PhD defense

Algebraic Modeling of the Multi-Scale Dynamics of Biological Regulatory Networks

Maxime FOLSCHETTE
MeForBio / IRCCyN / École centrale de Nantes (Nantes, France)
maxime.folschette@irccyn.ec-nantes.fr
http://maxime.folschette.name/

2014/10/08

Rapporteurs : Jean-Paul COMET, Professeur des universités, Université de Nice – Sophia Antipolis
Anne SIEGEL, Directrice de recherche CNRS, IRISA (CNRS & Université Rennes 1), Inria Rennes

Examinateurs : Mireille RÉGNIER, Directrice de recherche Inria, École polytechnique & Université Paris-Sud 11
Denis THIEFFRY, Professeur des universités, École normale supérieure

Directeur de thèse : Olivier ROUX, Professeur des universités, École centrale de Nantes
Co-encadrant de thèse : Morgan MAGNIN, Maître de conférences, École centrale de Nantes
The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension
The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

The required analysis has an impact on modeling
- The modeling tools must be adapted to the observed properties
Introduction

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

The required analysis has an impact on modeling
- The modeling tools must be adapted to the observed properties

Modeling choices have an impact on the results of the analysis
- The level of details changes the quantity of obtained info
- The size of the model increases the analysis duration
The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

- The modeling tools must be adapted to the observed properties

The required analysis has an impact on modeling
- The modeling tools must be adapted to the observed properties

Modeling choices have an impact on the results of the analysis
- The level of details changes the quantity of obtained info
- The size of the model increases the analysis duration

The modeling and analysis steps of a system are strongly linked
Overview of This Presentation

State of the Art of the modeling of biological regulatory networks

- Discrete asynchronous representations and Thomas modeling
- Standard Process Hitting
Overview of This Presentation

State of the Art of the modeling of biological regulatory networks

- Discrete asynchronous representations and Thomas modeling
- Standard Process Hitting

Enriching the Process Hitting

- Integration of temporal constraints
- Synchronicity between actions
  → Adding of priorities, neutralizing edges or synchronous actions
Overview of This Presentation

State of the Art of the modeling of biological regulatory networks
- Discrete asynchronous representations and Thomas modeling
- Standard Process Hitting

Enriching the Process Hitting
- Integration of temporal constraints
- Synchronicity between actions
  \[\rightarrow\] Adding of priorities, neutralizing edges or synchronous actions

Analysis of the Process Hitting
- Correction of the cooperative sorts
- Static analysis of reachability
- Equivalences and links with other formalisms
Abstractions of the Representation
Abstractions of the Representation
Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]
Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]

- Unknown real values of concentrations or continuous activity levels → Abstracted as thresholds or discrete levels
- Continuous variations of the real values → Unitary dynamics
- Simultaneous crossings of two thresholds never occurs → Asynchronous dynamics

\[ \theta_{ba} \] concentration of B
\[ \theta_{ab} \] concentration of A
\[ 0 \quad 1 \quad \text{discrete levels of concentration for } B \]
\[ 0 \quad 1 \quad 2 \quad \text{discrete levels of concentration for } A \]
Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]

- Unknown real values of concentrations or continuous activity levels
  \(\rightarrow\) Abstracted as thresholds or **discrete levels**
Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]

- **Unknown real values of concentrations or continuous activity levels**
  - Abstracted as thresholds or **discrete levels**

- **Continuous variations of the real values**
  - **Unitary** dynamics
Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]

- **Unknown real values of concentrations or continuous activity levels**
  - Abstracted as thresholds or **discrete levels**
- **Continuous variations of the real values**
  - **Unitary** dynamics
- **Simultaneous crossings of two thresholds never occurs**
  - **Asynchronous** dynamics
Discrete Networks / Thomas Modeling

[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
Discrete Networks / Thomas Modeling

[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components \( N = \{a, b, z\} \)
- A set of discrete expression levels for each component \( z \in F^z = \{0; 2\} \)
- The set of global states \( F = F^a \times F^b \times F^z \)
Discrete Networks / Thomas Modeling

[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components \( N = \{a, b, z\} \)
- A set of discrete expression levels for each component \( z \in \mathbb{F}^z = [0; 2] \)
- The set of global states \( \mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z \)
- An evolution function for each component \( f^z : \mathbb{F} \rightarrow \mathbb{F}^z \)

\[
\begin{array}{c|c}
 b & f^a(b) \\
\hline
0 & 1 \\
1 & 0 \\
\end{array}
\quad
\begin{array}{c|c|c}
 a & b & f^b(a, b) \\
\hline
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\quad
\begin{array}{c|c|c}
 a & b & f^z(a, b) \\
\hline
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 2 \\
\end{array}
\]
Discrete Networks / Thomas Modeling

[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components \( N = \{a, b, z\} \)
- A set of discrete expression levels for each component \( z \in F^z = [0; 2] \)
- The set of global states \( F = F^a \times F^b \times F^z \)
- An evolution function for each component \( f^z : F \rightarrow F^z \)
- Signs and thresholds on the edges \( a \xrightarrow{+1} z \)

<table>
<thead>
<tr>
<th>( b )</th>
<th>( f^a(b) )</th>
<th>( a )</th>
<th>( b )</th>
<th>( f^b(a, b) )</th>
<th>( a )</th>
<th>( b )</th>
<th>( f^z(a, b) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Analysis of Thomas Modeling

