Hybrid and Algebraic Modeling of Biological Systems for the Analysis of their Timing Features

Olivier Roux

IRCCyN (UMR CNRS 6597) - Nantes (FRANCE)
olivier.roux@irccyn.ec-nantes.fr

joint work with Jamil Ahmad, Morgan Magnin, Loïc Paulevé

— Luxembourg, april 25, 2012 —
Part:

Introduction - Motivations
Overview of the part: Introduction - Motivations

1 Introduction
Introduction

Biological Systems Modeling

Model $\leadsto$ abstract formal system ...

Different abstraction levels...

- boolean, integer, ...
- chronologic, ...
- chronometric $\leadsto$ best knowledge of the behaviors

$\leadsto$ simulations, model checking, ...

Olivier Roux (IRCCyN)
Biological Systems Modeling

- Model $\leadsto$ abstract formal system ...
- Different abstraction levels...
  - boolean, integer, ...
  - chronologic, ...
  - chronometric $\leadsto$ best knowledge of the behaviors $\leadsto$ simulations, model checking, ...

Olivier Roux (IRCCyN)

Hybrid and Algebraic Modeling of Biological Systems for the Analysis
Model $\leadsto$ *abstract* formal system ... $\leadsto$ simulations, model checking, ...

Different *abstraction* levels...

- boolean, integer, ...
- chronologic, ...
- *chronometric* $\leadsto$ best knowledge of the behaviors
Ultimately, the systems we specify are physical objects, and mathematics cannot prove physical properties. We can prove properties only of a mathematical model of the system; whether or not the system correctly implements the model must remain a question of law and not of mathematics.

*Leslie Lamport*
Comm. ACM, 1989
activation and inhibition of genes (through the production of proteins)

\[ f_i(x) = k_i + \sum_{j \in L_i} k_{ij} b_{ij}(x) \]
activation and inhibition of genes (through the production of proteins)

~ Model checking biologic Regulatory Networks

▶ understand (modeling)
▶ verify (analysis, simulation and tests, model-checking)
▶ remedy... (control)
Part:

Hybrid Modeling of Biological Regulatory Networks
Overview of the part: Hybrid Modeling of Biological Regulatory Networks

2. Discrete Modeling

3. Hybrid Modeling

4. Discussion and Refinement
Figure 1: Premiers paramètres discrets

Figure 2: Exemple: Pseudomonas æruginosa
Figure 3 : Premiers paramètres discrets

L'inconvénient de l'approche booléen est son adaptation aux systèmes biologiques. Cela est dû au fait que chaque état de RRB n'a qu'un seul successeur, et donc pas de choix de différents chemins. Ce problème a été ensuite abordé par René Thomas par le développement d'un formalisme logique asynchrone [Tho73, Tho78, Tho79, Tho83]. Dans une forme très naïve, le formalisme René Thomas décrit l'état d'une entité biologique d'une manière booléen [Tho73]. Les travaux suivants de même auteur introduisent des variables de plus de deux niveaux. Ce formalisme suppose que, si une variable agit sur plus d'une variable, les seuils de concentration pour chacune de ses actions ne sont pas égaux. Nous donnons un exemple sur la Figure [3].

La variable \( u \) a un effet positif sur le gène \( v \) quand il atteint sa concentration \( \theta_{uv} \) et un effet négatif sur le gène \( z \) à \( \theta_{uz} \). La relation \( \theta_{uv} < \theta_{uz} \) permet d'obtenir le discrétisation de la concentration de \( u \) dans trois niveaux 0, 1 et 2. Les valeurs du niveau qualitatif sont appelées les valeurs logiques. L'utilisation des valeurs logiques évite la nécessité de définir les valeurs d'un grand nombre de paramètres logiques (Cf. Section 2).

Le formalisme de René Thomas permet une mise à jour des variables d'une manière logique sophistiquée. Ce formalisme introduit les paramètres logiques [Sno89] et les valeurs logiques pour les seuils. Ces arrangements permettent d'avoir les différents comportements, par exemple état stable et les cycles [Sno89].

Le temps pris en compte dans ce formalisme est considéré continue. Ce qui permet une mise à jour asynchrone des variables [Tho91]. Un ensemble des fonctions qui décrit les états et leurs cibles est défini par ce formalisme.
States, Transitions

Multivalued Regulatory Graphs

\[ x \quad y \quad z \]

\[ x \quad 1 \quad y \quad z \]

\[ x \quad 2 \quad y \quad z \]

\[ y' \quad z' \]

\[ 0 \quad 1 \quad 2 \]

Olivier Roux (IRCCyN)
States, Transitions

**Figure 5**: A sigmoid (a), and its logical caricature (b)

(abstract states) 0, 1, 2 \ldots  of each component

\[ \Rightarrow \]

global state as a tuple.
Under the hypothesis of knowing some parameters values, we have:

- (0,0) to (1,0)
- (1,0) to (2,0)
- (2,0) to (0,1)
- (0,1) to (1,1)
- (1,1) to (2,1)
- (2,1) to (0,0)
And while desynchronizing with unit of the “Manhattan distance”, we get:
Discrete State Graph

Hence, trajectories: ...
Discrete State Graph

Hence, trajectories: ...
**Figure 6**: Pseudomonas ëruginosa (cont’): Behaviors and ... need for improvement

