space of logical models using model checking
    - Bernot et al., *J. Theor. Biol.*, 04
    - Corblin et al., *BMC Bioinfo*, 10
  - Direct approaches for reachability analysis
    - Dang et al., *TCS*, 11

- **Further work**
  - Automatic state-space partition refinement
  - Verification of properties involving timing constraints
  - Compositional verification to exploit network modularity
Acknowledgements

Thank you for your attention!

References