The State graph is computed in a unitary and asynchronous fashion

\[ \langle a_0, b_0, z_0 \rangle \leftrightarrow \langle a_0, b_0, z_1 \rangle \leftrightarrow \langle a_0, b_0, z_2 \rangle \]
\[ \langle a_0, b_1, z_0 \rangle \rightarrow \langle a_0, b_1, z_1 \rangle \rightarrow \langle a_0, b_1, z_2 \rangle \]
\[ \langle a_1, b_0, z_0 \rangle \rightarrow \langle a_1, b_0, z_1 \rangle \rightarrow \langle a_1, b_0, z_2 \rangle \]
\[ \langle a_1, b_1, z_0 \rangle \rightarrow \langle a_1, b_1, z_1 \rangle \rightarrow \langle a_1, b_1, z_2 \rangle \]

→ **Exponential** size in the number of components
Analysis of Thomas Modeling

The State graph is computed in a unitary and asynchronous fashion

\[ \langle a_0, b_0, z_0 \rangle \leftrightarrow \langle a_0, b_0, z_1 \rangle \leftrightarrow \langle a_0, b_0, z_2 \rangle \]

\[ \langle a_1, b_0, z_0 \rangle \leftrightarrow \langle a_1, b_0, z_1 \rangle \leftrightarrow \langle a_1, b_0, z_2 \rangle \]

\[ \langle a_1, b_1, z_0 \rangle \leftrightarrow \langle a_1, b_1, z_1 \rangle \leftrightarrow \langle a_1, b_1, z_2 \rangle \]

→ Exponential size in the number of components

Some works all to link the structure of the model and some dynamic properties:

- Thomas’ conjectures (conditions for multi-stationarity or sustained oscillations)
  - Boolean case: [Remy, Ruet, Thieffry in Advances in Applied Mathematics, 2008]
  - Multivalued case: [Richard, Comet in Discrete Applied Mathematics, 2007]
Analysis of Thomas Modeling

The State graph is computed in a unitary and asynchronous fashion

\[
\langle a_0, b_0, z_0 \rangle \leftrightarrow \langle a_0, b_0, z_1 \rangle \leftrightarrow \langle a_0, b_0, z_2 \rangle \\
\langle a_0, b_1, z_0 \rangle \rightarrow \langle a_0, b_1, z_1 \rangle \leftrightarrow \langle a_0, b_1, z_2 \rangle \\
\langle a_1, b_0, z_0 \rangle \rightarrow \langle a_1, b_0, z_1 \rangle \leftrightarrow \langle a_1, b_0, z_2 \rangle \\
\langle a_1, b_1, z_0 \rangle \rightarrow \langle a_1, b_1, z_1 \rangle \rightarrow \langle a_1, b_1, z_2 \rangle
\]

→ **Exponential** size in the number of components

Some works all to link the structure of the model and some dynamic properties:

- **Thomas’ conjectures** (conditions for multi-stationarity or sustained oscillations)
  - Boolean case: [Remy, Ruet, Thieffry in *Advances in Applied Mathematics*, 2008]
  - Multivalued case: [Richard, Comet in *Discrete Applied Mathematics*, 2007]

But reachability properties require to compute the whole state graph:

Example: From the initial state \((a, b, z) = (0, 0, 0)\), is it possible to reach \(z = 2\)?

- **Temporal logics**
  - CTL: [Berlot, Comet, Richard, Guespin in *Journal of Theoretical Biology*, 2004]
  - LTL: [Ito, Izumi, Hagihara, Yonezaki in *BioInformatics and BioEngineering*, 2010]
Standard Process Hitting is:

- Well-adapted to the modeling of BRNs
- An atomistic and qualitative modeling (explicit & discrete expression levels)
- Simple but powerful dynamics (constraints on the form of actions)
Standard Process Hitting is:

- Well-adapted to the modeling of BRNs
- An **atomistic and qualitative** modeling (explicit & discrete expression levels)
- **Simple but powerful** dynamics (constraints on the form of actions)

Previously developed tools:

- **Reachability analysis** by abstract interpretation
- Fixed points enumeration
- Stochastic parameters

→ Well-adapted formalism to study **large BRNs**
Standard Process Hitting is:

- Well-adapted to the modeling of BRNs
- An atomistic and qualitative modeling (explicit & discrete expression levels)
- Simple but powerful dynamics (constraints on the form of actions)

Previously developed tools:

- Reachability analysis by abstract interpretation
- Fixed points enumeration
- Stochastic parameters

→ Well-adapted formalism to study large BRNs

Several missing features:

- Faulty representation cooperations
- Possible enrichment of the expressivity
  → Which requires to adapt the previous tools
Standard Process Hitting

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

**Sorts:** components  \( a, b, z \)
State of the Art of Modeling

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

Standard Process Hitting

**Sorts:** components $a, b, z$

**Processes:** local states / discrete expression levels $z_0, z_1, z_2$
Standard Process Hitting

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

**Sorts:** components  \( a, b, z \)

**Processes:** local states / discrete expression levels  \( z_0, z_1, z_2 \)

**States:** sets of active processes  \( \langle a_0, b_1, z_0 \rangle \)
Standard Process Hitting

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

**Sorts:** components $a, b, z$

**Processes:** local states / discrete expression levels $z_0, z_1, z_2$

**States:** sets of active processes $\langle a_0, b_1, z_0 \rangle$

**Actions:** dynamics $b_1 \rightarrow z_0 \uparrow z_1, a_0 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow z_1 \uparrow z_2$
Standard Process Hitting