The discrete state graph and refining are used to illustrate the behavior of the system. The graph shows the transitions between different states, represented by points (0, 0), (1, 1), (2, 1), (0, 1), (1, 0), and (2, 0). The arrows indicate the transitions between these states. The graph also shows a cycle: (0, 0) → (1, 0) → (2, 0) → (0, 0), indicating the system's behavior over time.
Overview of the part: Hybrid Modeling of Biological Regulatory Networks

2. Discrete Modeling

3. Hybrid Modeling

4. Discussion and Refinement
Figure 7: A sigmoid (a), its logical caricature (b) and its piecewise linear caricature (c)
Relative Values of durations and delays et retards infer actual behaviors

Figure 8: Example of the Escherichia coli bacteria

Ropers et al. 2007

Olivier Roux (IRCCyN)
Introduction aux “Automates hybrides”

des horloges pour mesurer les délais de franchissement de seuil

Figure 9 : modélisation des délais par des “horloges”
Related works

- **Automates temporisés et hybrides**
  - [H. Sieber and A. Bockmayr]. "Incorporating Time Delays into the Logical Analysis of Gene Regulatory Networks". In *Computational Methods in Systems Biology, (CMSB 2006),*
  - [J. Ahmad, G. Bernot, J.-P. Comet, D. Lime and O. Roux]. "Hybrid modelling and dynamical analysis of gene regulatory networks with delays". *ComPlexUs, 3(4):231-251, 2006 (Cover Date: November 2007)*
et aussi:


- [G. Batt, C. Belta and R. Weiss]. "Model checking liveness properties of genetic regulatory networks". In *Tools and algorithms for the construction and analysis of systems (TACAS’07)*, LNCS, 2007. [BBW07b]


- [A. Aswani and C. Tomlin]. "Reachability algorithm for biological piecewise-affine hybrid systems". In *Hybrid systems (HSCC’07)*, LNCS, 2007. [AT07]
Escherichia coli

- a bacterium detecting the food concentration through a set of receptors;
- moving by flagellar rotations.

Depending on the concentration of attractants and repellents, E. coli responds to stimuli in one of two ways:

- “RUNS” – it moves in a straight line by moving its flagella counterclockwise (CCW)
- “TUMBLES” – it randomly changes its heading by moving its flagella clockwise (CW)
E. coli IDA Model

RUN [CCW]

\[ \begin{align*}
\omega &= -1 \\
\dot{Y}_P &= k_y P(Y_0 - Y_P) - k_{-y} Z Y_P \\
\dot{B}_P &= k_b P(B_0 - B_P) - k_{-b} B_P \\
P &= LT_{2p} + LT_{3p} + LT_{4p} + T_{2p} + T_{3p} + T_{4p}
\end{align*} \]

TUMBLE [CW]

\[ \begin{align*}
\omega &= +1 \\
\dot{Y}_P &= k_y P(Y_0 - Y_P) - k_{-y} Z Y_P \\
\dot{B}_P &= k_b P(B_0 - B_P) - k_{-b} B_P \\
P &= LT_{2p} + LT_{3p} + LT_{4p} + T_{2p} + T_{3p} + T_{4p}
\end{align*} \]

A. Casagrande et al., *Independent Dynamics Hybrid Automata in Systems Biology*, AB’05 Tokyo, 2005
Hybrid Modeling
Temporal Zones

Temporal Zones  for biological systems modeling

Figure 11: Example of a discrete model of a GRN (again)

Principle: Expanding each discrete state in a zone (symbolic region):
zoom on a discrete state (1,0)...

Olivier Roux (IRCCyN)
Hybrid and Algebraic Modeling of Biological Systems for the Analysis of their Timing Features
Temporal Zones for biological systems modeling

Figure 12: Example of a discrete model of a GRN (again)

Principle:
Expanding each discrete state in a zone (symbolic region): zoom on a discrete state (1,0)...
Expanding a state such as \((1,0)\) and introducing the clocks

Expansion: \(x = 1\) stands for a segment of length: \(|d_x^-| + |d_x^+|\)

- when \(x\) increases, \(h_x\) evolves from 0 to \(d_x^+\);
- when \(x\) decreases, \(h_x\) evolves from 0 to \(d_x^-\).
Temporal Zones for biological systems modeling (cont’)

Figure 13: Expansion de la localité (1, 0) en une zone temporelle: \((b_1, b_2)\)
Temporal Zones for biological systems modeling cont’)

Executions consistent with the dynamics (in green)

Figure 14: Example of the “temporal” modeling of a GRN (partial view)
Temporal Zones for biological systems modeling (cont’)

Two kinds of steps in this hybrid modeling:
- Continuous step: time elapsing
- Discrete step: transition towards the next location
Space of the temporal zones

Figure 15: Full view of the temporal modeling
Hybrid Model of the example

Figure 16: Pseudomonas æruginosa

(bold arrows stand for the transitions of the discrete model)
Hybrid Modeling

Trajectories

Sequence of alternating continuous and discrete transitions
Analysis, first verification results and discussion

- Evolution (analysis of the trajectories)
- Results
- Limitations
Definition (Invariant vs Divergent Trajectories)

The largest set of points such that each trajectory starting from one of these points, ever stays in this set is the "Invariance kernel". This kernel is expressed as a constraint on the delay parameters, and is algorithmically synthesized. On the contrary, outside this kernel are the divergent trajectories leading in an "Attraction basin".
Definition (Invariant vs Divergent Trajectories)

The largest set of points such that each trajectory starting from one of these points, ever stays in this set is the "Invariance kernel". This kernel is expressed as a constraint on the delay parameters, and is algorithmically synthesized.

On the contrary, outside this kernel are the divergent trajectories leading in an "Attraction basin".

Olivier Roux (IRCCyN)
Definition (Invariant vs Divergent Trajectories)

The largest set of points such that each trajectory starting from one of these points, ever stays in this set is the “Invariance kernel”.

This kernel is expressed as a constraint on the delay parameters, and is algorithmically synthetised. On the contrary, outside this kernel are the divergent trajectories leading in an “Attraction basin”.

Olivier Roux  (IRCCyN)
Analysis (cont’)

Aims:

Among the trajectories, find:

1. Cycles: infinite behaviors
2. Attraction basins and bifurcations

Solution:

Algorithm

- Written in HyTech [HHWT97]
- Result: parameterized polyhedra
- Interpretation in PolyLib
  [http://icps.u-strasbg.fr/polylib/]
Analysis (cont’)

Aims:
Among the trajectories, find:
1. Cycles: infinite behaviors
2. Attraction basins and bifurcations

Solution:
Algorithm
- Written in HyTech [HHWT97]
- Result: parameterized polyhedra
- Interpretation in PolyLib
  [http://icps.u-strasbg.fr/polylib/]
Analysis (cont’)

Figure 17: Simple or Nested Cycles
Overview of the part: Hybrid Modeling of Biological Regulatory Networks

1. Discrete Modeling
2. Hybrid Modeling
3. Discussion and Refinement
Features of our modeling approach:

1. consistent with the one of René Thomas
   (leads to the same model if we set to zero parameters values)

2. more accurate determination of the evolutions,

3. taking accumulations into account

but

4. Not fully exact due since homothetic transformations are not considered.
Features of our modeling approach:

1. consistent with the one of René Thomas
   (leads to the same model if we set to zero parameters values)
2. more accurate determination of the evolutions,
3. taking accumulations into account

but

4. Not fully exact due since homothetic transformations are not considered.
Discussion

Features of our modeling approach:

1. consistent with the one of René Thomas
   (leads to the same model if we set to zero parameters values)
2. more accurate determination of the evolutions,
3. taking accumulations into account

but

4. Not fully exact due since homothetic transformations are not considered.
Discussion

Features of our modeling approach:

1. consistent with the one of René Thomas
   (leads to the same model if we set to zero parameters values)
2. more accurate determination of the evolutions,
3. taking accumulations into account

but

4. Not fully exact due since homothetic transformations are not considered.
Modeling Refinement . . .
Discussion and Refinement: Asymptotic Evolution towards the focal point

Figure 18: Zone partitioning into “sub-domains” (according to the position of the focal point)

Modeling Refinement:

Figure 5: Partition of a domain into subdomains. In each subdomain, the vector of derivative signs is a constant.

1. The subdomains of a domain \( d \) are the equivalence classes of the set of states under the equivalence relation \( \equiv \) defined by:

\[
(x_i)_{i \in [1,n]} \equiv (y_i)_{i \in [1,n]} \Leftrightarrow \forall i \in [1,n], \text{sign}(\dot{x}_i) = \text{sign}(\dot{y}_i)
\]

where \( \text{sign}(x) = 1 \) if \( x > 0 \), \( 0 \) if \( x = 0 \) and \( -1 \) if \( x < 0 \).

2. The temporal variation vector \( v_{sd} \) of a subdomain \( sd \) is

\[
v_{sd} = (\text{sign}(\dot{x}_i))_{i \in [1,n]}
\]

for any point \((x_i)_{i \in [1,n]} \in sd\).

In the example of Figure 5, there are three subdomains:

- The subdomain where variable \( x_2 \) stays constant (\( \dot{x}_2 = 0 \)), it is composed of the points of the horizontal dashed line,
- The subdomain where variable \( x_2 \) increases (\( \dot{x}_2 > 0 \)), it is composed of the points below the dashed line,
- The subdomain where variable \( x_2 \) decreases (\( \dot{x}_2 < 0 \)), it is composed of the points above the dashed line.

The dimension of a subdomain \( sd \) is then defined as the number of variables \( x_i \) such that \( \dot{x}_i \neq 0 \). Then, we introduced \( \text{dim}(sd) \) to these variables \( x_i \), which do not stay constant:

\[
\text{dim}(sd) = \{x_i | i \in [1,n], \text{sign}(\dot{x}_i) \neq 0\}
\]

Figure 6 shows some 3D-examples of the domain partition into subdomains.
Figure 19: Hybrid dynamics with centripetal spirals

Figure 15: Hybrid dynamics with convergent spirals. Dashed thick red line determines the frontier between divergence towards domain (2,1) and convergence.

Figure 16: Hybrid dynamics with torus (set of cyclic temporal trajectories) and its boundaries in thick red lines.
Modeling Refinement (cont’): idea and overview

Figure 20: Hybrid dynamics with torus (set of cyclic temporal trajectories) and its boundaries in thick red lines.

Figure 15: Hybrid dynamics with convergent spirals. Dashed thick red line determines the frontier between divergence towards domain $2,1$ and convergence.