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

**Sorts:** components  \( a, b, z \)

**Processes:** local states / discrete expression levels  \( z_0, z_1, z_2 \)

**States:** sets of active processes  \( \langle a_0, b_1, z_0 \rangle \)

**Actions:** dynamics  \( b_1 \rightarrow z_0 \xrightarrow{\uparrow} z_1, \quad a_0 \rightarrow a_0 \xrightarrow{\uparrow} a_1, \quad a_1 \rightarrow z_1 \xrightarrow{\uparrow} z_2 \)
**Standard Process Hitting**

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

- **Sorts**: components $a, b, z$
- **Processes**: local states / discrete expression levels $z_0, z_1, z_2$
- **States**: sets of active processes $\langle a_0, b_1, z_1 \rangle$
- **Actions**: dynamics $b_1 \rightarrow z_0 \uparrow z_1$, $a_0 \rightarrow a_0 \uparrow a_1$, $a_1 \rightarrow z_1 \uparrow z_2$
**Standard Process Hitting**

[Paulevé et al. in *Transactions on Computational Systems Biology, 2011*]

**Sorts:** components  \( a, b, z \)

**Processes:** local states / discrete expression levels  \( z_0, z_1, z_2 \)

**States:** sets of active processes  \( \langle a_1, b_1, z_1 \rangle \)

**Actions:** dynamics  \( b_1 \to z_0 \uparrow z_1, a_0 \to a_0 \uparrow a_1, a_1 \to z_1 \uparrow z_2 \)
Standard Process Hitting

[Paulevé et al. in *Transactions on Computational Systems Biology, 2011*]

**Sorts:** components  \( a, b, z \)

**Processes:** local states / discrete expression levels  \( z_0, z_1, z_2 \)

**States:** sets of active processes  \( \langle a_1, b_1, z_2 \rangle \)

**Actions:** dynamics  \( b_1 \rightarrow z_0 \uparrow z_1, a_0 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow z_1 \uparrow z_2 \)
Cooperation between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$
Cooperations

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

\[
\text{Cooperation between } a_1 \text{ and } b_1: \quad a_1 \land b_1 \rightarrow z_0 \uparrow z_1
\]
Cooperations

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

**Cooperation** between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$
Cooperations

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

**Cooperation** between $a_1$ and $b_1$: $a_1 \land b_1 \to z_0 \uparrow z_1$
*Cooperations*

[Paulevé et al. in *Transactions on Computational Systems Biology, 2011*]

**Cooperation** between \( a_1 \) and \( b_1 \): 

\[
\begin{align*}
\text{Solution: a cooperative sort } & \quad ab \\
& \quad \text{to express } \quad a_1 \land b_1
\end{align*}
\]
Cooperations

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

**Cooperation** between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$

Solution: a **cooperative sort** $ab$ to express $a_1 \land b_1$
Cooperations

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

**Cooperation** between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$

Solution: a **cooperative sort** $ab$ to express $a_1 \land b_1$
Cooperation between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$

Solution: a cooperative sort $ab$ to express $a_1 \land b_1$
Cooperations

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

**Cooperation** between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \Rightarrow z_1$

Solution: a **cooperative sort** $ab$ to express $a_1 \land b_1$

Each configuration is represented by one process $a_1 \land b_1 \Rightarrow ab_{11}$
Cooperation between $a_1$ and $b_1$: $a_1 \wedge b_1 \rightarrow z_0 \uparrow z_1$

Solution: a cooperative sort $ab$ to express $a_1 \wedge b_1$

Each configuration is represented by one process $a_1 \wedge b_1 \Rightarrow ab_{11}$
Cooperations

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

**Cooperation** between $a_1$ and $b_1$: \[ a_1 \land b_1 \rightarrow z_0 \uparrow z_1 \]

Solution: a **cooperative sort** $ab$ to express $a_1 \land b_1$

Each configuration is represented by one process $a_1 \land b_1 \Rightarrow ab_{11}$
Cooperation between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$

Solution: a cooperative sort $ab$ to express $a_1 \land b_1$

Each configuration is represented by one process $a_1 \land b_1 \Rightarrow ab_{11}$
Cooperations

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

Cooperation between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$

Solution: a cooperative sort $ab$ to express $a_1 \land b_1$

Each configuration is represented by one process $a_1 \land b_1 \Rightarrow ab_{11}$
Cooperation between $a_1$ and $b_1$: $a_1 \land b_1 \rightarrow z_0 \uparrow z_1$

Solution: a cooperative sort $ab$ to express $a_1 \land b_1$

Each configuration is represented by one process $a_1 \land b_1 \Rightarrow ab_{11}$
Approximations for the Reachability Analysis

[Paulevé et al. in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$
Approximations for the Reachability Analysis

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Check reachability properties:

« From an initial state \( s_0 \), is it possible to reach a state \( s_n \) where \( a_i \) is active? »

Approximations: \( P \) and \( Q \), built so that \( P \Rightarrow R \Rightarrow Q \)
Approximations for the Reachability Analysis

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$
Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$
Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$
Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$
Approximations for the Reachability Analysis

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$
Approximations for the Reachability Analysis

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Check reachability properties:

« From an initial state $s_0$, is it possible to reach a state $s_n$ where $a_i$ is active? »

Approximations: $P$ and $Q$, built so that $P \Rightarrow R \Rightarrow Q$

Polynomial complexity in the number of sorts
Exponential complexity in the number of processes in each sort
→ Efficient for big models with few expression levels
Standard Process Hitting