Figure 16: Hybrid dynamics with torus (set of cyclic temporal trajectories) and its boundaries in thick red lines.
Figure 21: Hybrid dynamics with divergent spirals.
Figure 22: Hybrid dynamics with a limit cycle (in thick red line)
Part:

Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):

5 Basics
6 Presentation of the “Process Hitting”
7 Verification (and control)
8 Simulation and temporal properties
9 Static Analysis by abstraction
10 Conclusion
Towards an algebraic approach
(yet with timie concerns)
The \( \pi \)-calculus [R. Milner 1992]

We use only a fragment

\[
\begin{align*}
\text{Dec} & ::= \text{new x \{@r\} : t} & \text{Channel Declaration} \\
& | \text{type n = t} & \text{Type Declaration} \\
& | \text{val m = v} & \text{Value Declaration} \\
& | \text{run P} & \text{Process Declaration} \\
& | \text{let D1 and ... and DN} & \text{Definitions, } N \geq 1 \\
\text{D} & ::= X(m_1, \ldots, m_N) = P & \text{Definition, } N \geq 0 \\
\text{P} & ::= () & \text{Null Process} \\
& | (P_1 | \ldots | P_M) & \text{Parallel, } M \geq 2 \\
& | X(v_1, \ldots, v_N) & \text{Instantiation, } N \geq 0 \\
& | a\{; P\} & \text{Action} \\
& | \text{do a}_1\{; P_1\} \text{ or } \ldots \text{ or } a_M\{; P_M\} & \text{Choice, } M \geq 2 \\
& | (\text{Dec}_1 \ldots \text{Dec}_N P) & \text{Declarations, } N \geq 0 \\
\text{aI} & ::= !x\{(v_1, \ldots, v_N)\} & \text{Output, } N \geq 0 \\
& | ?x\{(m_1, \ldots, m_N)\} & \text{Input, } N \geq 0 \\
& | \text{delay } @r & \text{Delay}
\end{align*}
\]

rate: \( r \sim \) mean duration: \( 1/r \), variance: \( 1/r^2 \)
The $\pi$-calculus: *Fragment*

<table>
<thead>
<tr>
<th>Action $\pi =::$</th>
<th>$!x(m)$</th>
<th>$?x(n)$</th>
<th>$\tau_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sends value $m$ on channel $x$</td>
<td>Receives value $n$ on channel $x$</td>
<td>Delay at rate $r$</td>
<td></td>
</tr>
</tbody>
</table>

| Process $P =::$ | $\pi \cdot P_1$ | $\pi_1 \cdot P_1 + \ldots + \pi_N \cdot P_N$ | $\pi \cdot (P_1 | \ldots | P_N)$ |
|-----------------|----------------|--------------------------------|--------------------------------|
| Nul process | makes an action | Chooses between actions | Parallel composition of processes |

Olivier Roux (IRCCyN)
Related works

About modeling and verification of biological systems (not exhaustive):

- "Timed and Stochastic Petri Nets (GinSim)"
  [C. Chaouiya & E. Remy & D. Thieffry] [CRT08],
- "Probabilistic Model Checking (Prism)"
  [M. Kwiatkowska & D. Parker] [HKN+08],
- "Process algebra with stochasticity (BioSpi et Spim)"
  [C. Priami & A. Regev] [PRSS01],
- "Constraints (Biocham)" [F. Fages] [RBFS08],
- "Decision and Control of Dynamical Complex Systems"
  [L. Paulevé & M. Magnin & O. Roux],
- "Cellular automata" [AKGJ02].
... and especially algebraic modeling in biology:

- **Bio-PEPA**: "Bio-PEPA: A framework for the modelling and analysis of biological systems" [F. Ciocchetta & J. Hillston] [CH09],
- **π-calcul et SPIM**: "Compositionality, Stochasticity and Cooperativity in Dynamic Models of Gene Regulation" [R. Blossey & L. Cardelli & A. Phillips] [BCP08],
- "A programming language for composable DNA circuits" [A. Phillips & L. Cardelli] [PC09].
- **CBS**: "A Language for Biochemical Systems" [M. Pedersen & G. Plotkin] [PP08],
- **Blenx**: "Modelling and simulation of biological processes in BlenX" [L. Dematté & C. Priami & A. Romanel] [DPR08],
- **BioSpi**: "Rule-Based Modelling of Cellular Signalling" [P. Lecca & C. Priami] [LP07],
- **κ-calcul et BioNetGen**: "Rule-Based Modelling of Cellular Signalling" [V. Danos & J. Feret & W. Fontana & R. Harmer & J. Krivine] [DFF+07],
- "Abstract interpretation and types for systems biology" [F. Fages & S. Soliman] [FS08].
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):

5 Basics
6 Presentation of the “Process Hitting”
7 Verification (and control)
8 Simulation and temporal properties
9 Static Analysis by abstraction
10 Conclusion
A new formalism for dynamic systems

**Idea:** split structure and dynamics...

...avoiding the whole transitions relation:

\[
\begin{align*}
 f_0 c_0 a_0 & \rightarrow f_0 c_0 a_1 \\
 f_0 c_1 a_1 & \rightarrow f_0 c_1 a_0 \\
 f_1 c_1 a_1 & \rightarrow f_1 c_1 a_0 \\
 \cdots
\end{align*}
\]

... in a set of \(2^n\) states....

n "sorts" 1 and only 1 "processus" is alive at each time
A new formalism for dynamic systems

Idea: split structure and dynamics . . .

. . . avoiding the whole transitions relation:

\[
\begin{align*}
f_0c_0a_0 & \rightarrow f_0c_0a_1 \\
\ldots
\end{align*}
\]

\[
\begin{align*}
f_0c_1a_1 & \rightarrow f_0c_1a_0 \\
\end{align*}
\]

\[
\begin{align*}
f_1c_1a_1 & \rightarrow f_1c_1a_0 \\
\ldots
\end{align*}
\]

in a set of \(2^n\) states . . .

n "sorts" 1 and only 1 "processus" is alive at each time
n ”sorts” 1 and only 1 ”processus” is alive at each time
Presentation of the “Process Hitting” Basic Elements

Hybrid and Algebraic Modeling of Biological Systems for the Analysis of their Timing Features
Presentation of the “Process Hitting” Basic Elements

Hybrid and Algebraic Modeling of Biological Systems for the Analysis of their Timing Features

Olivier Roux (IRCCyN)
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):

- Basics
- Presentation of the “Process Hitting”
- Verification (and control)
- Simulation and temporal properties
- Static Analysis by abstraction
- Conclusion
Verification of structural properties: stable states

Hypergraph of the "Process Hitting"
Verification of structural properties: stable states

Hypergraph of the "Process Hitting"
Verification of structural properties: stable states

Hypergraph of the "Process Hitting"

The $n$-cliques are the stable states: $f_0c_0a_0$
Verification of structural properties: stable states

The $n$-cliques are the stable states: $f_0c_0a_0$
Verification of structural properties: stable states

The \( n \)-cliques are the stable states: \( f_0c_0a_0 \) and \( f_0c_0a_1 \)
Verification of dynamic properties: reachability

Accessibility of $f_2$?
Verification of dynamic properties: reachability

Accessibility of $f_2$?
Verification of dynamic properties: reachability

Accessibility of \(f_2\)?
Verification of dynamic properties: reachability

Accessibility of $f_2$?
Verification of dynamic properties: reachability

$f_2$ accessible!
Verification of dynamic properties: reachability

Accessibility of $f_2$?
Verification of dynamic properties: reachability

Accessibility of \( f_2 \)?
Verification of dynamic properties: reachability

Accessibility of $f_2$?
Verification of dynamic properties: reachability

Lock: \( f_{-2} \) non accessible!
Verification of dynamic properties: control

Control of $f_{2}$-unreachability: prevent from $a_{0}$
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):

5. Basics
6. Presentation of the "Process Hitting"
7. Verification (and control)
8. Simulation and temporal properties
9. Static Analysis by abstraction
10. Conclusion
Translations

- in the π-calcul
- in SPIM ➝ simulations
- in PRISM ➝ probabilistic model-checking

and

- the inference of temporal parameters
**Stochastic Framework**

\[ h = c_0 \rightarrow a_0 \equiv \begin{cases} 
C_0 ::= \cdots +!\gamma_h \cdot C_0 \\
A_0 ::= \cdots +?\gamma_h \cdot A_1 
\end{cases} \]

- straightforward translation in the stochastic π-calculus
- to each channel \( \gamma_h \): use rate \( r_h \)
- natural introduction of stochastic parameters into the Process Hitting framework
- Gillespie: reaction duration follows an exponential law
- average duration of an action with use rate \( r \): \( \frac{1}{r} \)
Tuning stochasticity: temporal parameter synthesis

Example: *self-hitting* process:

\[ a_0 \rightarrow a_1 \]

Use rate \( r \) for each action (average duration: \( \frac{1}{r} \)).
Tuning stochasticity: temporal parameter synthesis

Example: *self-hitting* process:

Use rate $r$ for each action (average duration: $\frac{1}{r}$).

Simulation through SPIM [A. Phillips]:

Olivier Roux (IRCCyN)
Stochasticity absorption factor

“Duration follows one exponential random variable of rate $r$” becomes

“Duration follows the sum of $sa$ exponential random variables of rate $r.sa$”

Erlang distribution (particular Gamma) of shape $sa$ and rate $r.sa$:

\[ sa = 1, \quad sa = 5, \quad sa = 50 \]
Stochasticity Absorption

Use rate $r = 1$ ($m = 1, v = 1$)

- $\hat{m} = 1.144, \hat{v} = 0.764$
- $\hat{m} = 5.72, \hat{v} = 0.539$

Stochasticity absorption factor $sa = 5$ ($r = 1 \times 5$):

- $\hat{m} = 1.004, \hat{v} = 0.344$
Stochasticity Absorption

Use rate \( r = 1 \) \((m = 1, \nu = 1)\)

\[
\hat{m} = 1.144, \hat{\nu} = 0.764
\]

\[
\hat{m} = 5.72, \hat{\nu} = 0.539
\]

Stochasticity absorption factor \( sa = 5 \) \((r = 1 \times 5)\):

\[
\hat{m} = 1.004, \hat{\nu} = 0.344
\]
**Stochasticity Absorption**

Use rate $r = 1$ ($m = 1, v = 1$)