- Process Hitting with classes of priority
- Canonical Process Hitting
- Process Hitting with neutralizing edges
- Process Hitting with synchronous actions

Standard Process Hitting
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]

Pigment productions

\[
\begin{array}{c|c}
0 & \text{inactive} \\
1 & \text{production} \\
\end{array}
\]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics
Model extracted from [François et al. in Molecular Systems Biology, 2007]

Wavefront progression
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]

Enriching the Process Hitting

Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics
Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics
Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Permissiveness of the Standard Dynamics

Model extracted from [François et al. in Molecular Systems Biology, 2007]
Process Hitting with Classes of Priorities

- Process Hitting with classes of priority
  - Canonical Process Hitting
- Process Hitting with neutralizing edges
- Process Hitting with synchronous actions
- Standard Process Hitting
Addition of classes of priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable
Addition of classes of priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable

\[ \begin{array}{cccccc}
1 & 2 & 3 & \cdots & n \\
\text{highest priority} & \text{lowest priority}
\end{array} \]

\[ \begin{array}{cccccc}
a \quad 0 & 1 & 2 & \cdots & 1 \\
0 & 1 & 2 & \cdots & 1 \\
a \quad 0 & 1 & 2 & \cdots & 1 \\
\end{array} \]

\[ \rightarrow b_1 \text{ cannot be reached} \]
Addition of classes of priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable

\[a_0 \xrightarrow{1} a_1 \xrightarrow{2} b_1 \xrightarrow{1} b_0 \]

\[a \xrightarrow{1} b \xrightarrow{2} \]

\[b_1 \text{ cannot be reached} \]
Addition of classes of priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable

\[ \begin{array}{c}
   1 \\
   2 \\
   3 \\
   \vdots \\
   n \\
\end{array} \]

\[ \begin{array}{c}
   \text{highest priority} \\
   \text{lowest priority} \\
\end{array} \]

- Allow to model classes of actions with similar speeds or temporal parameters

\[ \begin{array}{c}
   A \\
   B \\
   C \\
   \vdots \\
   N \\
\end{array} \]

\[ \begin{array}{c}
   \text{instantaneous} \\
   \text{very fast} \\
   \text{very slow} \\
\end{array} \]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Process Hitting with Neutralizing Edges

- Process Hitting with classes of priority
- Canonical Process Hitting
- Process Hitting with neutralizing edges
- Process Hitting with synchronous actions
- Standard Process Hitting
Addition of Neutralizing Edges

- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to “atomistic priorities”

\[ c_0 \rightarrow d_0 \uparrow d_1 \text{ cannot be plays while } \]
\[ a_0 \rightarrow b_0 \uparrow b_1 \text{ is playable } \]
\[ \rightarrow d_1 \text{ is always reached after } b_1 \]
Addition of Neutralizing Edges

- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to “atomistic priorities”

\[ c_0 \rightarrow d_0 \swarrow d_1 \text{ cannot be plays while} \]
\[ a_0 \rightarrow b_0 \swarrow b_1 \text{ is playable} \]
\[ \rightarrow d_1 \text{ is always reached after } b_1 \]
Addition of Neutralizing Edges

- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to “atomistic priorities”

\[ c_0 \rightarrow d_0 \uparrow d_1 \text{ cannot be played while } \]
\[ a_0 \rightarrow b_0 \uparrow b_1 \text{ is playable } \]
\[ \rightarrow d_1 \text{ is always reached after } b_1 \]
Use of Neutralizing Edges
Process Hitting with Synchronous Actions

- Process Hitting with classes of priority
- Process Hitting with neutralizing edges
- Process Hitting with synchronous actions
- Standard Process Hitting
Addition of Synchronous Actions

- Synchronizations between actions:
  - All catalysts must be present
  - Reactants are consumed all together
  - Simultaneous creation of the products

- Representation of biochemical equations:
  \[ X \xrightarrow{Y} Z \]
  under the form:
  \[ h_2 = \{x_1, y_1, z_0\} \rightarrow \{x_0, z_1\} \]

All processes of \(A\) must be present to play \(A \rightarrow B\)

After the play of \(A \rightarrow B\), all processes of \(B\) are present
Addition of Synchronous Actions

- Synchronizations between actions:
  - All catalysts must be present
  - Reactants are consumed all together
  - Simultaneous creation of the products

- Representation of biochemical equations:

  \[ X \xrightarrow{Y} Z \]

  under the form:

  \[ h_2 = \{x_1, y_1, z_0\} \rightarrow \{x_0, z_1\} \]

  All processes of \( A \) must be present to play \( A \rightarrow B \).
  After the play of \( A \rightarrow B \), all processes of \( B \) are present.
Addition of Synchronous Actions

- Synchronizations between actions:
  - All catalysts must be present
  - Reactants are consumed all together
  - Simultaneous creation of the products

- Representation of biochemical equations:

\[ X \xrightarrow{Y} Z \]

under the form:

\[ h_2 = \{x_1, y_1, z_0\} \mapsto \{x_0, z_1\} \]

All processes of A must be present to play \( A \rightarrow B \)

After the play of \( A \rightarrow B \),
all processes of B are present
Use of Synchronous Actions

→ Same dynamics than classes of priorities, except for of the missing cooperative sort
Canonical Process Hitting

- Process Hitting with classes of priority
  - Canonical Process Hitting
- Process Hitting with neutralizing edges
- Process Hitting with synchronous actions
- Standard Process Hitting
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