- $\hat{m} = 1.144, \hat{v} = 0.764$
- $\hat{m} = 5.72, \hat{v} = 0.539$

Stochasticity absorption factor $sa = 5$ ($r = 1 \times 5$):

- $\hat{m} = 1.004, \hat{v} = 0.344$
Stochasticity Absorption

Use rate $r = 1$ ($m = 1, \nu = 1$)

- $\hat{m} = 1.144, \hat{\nu} = 0.764$
- $\hat{m} = 5.72, \hat{\nu} = 0.539$

Stochasticity absorption factor $sa = 5$ ($r = 1 \ast 5$):

- $\hat{m} = 1.004, \hat{\nu} = 0.344$
Temporal and stochastic parameters

Example: *self-hitting* process (*again*):

Use rate $r$ for each action, stochasticity absorption factor $sa$ (average duration: $\frac{1}{r}$ unchanged).

- $sa = 1$
- $sa = 5$
- $sa = 50$
Firing intervals

Given a confidence $1 - \alpha$:

$$(r, sa) \iff$$

lower $sa$

higher $sa$
Further results: behaviour avoidance

Example: \( f_0 \rightarrow c_1 \xRightarrow{\delta} c_0 \) and \( c_1 \rightarrow a_1 \xRightarrow{\delta} a_0 \)

*Prevent a from being bounced from 1 to 0*
Further results: behaviour avoidance

Example: $f_0 \rightarrow c_1 \rightarrow c_0$ and $c_1 \rightarrow a_1 \rightarrow a_0$

Prevent $a$ from being bounced from 1 to 0
Translation in PRISM (example)

Previous example ($r_{c_1} = 0.25$, $r_{f_0} = 2 \sim \delta_{f_0} < \delta_{c_1}$):

```plaintext
c o n t i n u o u s - t i m e  m a r k o v  c h a i n s
module proc_c
  c: [0..1] init 1;
  [h_0] c=1 -> 0.25: (true);
  [h_1] c=1 -> (c'=0);
endmodule

module proc_f
  f: [0..1] init 0;
  [h_1] f=0 -> 2.0: (true);
endmodule

module proc_a
  a: [0..1] init 1;
  [h_0] a=1 -> (a'=0);
endmodule
```

Olivier Roux (IRCCyN) Hybrid and Algebraic Modeling of Biological Systems for the Analysis
PRISM model-checking and simulation

without and with stochasticity absorption factor:

Model checking: $P = \? \ [ F (a=0) ]$
Time for model checking: 0.0010 seconds.
Result (probability): 0.1111111111111111

---------------------------------
Model checking: $P = \? \ [ F (a=0) ]$
Time for model checking: 0.042 seconds.
Result (probability): 3.4632066773729244E-22

---------------------------------
Simulating: $P = \? \ [ F (a=0) ]$
Sampling complete: 402412 iterations in 3.38 seconds (average 0.000008)
Result: 0.11075464946373369

---------------------------------
Simulating: $P = \? \ [ F (a=0) ]$
Sampling complete: 402412 iterations in 125.75 seconds (average 0.000312)
Result: 0.0
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):

- Basics
- Presentation of the "Process Hitting"
- Verification (and control)
- Simulation and temporal properties
- Static Analysis by abstraction
- Conclusion
Over- and Under-Approximation

1. Over-Approximation

2. Under-Approximation

Exact solution
The Process Reachability Problem: Running Example

Is $d_2$ reachable from $d_0$ in $< a_0, b_1, c_1, d_0 >$?

E.g. initial state $< a_0, b_1, c_1, d_0 >$
The Process Reachability Problem: Running Example

Is \( d_2 \) reachable from \( d_0 \) in \( < a_0, b_1, c_1, d_0 > \)?

\[
\begin{align*}
 b_1 &\rightarrow a_0 \rightarrow a_1, \\
a_1 &\rightarrow b_1 \rightarrow b_0, \\
b_0 &\rightarrow d_0 \rightarrow d_1, \\
d_1 &\rightarrow b_0 \rightarrow b_2, \\
c_1 &\rightarrow d_1 \rightarrow d_0, \\
b_2 &\rightarrow d_0 \rightarrow d_2
\end{align*}
\]

e.g. initial state \( < a_0, b_1, c_1, d_0 > \)
The Process Reachability Problem: Running Example

Is \( d_2 \) reachable from \( d_0 \) in \( < a_0, b_1, c_1, d_0 > \)?

\[
b_1 \rightarrow a_0 \; \uparrow \; a_1 \; \uparrow \; b_1 \; \uparrow \; b_0, \; b_0 \rightarrow d_0 \; \uparrow \; d_1, \; d_1 \rightarrow b_0 \; \uparrow \; b_2, \; c_1 \rightarrow d_1 \; \uparrow \; d_0, \; b_2 \rightarrow d_0 \; \uparrow \; d_2
\]
The Process Reachability Problem: Running Example

Is $d_2$ reachable from $d_0$ in $< a_0, b_1, c_1, d_0 >$?