**Drawback**: the cooperative sorts are too “loose” (temporal shift)
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)
**Temporal Shift in Cooperative Sorts**

[Folschette et al. in *Workshop on Interactions between Computer Science and Biology*, 2013]

**Drawback:** the cooperative sorts are too “loose” (temporal shift)
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)

\[ \langle a_0, b_0, ab_{00}, z_0 \rangle \]
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)

\[ \langle a_0, b_0, ab_{00}, z_0 \rangle \]
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)

\[ \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \]
Temporal Shift in Cooperative Sorts

[Folschette et al., in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)

\[ \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \]
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)

\[
\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle
\]
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

**Drawback:** the cooperative sorts are too “loose” (temporal shift)

\[
\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle \\
\rightarrow \langle a_0, b_1, ab_{10}, z_0 \rangle
\]
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

**Drawback:** the cooperative sorts are too “loose” (temporal shift)

\[
\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle \\
\rightarrow \langle a_0, b_1, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{11}, z_0 \rangle
\]
Temporal Shift in Cooperative Sorts

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Drawback: the cooperative sorts are too “loose” (temporal shift)

\[
\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle \\
\rightarrow \langle a_0, b_1, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{11}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{11}, z_1 \rangle
\]

Expected behavior: \( a_1 \land b_1 \) simultaneously i.e. “in the same state”

Obtained behavior: \( P(a_1) \land P(b_1) \) with \( P = \) “previously”
Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Primary actions (updating cooperative sorts) → 1
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → 2
  biological / controllable actions / with delays
Primary actions (updating cooperative sorts) → 1
- non-biological / non-controllable actions

Secondary actions (all the other ones) → 2
- biological / controllable actions / with delays

Whenever a secondary action is played, all cooperative sorts are already updated
Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Primary actions (updating cooperative sorts) $→ 1$
  non-biological / non-controllable actions
- Secondary actions (all the other ones) $→ 2$
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle$
Primary actions (updating cooperative sorts) → 1
non-biological / non-controllable actions

Secondary actions (all the other ones) → 2
biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

\[ \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \]
Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Primary actions (updating cooperative sorts) $\rightarrow$ 1
  non-biological / non-controllable actions

- Secondary actions (all the other ones) $\rightarrow$ 2
  biological / controllable actions / with delays

$\Rightarrow$ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle$
Primary actions (updating cooperative sorts) $\rightarrow 1$

- non-biological / non-controllable actions

Secondary actions (all the other ones) $\rightarrow 2$

- biological / controllable actions / with delays

$\Rightarrow$ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle$
Primary actions (updating cooperative sorts) → Primary actions (updating cooperative sorts) → non-biological / non-controllable actions

Secondary actions (all the other ones) → Secondary actions (all the other ones) → biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

\[ \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{00}, z_0 \rangle \]
Primary actions (updating cooperative sorts) \(\rightarrow 1\)
non-biological / non-controllable actions

Secondary actions (all the other ones) \(\rightarrow 2\)
biological / controllable actions / with delays

\(\Rightarrow\) Whenever a secondary action is played, all cooperative sorts are already updated

\[\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{00}, z_0 \rangle\]
Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

- Primary actions (updating cooperative sorts) → ① non-biological / non-controllable actions
- Secondary actions (all the other ones) → ② biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

\[
\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle \\
\rightarrow \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{00}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{01}, z_0 \rangle
\]
Adding priorities restricts the possible dynamics (preemptions)
→ Invalidates the previous under-approximation
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation
Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation
Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Adding priorities restricts the possible dynamics (preemptions)
→ Invalidates the previous under-approximation

Similar complexity for a more expressive formalism
→ Still efficient for big models
→ Finer under-approximation
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:
• no cycle
• each objective has a solution
• cooperations are coherent

Required process

Objective

Solution to an objective

Solution to a cooperative sort process
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:
- no cycle
- each objective has a solution
- cooperations are coherent
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:
- no cycle
- each objective has a solution
- cooperations are coherent

Required process
Objective
Solution to an objective
Solution to a cooperative sort process
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

**Sufficient condition:**

- no cycle
- each objective has a solution
- cooperations are coherent

![Diagram of process hitting and analysis]

- Required process: $z_1$
- Objective: $z_0 \xrightarrow{\text{hit}} z_1$
- Solution to an objective: $\bigcirc$
- Solution to a cooperative sort process: $ab_{11}$
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

**Sufficient condition:**

- no cycle
- each objective has a solution
- cooperations are coherent
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

**Sufficient condition:**

- no cycle
- each objective has a solution
- cooperations are coherent

![Diagram of canonical process hitting](image-url)
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

\[
\begin{align*}
  z_1 & \rightarrow z_0 \xrightarrow{\ast} z_1 \\
  a_1 \xrightarrow{\ast} a_1 & \rightarrow \emptyset \\
  b_1 \xrightarrow{\ast} b_0 & \rightarrow \emptyset \\
  b_1 & \xrightarrow{\ast} b_1 \\
  a_0 & \xrightarrow{\ast} a_0 \\
\end{align*}
\]
Static Analysis of Canonical Process Hitting

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

\[ z_1 \rightarrow z_0 \xrightarrow{\ast} z_1 \rightarrow ab_{11} \]

Non-conclusive

\[ a_1 \xrightarrow{\ast} a_1 \rightarrow \bigcirc \quad b_1 \xrightarrow{\ast} b_0 \rightarrow \bigcirc \]

\[ a_1 \rightarrow a_0 \xrightarrow{\ast} a_1 \rightarrow \bigcirc \quad b_0 \rightarrow b_0 \xrightarrow{\ast} b_0 \rightarrow \bigcirc \]

\[ b_1 \xrightarrow{\ast} b_1 \rightarrow \bigcirc \quad a_0 \rightarrow a_0 \xrightarrow{\ast} a_0 \rightarrow \bigcirc \]

Required process

Objective

Solution to an objective

Solution to a cooperative sort process
Implementation of the Static Analysis Into PINT

Complexity:

- Computation of the local causality graph:
  - Polynomial in the number of sorts
  - Exponential in the number of processes of each sort
- Analysis of the graph (sufficient condition):
  - Polynomial in the size of the graph
Implementation of the Static Analysis Into PINT

 Complexity:

- Computation of the local causality graph:
  - Polynomial in the number of sorts
  - Exponential in the number of processes of each sort
- Analysis of the graph (sufficient condition):
  - Polynomial in the size of the graph

Makes the study of large networks tractable:

<table>
<thead>
<tr>
<th>Modèle</th>
<th>Sortes</th>
<th>Processus</th>
<th>Actions</th>
<th>Êtats</th>
<th>libddd$^1$</th>
<th>GINsim$^2$</th>
<th>PINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>egfr20</td>
<td>35</td>
<td>196</td>
<td>670</td>
<td>$2^{64}$</td>
<td>&lt;1s</td>
<td>0.02s</td>
<td></td>
</tr>
<tr>
<td>tcrsig40</td>
<td>54</td>
<td>156</td>
<td>301</td>
<td>$2^{73}$</td>
<td>$\infty$</td>
<td>0.02s</td>
<td></td>
</tr>
<tr>
<td>tcrsig94</td>
<td>133</td>
<td>448</td>
<td>1124</td>
<td>$2^{194}$</td>
<td>[13min – $\infty$]</td>
<td>0.03s</td>
<td></td>
</tr>
<tr>
<td>egfr104</td>
<td>193</td>
<td>748</td>
<td>2356</td>
<td>$2^{320}$</td>
<td></td>
<td>0.16s</td>
<td></td>
</tr>
</tbody>
</table>

$^1$ LIP6/Move [Couvreur et al., Lecture Notes in Computer Science, 2002]
$^2$ TAGC/IGC [Chaouiya, Naldi, Thieffry, Methods in Molecular Biology, 2012]

**egfr20**: Epithelial Growth Factor Receptor (20 components) [Sahin et al., 2009]
**egfr104**: Epithelial Growth Factor Receptor (104 components) [Samaga et al., 2009]
**tcrsig40**: T-Cell Receptor (40 composants) [Klamt et al., 2006]
**tcrsig94**: T-Cell Receptor (94 composants) [Saez-Rodriguez et al., 2007]
Formal Translation Into Canonical Form

Same dynamics (with supplemental cooperative sorts)

The canonical form can be computed for all Process Hitting extensions, with classes of priorities, neutralizing edges or synchronous actions.
Formal Translation Into Canonical Form
Formal Translation Into Canonical Form

The canonical form can be computed for all Process Hitting extensions, with classes of priorities, neutralizing edges or synchronous actions.
Formal Translation Into Canonical Form

The canonical form can be computed for all Process Hitting extensions, with classes of priorities, neutralizing edges or synchronous actions.
Formal Translation Into Canonical Form

The canonical form can be computed for all Process Hitting extensions, with classes of priorities, neutralizing edges or synchronous actions.
Formal Translation Into Canonical Form

→ Same dynamics (with supplemental cooperative sorts)

→ The canonical form can be computed for all Process Hitting extensions, with classes of priorities, neutralizing edges or synchronous actions
All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation
All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation
All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation
All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation
All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation
Translation From and To Other Discrete Models

- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham
Translation From and To Other Discrete Models

- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham
Translation From and To Other Discrete Models

- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham
Translation From and To Other Discrete Models

- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham
Translation From and To Other Discrete Models

- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham
Translation To Thomas Modeling

[Folschette et al. in *Computational Methods in Systems Biology*, 2012]

- Two successive inferences: 1) interaction graph; 2) parameters
- Exhaustive analysis of the local dynamics for each regulator
- Enumeration of all parametrizations compatible with the dynamics

Complexity:
Linear in the number of genes,
Exponential in the number of regulators of one component
Translation To Thomas Modeling

[Folschette et al. in Computational Methods in Systems Biology, 2012]

- Two successive inferences: 1) interaction graph; 2) parameters
- Exhaustive analysis of the local dynamics for each regulator
- enumeration of all parametrizations compatible with the dynamics

Complexity:
Linear in the number of genes,
Exponential in the number of regulators of one component

<table>
<thead>
<tr>
<th>Name</th>
<th>Sorts</th>
<th>Processes</th>
<th>Actions</th>
<th>Inference of the IG</th>
<th>Inference of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>egfr20</td>
<td>22</td>
<td>152</td>
<td>399</td>
<td>1s</td>
<td>1s</td>
</tr>
<tr>
<td>tcrsig40</td>
<td>14</td>
<td>156</td>
<td>301</td>
<td>1s</td>
<td>1s</td>
</tr>
<tr>
<td>tcrsig94</td>
<td>39</td>
<td>448</td>
<td>1124</td>
<td>13s</td>
<td>∞</td>
</tr>
<tr>
<td>egfr104</td>
<td>89</td>
<td>748</td>
<td>2356</td>
<td>4min</td>
<td>1min 30s</td>
</tr>
</tbody>
</table>

egfr20 : Epithelial Growth Factor Receptor (20 components) [Sahin et al., 2009]
egfr104 : Epithelial Growth Factor Receptor (104 components) [Samaga et al., 2009]
tcrsig40 : T-Cell Receptor (40 composants) [Klamt et al., 2006]
tcrsig94 : T-Cell Receptor (94 composants) [Saez-Rodriguez et al., 2007]
General Conclusion

Standard Process Hitting allows to represent biological regulatory networks in an **atomistic** fashion:

- Existing efficient static analysis
- But temporal shift issues
- Limited modeling power

**Extensions of the Process Hitting** to improve the expressivity:

- Rectification of the temporal shift → Strictly higher expressivity
- Allows to abstract temporal parameters
- New links to other formalisms (Thomas, PN, etc.)

**Static analysis** of the Canonical Process Hitting:

- Efficient analysis of reachability properties
- Applicable to the extensions at the cost of a translation
- New kind of property: simultaneous activation
Outlooks

New **exploitation** possibilities:
- Modeling and analysis of full databases
- Study of uncontrollable behaviors or punctual perturbations
- Research of interesting properties (attractors, oscillations, ...)

Improvement of the static analysis:
- Refining in order to reduce the non-conclusiveness
- New methods using by-products such as the local causality graph
- New properties to check (temporal logic, counters, ...)

Enrichment of the **modeling power**:
- Dynamical classes of priorities
- Guarded actions or complex logic gates
- New model checking tools (Hoare logic, ...)

Maxime FOLSCHETTE 34/37 PhD defense — 2014/10/08
Collaborations

Participation to the **ANR blanc** project **BioTempo** (March 2011 – November 2014):

“Language, time representations and hybrid models for the analysis of incomplete models in molecular biology”

Task 3: Introduce synchronization and continuous time in chronological models: programming language, multi-clocks and hybrid systems

3 months PhD internship (March – May 2012):
**National Institute of Informatics** (Tokyo, Japan)
Invited in the team of **Katsumi Inoue**

“Automated Reasoning and Hypothesis Finding for Systems Biology”

Partnership organized with AtlanSTIC Financial participation of Centrale Initiatives
Personal Contributions

Book chapter:

Conferences and workshops:

Current journal submissions:
- Folschette, Paulevé, Magnin, Roux: *Sufficient Conditions for Reachability in Automata Networks with Priorities*, soumis à un numéro spécial de *Theoretical Computer Science*
- Folschette, Paulevé, Inoue, Magnin, Roux: *Constructing Biological Regulatory Networks from Process Hitting models*, en cours de révision pour *Theoretical Computer Science*
- Paulevé, Folschette, Magnin, Roux: *Analyses statiques de la dynamique des réseaux d’automates indéterministes*, soumis à un numéro spécial de *Technique et Science Informatiques*
Thank you for your attention


Bibliographie


Stochastic Parameters

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]

- Introduction of temporal properties
- Stochastic parameters \((r, sa)\) equivalent to a firing interval \([d; D]\)

![Diagrams showing firing intervals](image-url)
Stochastic Parameters

[Introduction of Stochastic Parameters][1]

- Introduction of temporal properties
- Stochastic parameters \((r, sa)\) equivalent to a **firing interval** \([d; D]\)

![Diagram](https://via.placeholder.com/150)

- Simulation → not formal
- **Model-checking** → High complexity for an acceptable precision

---

[1]: Paulevé et al. in *Transactions on Computational Systems Biology, 2011*
Use of Stochastic Parameters

[Paulevé et al. in Transactions on Computational Systems Biology, 2011]
Simulation with stochastic parameters:

- Other possible analysis: stochastic model checkers (PRISM)
  - But combinatoric explosion: PRISM fails for more than 5 components
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Introduction of Stochastic Parameters

Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Introduction of Stochastic Parameters

Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Use of Classes of Priorities

[Folschette et al. in Workshop on Interactions between Computer Science and Biology, 2013]
Static analysis: successive reachability

[Paulevé et al. in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:

Initial state \( \langle a_1, b_0, c_0, d_0 \rangle \)
Static analysis: successive reachability

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Successive reachability of processes:

- Initial state
  \[\langle a_1, b_0, c_0, d_0 \rangle\]

- Objectives
  \[\langle \uparrow d_1 : \uparrow d_2 \rangle\]
Static analysis: successive reachability
[Paulevé et al. in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:

- Initial state
  \[ \langle a_1, b_0, c_0, d_0 \rangle \]

- Objectives
  \[ [ \overset{\rightharpoonup}{d_1} :: \overset{\rightharpoonup}{d_2} ] \]
  \[ [ \overset{\rightharpoonup}{d_1} :: \overset{\rightharpoonup}{b_1} :: \overset{\rightharpoonup}{d_2} ] \]
Static analysis: successive reachability

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Successive reachability of processes:

- Initial state
- Objectives

\[ \langle a_1, b_0, c_0, d_0 \rangle \]
\[ \{ \overset{1}{\rightarrow} d_1 \vdash \overset{1}{\rightarrow} d_2 \} \]
\[ \{ \overset{1}{\rightarrow} d_1 \vdash \overset{1}{\rightarrow} b_1 \vdash \overset{1}{\rightarrow} d_2 \} \]
\[ \{ \overset{1}{\rightarrow} d_2 \} \]
Static analysis: successive reachability

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Successive reachability of processes:

- Initial state

\[
\langle a_1, b_0, c_0, d_0 \rangle
\]

- Objectives

\[
\begin{align*}
[r^* d_1 &:: r^* d_2 ] \\
[r^* d_1 &:: r^* b_1 :: r^* d_2 ] \\
[r^* d_2 ]
\end{align*}
\]

→ Concretization of the objective = scenario

\[
\begin{align*}
 a_0 &\rightarrow c_0 \xrightarrow{r^*} c_1 :: b_0 \rightarrow d_0 \xrightarrow{r^*} d_1 :: c_1 \rightarrow b_0 \xrightarrow{r^*} b_1 :: b_1 \rightarrow d_1 \xrightarrow{r^*} d_2
\end{align*}
\]
Static analysis: successive reachability

[Paulevé et al. in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:

- **Initial state**
  \[ \langle a_1, b_0, c_0, d_0 \rangle \]

- **Objectives**
  \[
  \begin{align*}
  & \uparrow d_1 :: \uparrow d_2 \\
  & \uparrow d_1 :: \uparrow b_1 :: \uparrow d_2 \\
  & \uparrow d_2 
  \end{align*}
  \]

→ **Concretization of the objective = scenario**

\[ a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2 \]
Static analysis: successive reachability

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Successive reachability of processes:

- Initial state
- Objectives

\[ \langle a_1, b_0, c_0, d_0 \rangle \]

\[ [ \uparrow d_1 :: \uparrow d_2 ] \]

\[ [ \uparrow d_1 :: \uparrow b_1 :: \uparrow d_2 ] \]

\[ [ \uparrow d_2 ] \]

→ Concretization of the objective = scenario

\[ a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2 \]
Static analysis: successive reachability

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Successive reachability of processes:

- Initial state
  \[ \langle a_1, b_0, c_0, d_0 \rangle \]

- Objectives
  \[
  \begin{align*}
    \langle \uparrow d_1 \uparrow d_2 \rangle \\
    \langle \uparrow d_1 \uparrow b_1 \uparrow d_2 \rangle \\
    \langle \uparrow d_2 \rangle
  \end{align*}
  \]

→ Concretization of the objective = scenario

\[
 a_0 \rightarrow c_0 \uparrow c_1 \rightarrow b_0 \uparrow d_0 \uparrow d_1 \rightarrow c_1 \rightarrow b_0 \uparrow b_1 \rightarrow d_1 \uparrow d_2
\]
Static analysis: successive reachability

[Paulevé et al. in Mathematical Structures in Computer Science, 2012]

Successive reachability of processes:

- Initial state
- Objectives

\[ \langle a_1, b_0, c_0, d_0 \rangle \]

\[
\begin{align*}
\{ \overset{\Rightarrow}{d_1} & : \overset{\Rightarrow}{d_2} \} \\
\{ \overset{\Rightarrow}{d_1} & : \overset{\Rightarrow}{b_1} : \overset{\Rightarrow}{d_2} \} \\
\{ \overset{\Rightarrow}{d_2} \}
\end{align*}
\]

→ Concretization of the objective = scenario

\[ a_0 \rightarrow c_0 \overset{\Rightarrow}{c_1} : b_0 \rightarrow d_0 \overset{\Rightarrow}{d_1} : c_1 \rightarrow b_0 \overset{\Rightarrow}{b_1} : b_1 \rightarrow d_1 \overset{\Rightarrow}{d_2} \]
**Under-approximation**

**Sufficient condition:**
- no cycle
- each objective has a solution
Under-approximation

**Sufficient condition:**

- no cycle
- each objective has a solution

*R is true*
Sufficient condition:
- no cycle
- each objective has a solution
Under-approximation

Sufficient condition:
- no cycle
- each objective has a solution

Inconclusive
Over-approximation

Necessary condition:

\[
\begin{align*}
&d_2 \\
&d_1 \not\rightarrow^* d_2 \\
&b_2 \rightarrow b_0 \not\rightarrow^* b_2 \rightarrow d_1 \rightarrow d_1 \not\rightarrow^* d_1 \\
&b_1 \rightarrow b_0 \not\rightarrow^* b_1 \rightarrow c_1 \rightarrow c_0 \not\rightarrow^* c_1 \rightarrow a_0 \rightarrow a_1 \not\rightarrow^* a_0 \downarrow
\end{align*}
\]
Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Necessary condition:

There exists a traversal with no cycle

- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Over-approximation

**Necessary condition:**

There exists a traversal with no cycle

- objective → follow one solution
- solution → follow all processes
- process → follow all objectives

*R is false*
Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives

Inconclusive
**Static Analysis: Fixed Points**

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

**Fixed point** = state where no action can be fired

→ avoid couples of processes bounded by an action
**Fixed point** = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph
**Static Analysis: Fixed Points**

[Paulevé et al. in *Transactions on Computational Systems Biology, 2011*]

**Fixed point** = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points
**Static Analysis: Fixed Points**

[Paulevé et al. in *Transactions on Computational Systems Biology*, 2011]

**Fixed point** = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points
**Static Analysis: Fixed Points**

[Paulevé et al. in *Transactions on Computational Systems Biology, 2011*]

**Fixed point** = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points
**Static Analysis: Fixed Points**

[Paulevé et al. in *Transactions on Computational Systems Biology, 2011*]

**Fixed point** = state where no action can be fired

→ avoid couples of processes bounded by an action
→ Hitless Graph → **n-cliques** = fixed points

Exponential complexity w.r.t. the number of sorts