E.g. initial state $< a_0, b_1, c_1, d_0 >$

$b_1 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow b_1 \uparrow b_0, b_0 \rightarrow d_0 \uparrow d_1, d_1 \rightarrow b_0 \uparrow b_2, c_1 \rightarrow d_1 \uparrow d_0, b_2 \rightarrow d_0 \uparrow d_2$
The Process Reachability Problem: Running Example

Is $d_2$ reachable from $d_0$ in $< a_0, b_1, c_1, d_0 >$?

e.g. initial state $< a_0, b_1, c_1, d_0 >$

$b_1 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow b_1 \uparrow b_0, b_0 \rightarrow d_0 \uparrow d_1, d_1 \rightarrow b_0 \uparrow b_2, c_1 \rightarrow d_1 \uparrow d_0, b_2 \rightarrow d_0 \uparrow d_2$
The Process Reachability Problem: Running Example

Is $d_2$ reachable from $d_0$ in $< a_0, b_1, c_1, d_0 >$?

$\begin{align*}
  b_1 &\rightarrow a_0 \xrightarrow{1} a_1, \\
  a_1 &\rightarrow b_1 \xrightarrow{1} b_0, \\
  b_0 &\rightarrow d_0 \xrightarrow{1} d_1, \\
  d_1 &\rightarrow b_0 \xrightarrow{1} b_2, \\
  c_1 &\rightarrow d_1 \xrightarrow{1} d_0, \\
  b_2 &\rightarrow d_0 \xrightarrow{1} d_2
\end{align*}$

E.g., initial state $< a_0, b_1, c_1, d_0 >$
The Process Reachability Problem: Running Example

Is \( d_2 \) reachable from \( d_0 \) in \( < a_0, b_1, c_1, d_0 > \)?

e.g. initial state \( < a_0, b_1, c_1, d_0 > \)

\[
\begin{align*}
 b_1 &\rightarrow a_0 \uparrow a_1, \\
 a_1 &\rightarrow b_1 \uparrow b_0, \\
 b_0 &\rightarrow d_0 \uparrow d_1, \\
 d_1 &\rightarrow b_0 \uparrow b_2, \\
 c_1 &\rightarrow d_1 \uparrow d_0, \\
 b_2 &\rightarrow d_0 \uparrow d_2
\end{align*}
\]
The Process Reachability Problem: Running Example

Is $d_2$ reachable from $d_0$ in $< a_0, b_1, c_1, d_0 >$? e.g. initial state $< a_0, b_1, c_1, d_0 >$

$b_1 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow b_1 \uparrow b_0, b_0 \rightarrow d_0 \uparrow d_1, d_1 \rightarrow b_0 \uparrow b_2, c_1 \rightarrow d_1 \uparrow d_0, b_2 \rightarrow d_0 \uparrow d_2$
**Over-approximation of Process Reachability**

- Focus on *objectives* starting from the initial state;
- Add required objective redirections (not detailed);
- **Necessary condition**: there always exists a solution ending with a trivial objective.

Initial: $<a_1, b_0, c_0, d_1>$
Over-approximation of Process Reachability

- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailed);
- Necessary condition: there always exists a solution ending with a trivial objective.

Initial: \( \langle a_1, b_0, c_0, d_1 \rangle \)

Failure

Olivier Roux (IRCCyN)
Over-approximation of Process Reachability

Focus on objectives starting from the initial state;
Add required objective redirections (not detailed);
Necessary condition: there always exists a solution ending with a trivial objective.

Initial: \( \langle a_1, b_1, c_1, d_0 \rangle \)

\( d_0 \xrightarrow{\ast} d_2 \)

\( b_0 \xrightarrow{\ast} b_0 \) \( \xrightarrow{} \) \( a_1 \xrightarrow{\ast} a_1 \) \( \xrightarrow{} \)

\( b_1 \xrightarrow{\ast} b_2 \) \( \xrightarrow{} \) \( d_1 \xrightarrow{\ast} d_0 \) \( \xrightarrow{} \)

OK (inconclusive)
Under-approximation of Process Reachability

Initial: \(<a_1, b_1, c_1, d_0>\)

\[ b_0 \rightarrow b_1 \rightarrow^* b_0 \rightarrow \square \rightarrow a_1 \rightarrow a_1 \rightarrow^* a_1 \rightarrow \square \]

\[ d_0 \rightarrow^* d_2 \rightarrow \square \rightarrow b_1 \rightarrow b_1 \rightarrow^* b_1 \rightarrow \square \]
Under-approximation of Process Reachability

Initial: \( <a_1, b_1, c_1, d_0> \)

\[
\begin{align*}
b_0 & \rightarrow b_1 \rightarrow^* b_0 \\
d_0 & \rightarrow^* d_2 \\
b_1 & \rightarrow b_1 \\
b_0 & \rightarrow^* b_1 \\
c_1 & \rightarrow^* c_1
\end{align*}
\]

SUCCESS
Under-approximation of Process Reachability

Initial: \(< a_0, b_1, c_0, d_0 >\)

- \(b_0 \rightarrow b_1 \rightarrow^* b_0 \rightarrow^* a_1 \rightarrow^* a_0 \rightarrow^* a_1\)
- \(d_0 \rightarrow^* d_2\)
- \(b_1 \rightarrow^* b_1\)
Under-approximation of Process Reachability

Initial: \( \langle a_0, b_1, c_0, d_0 \rangle \)

- \( b_0 \mapsto b_1 \mapsto a_1 \mapsto a_0 \)
- \( d_0 \mapsto d_2 \)
- \( b_1 \mapsto b_1 \mapsto c_1 \mapsto a_0 \mapsto a_0 \)

Olivier Roux (IRCCyN)

Hybrid and Algebraic Modeling of Biological Systems for the Analysis of their Timing Features
Under-approximation of Process Reachability

Initial: < a₀, b₁, c₀, d₀ >

KO (inconclusive)
EGFR/ErbB Signalling Network
(104 components)


Process Hitting
193 sorts,
748 processes,
2356 actions:
$\approx 2 \cdot 10^{96}$ states.
### Temps d’exécution

Pour différentes analyses d’accessibilité:

<table>
<thead>
<tr>
<th>Model</th>
<th>sorts</th>
<th>procs</th>
<th>actions</th>
<th>states</th>
<th>Biocham(^1)</th>
<th>libdd(^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>[3s-KO]</td>
<td>[1s-150s]</td>
<td>0.007s</td>
</tr>
<tr>
<td>tcrsig40</td>
<td>54</td>
<td>156</td>
<td>301</td>
<td>2(^{73})</td>
<td>[1s-KO]</td>
<td>[0.6s-KO]</td>
<td>0.004s</td>
</tr>
<tr>
<td>tcrsig94</td>
<td>133</td>
<td>448</td>
<td>1124</td>
<td>2(^{194})</td>
<td>KO</td>
<td>KO</td>
<td>0.030s</td>
</tr>
<tr>
<td>egfr104</td>
<td>193</td>
<td>748</td>
<td>2356</td>
<td>2(^{320})</td>
<td>KO</td>
<td>KO</td>
<td>0.050s</td>
</tr>
</tbody>
</table>

[Inria Paris-Rocquencourt/Contraintes]
[LIP6/Move]
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the $\pi$-calculus):

5 Basics
6 Presentation of the “Process Hitting”
7 Verification (and control)
8 Simulation and temporal properties
9 Static Analysis by abstraction
10 Conclusion
Summary

- Take time into account through hybrid and algebraic approaches
- Verification by parametrized hybrid model-checking
- Verification by probabilistic model-checking
- Verification by tuned simulations
- Verification by abstraction
- Analyses oriented towards understanding $\leadsto$ detection $\leadsto$ and control
Summary

- Take time into account through hybrid and algebraic approaches
- Verification by parametrized hybrid model-checking
- Verification by probabilistic model-checking
- Verification by tuned simulations
- Verification by abstraction

Analyses oriented towards understanding $\leadsto$ detection $\leadsto$ and control
Summary

- Take time into account through hybrid and algebraic approaches
- Verification by parametrized hybrid model-checking
- Verification by probabilistic model-checking
- Verification by tuned simulations
- Verification by abstraction

- Analyses oriented towards understanding ⇔ detection ⇔ and control
Hybrid and Algebraic Modeling of Biological Systems for the Analysis of their Timing Features

Olivier Roux

IRCCyN (UMR CNRS 6597) - Nantes (FRANCE)
olivier.roux@irccyn.ec-nantes.fr

joint work with Jamil Ahmad, Morgan Magnin, Loïc Paulevé

— Luxembourg, april 25, 2012 —
Mark Alber, Maria Kiskowski, James Glazier, and Yi Jiang.
On cellular automaton approaches to modeling biological cells.

Delays in biological regulatory networks (BRN).

Anil Aswani and Claire Tomlin.
Reachability algorithm for biological piecewise-affine hybrid systems.

Grégoire Batt, Calin Belta, and Ron Weiss.
Model checking genetic regulatory networks with parameter uncertainty.

Grégoire Batt, Calin Belta, and Ron Weiss.
Model checking liveness properties of genetic regulatory networks.

Semantics of biological regulatory networks.

Ralf Blossey, Luca Cardelli, and Andrew Phillips.
Compositionality, stochasticity and cooperativity in dynamic models of gene regulation.

L. Bortolussi and A. Policriti.
Hybrid systems and biology: continuous and discrete modeling for systems biology.

Federica Ciocchetta and Jane Hillston.
Bio-pepa: A framework for the modelling and analysis of biological systems.

Claudine Chaouiya, Elisabeth Remy, and Denis Thieffry.
Petri net modelling of biological regulatory networks.

Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine.
Rule-based modelling of cellular signalling.
In CONCUR pages 17–81, 2007.

Lorenzo Dematté, Corrado Priami, and Alessandro Romanel.

Modelling and simulation of biological processes in blenx.

François Fages and Sylvain Soliman.

Abstract interpretation and types for systems biology.

T-A. Henzinger, P-H. Ho, and H. Wong-Toi.

HYTECH: A model checker for hybrid systems.

John Heath, Marta Kwiatkowska, Gethin Norman, David Parker, and Oksana Tymchyshyn.

Probabilistic model checking of complex biological pathways.

Paola Lecca and Corrado Priami.

Cell cycle control in eukaryotes: A biospi model.

P. Lincoln and A. Tiwari.

Symbolic systems biology: Hybrid modeling and analysis of biological networks.

Andrew Phillips and Luca Cardelli.

A programming language for composable dna circuits.
*Journal of the Royal Society Interface*, 6(S4), 2009.

Michael Pedersen and Gordon Plotkin.

A language for biochemical systems.

Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman.

Application of a stochastic name-passing calculus to representation and simulation of molecular processes.

Aurélien Rizk, Grégoire Batt, François Fages, and Sylvain Soliman.

On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology.