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

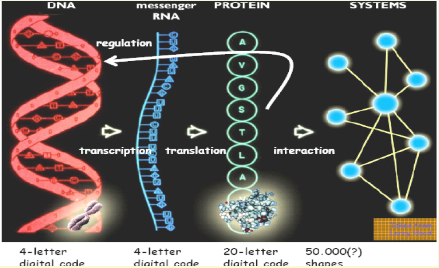
— Luxembourg, april 25, 2012 —

Part:

Introduction - Motivations

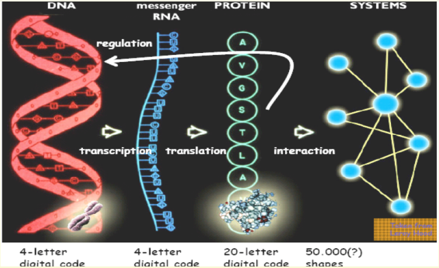
Overview of the part: Introduction - Motivations

Biological Systems Modeling



- ▷ Model ⇝ abstract formal system …
  ⇝ simulations, model checking, …
- ▷ Different abstraction levels...
  - boolean, integer, . . .
  - chronologic, . . .
  - chronometric ⇝ best knowledge of the behaviors

Biological Systems Modeling

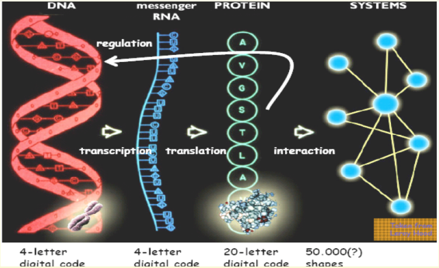


- ▷ Model ⇝ abstract formal system ...

  ⇝ simulations, model checking, ...

- ▷ Different abstraction levels...
  - boolean, integer, . . .
  - chronologic, . . .
  - chronometric ⇝ best knowledge of the behaviors

Biological Systems Modeling



- ▷ Model ⇝ abstract formal system ...

  ⇝ simulations, model checking, ...

- ▷ Different abstraction levels...
  - boolean, integer, . . .
  - chronologic, . . .
  - chronometric ⇝ best knowledge of the behaviors

Biological Systems Modeling

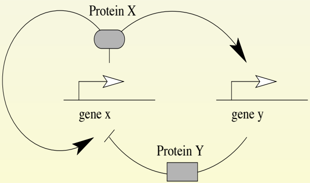
Ultimately, the systems we specify are physical objects, and mathematics cannot prove physical properties. We can prove properties only of a mathematical model of the system; whether or not the system correctly implements the model must remain a question of law and not of mathematics.

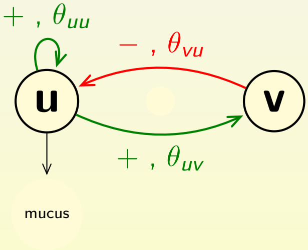
Leslie Lamport
Comm. ACM, 1989

Genetic Regulatory Networks (short introduction)

retour1

activation and inhibition of genes (through the production of proteins)





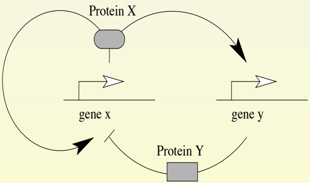
⇝ Model checking biologic Regulatory Networks

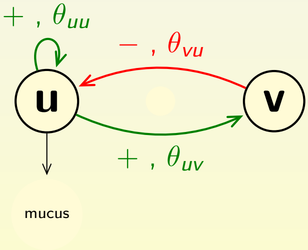
- ▷ understand (modeling)
- ▷ verify (analysis, simulation and tests, model-checking)
- ▷ remedy. . . (control)

Genetic Regulatory Networks(short introduction)

retour1

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)

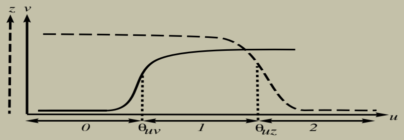
Part:

Hybrid Modeling of Biological Regulatory Networks

Overview of the part: Hybrid Modeling of Biological Regulatory Networks

States, Transitions

Figure 1 : Premiers paramètres discrets



States, Transitions

Figure 3 : Premiers paramètres discrets

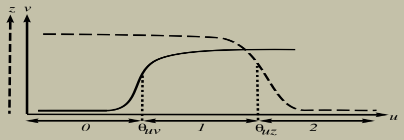
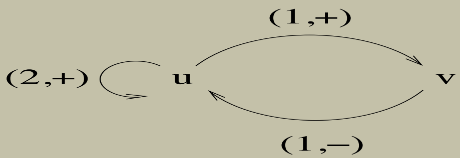
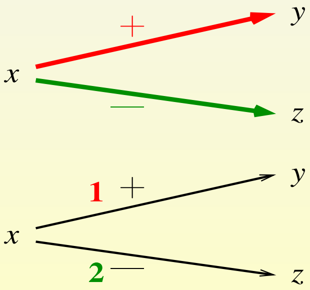
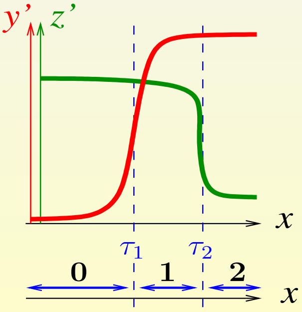


Figure 4 : Exemple: Pseudomonas æruginosa



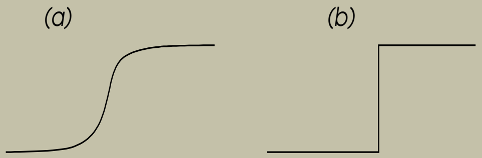
States, Transitions





States, Transitions

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



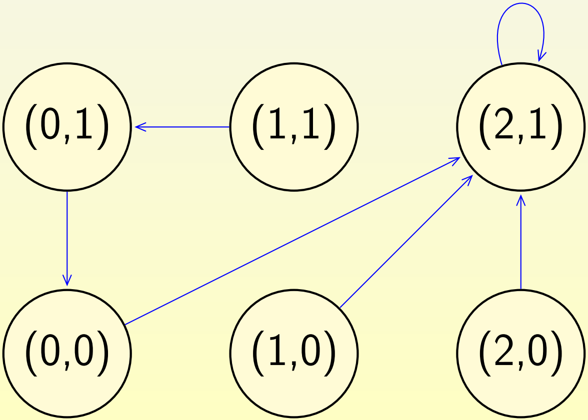
(abstract states) 0, 1, 2 . . . of each component

$\rightsquigarrow$

global state as a tuple.

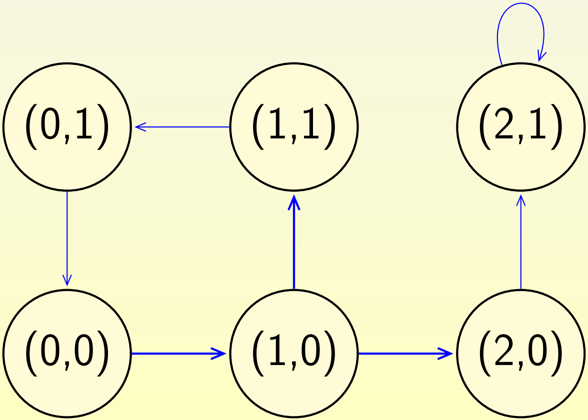
Discrete State Graph

Under the hypothesis of knowing some parameters values, we have:



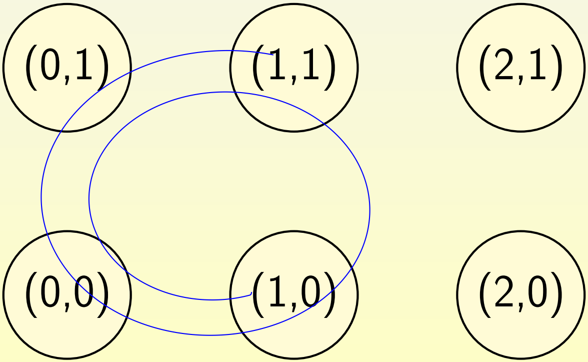
Discrete State Graph

And while de-synchronizing with unit of the “Manhattan distance ”, we get:



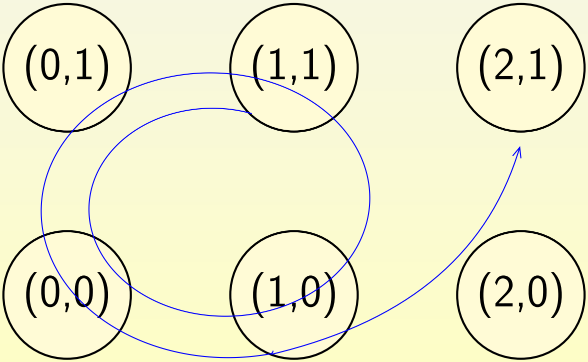
Discrete State Graph

Hence, trajectories: ...



Discrete State Graph

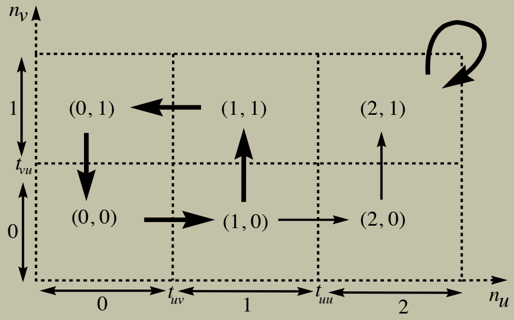
Hence, trajectories: ...



Discrete State Graph and refining

rappel retour2

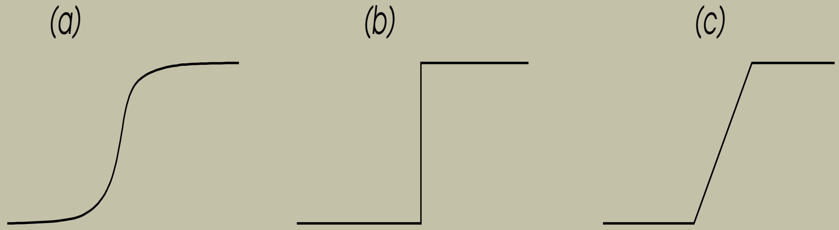
Figure 6 : Pseudomonas æruginosa (cont'): Behaviors and … need for improvement



Overview of the part: Hybrid Modeling of Biological Regulatory Networks

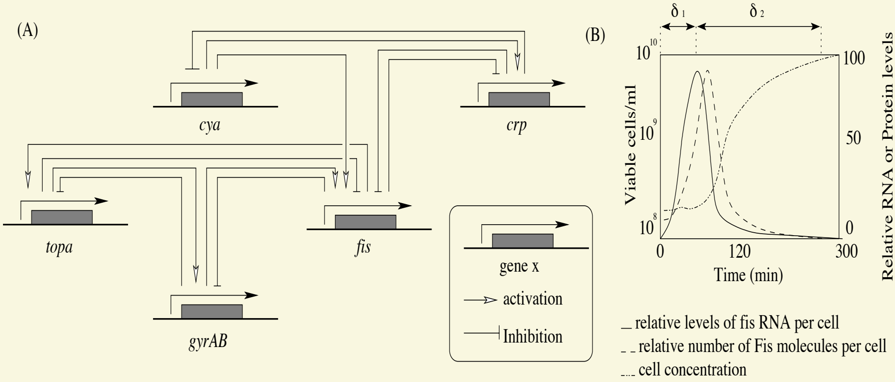
Introduction. Time: duration, delay...

Figure 7 : A sigmoïd (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*

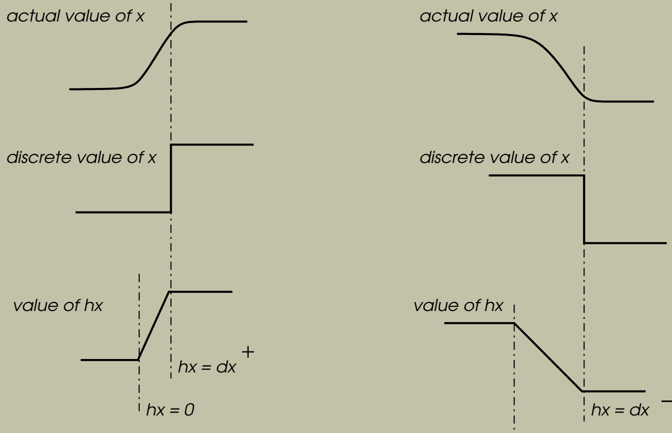


[Ropers et al. 2007]

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
  - [M. Adélaïde and G. Sutre]. "Parametric Analysis and Abstraction of Genetic Regulatory Networks". In *Concurrent Models in Molecular Biology (BioCONCUR'04), London, UK, Aug. 2004, Electronic Notes in Theor. Comp. Sci., Elsevier, 2004.*
  - [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)*
  - [O. Maler, G. Batt]. "Approximating Continuous Systems by Timed Automata", In *Formal Methods in Systems Biology, 2008*

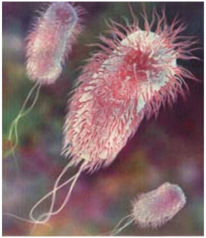
Related works (suite)

- et aussi:
  - [L. Bortolussi and A. Policriti]∗. ”Hybrid Systems and Biology. Continuous and Discrete Modeling for Systems Biology”. In *Formal Methods For Computational System Biology (FMCSB)*, LNCS 5016, 2006. [BP06]
  - [P. Lincoln and A. Tiwari]. ”Symbolic systems biology: Hybrid modeling and analysis of biological networks”. In *Hybrid Systems: Computation and Control (HSCC’04)*, LNCS 2993, 2004. [LT04]
  - [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]
  - [G. Batt, C. Belta and R. Weiss]. ”Model checking genetic regulatory networks with parameter uncertainty”. In *Hybrid systems (HSCC’07)*, LNCS, 2007. [BBW07a]
  - [A. Aswani and C. Tomlin]. ”Reachability algorithm for biological piecewise-affine hybrid systems”. *In Hybrid systems (HSCC’07)*, LNCS, 2007. [AT07]

Examples of use of HS for Systems Biology

Escherichia coli

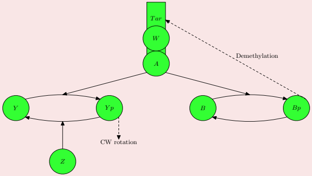
- a bacterium detecting the food concentration through a set of receptors;
- moving by flagellar rotations.



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

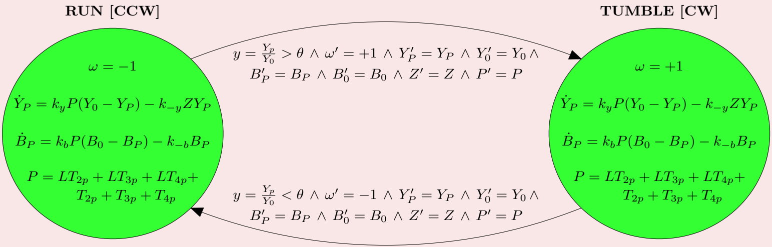
- "RUNS" – it moves in a straight line by moving its flagella counterclockwise (CCW)
- "TUMBLES" – it randomly changes its heading by moving its flagella clockwise (CW)

E. Coli Model



Examples of use of HS for Systems Biology

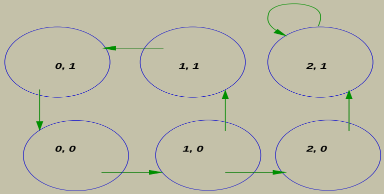
E. coli IDA Model



A. Casagrande et al., *Independent Dynamics Hybrid Automata in Systems Biology*, AB('05) Tokyo, 2005

Temporal Zones *for biological systems modeling*

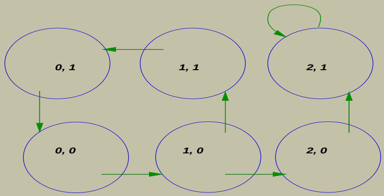
Figure 11 : Example of a discrete model of a GRN (again)



rappel retour

Temporal Zones *for biological systems modeling*

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



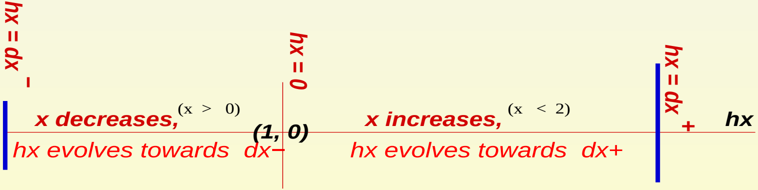
rappel retour

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')

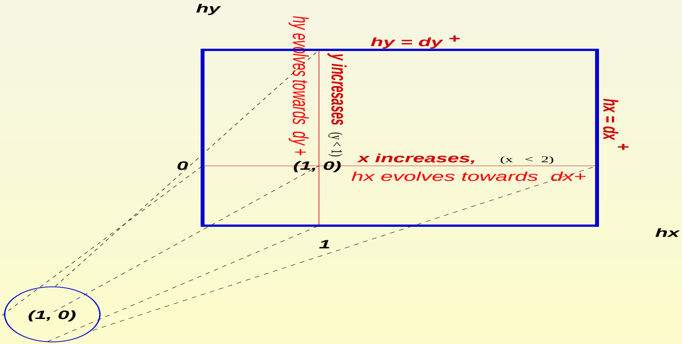


Figure 13 : Expansion de la localité (1, 0) *en une zone temporelle:* (h_x, h_y)

Temporal Zones *for biological systems modeling* cont')

Executions consistent with the dynamics (in green)

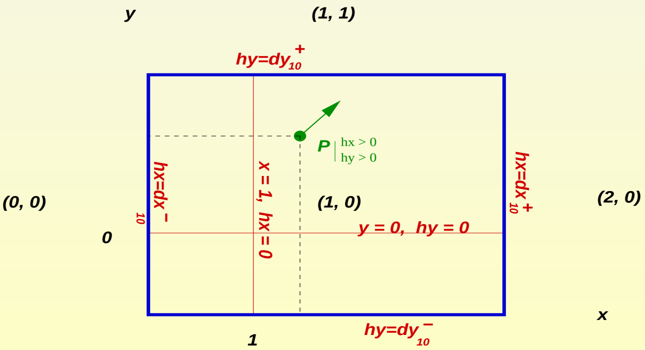


Figure 14 : Example of the "temporal" modeling of a GRN (partial view)

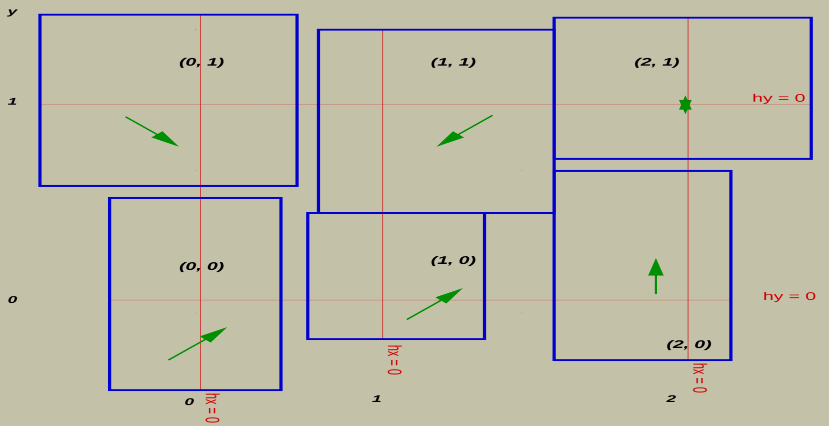
Temporal Zones *for biological systems modeling* (cont')

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



rappel

Hybrid Model of the example

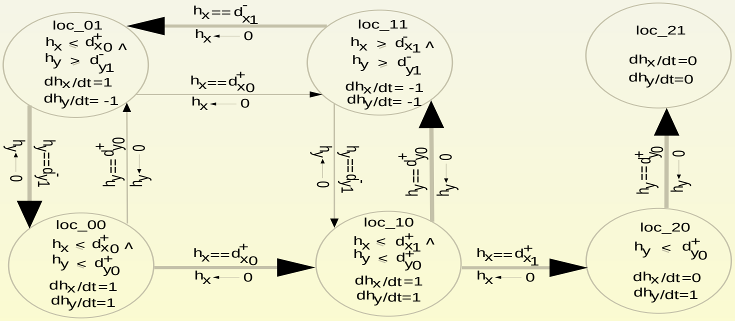
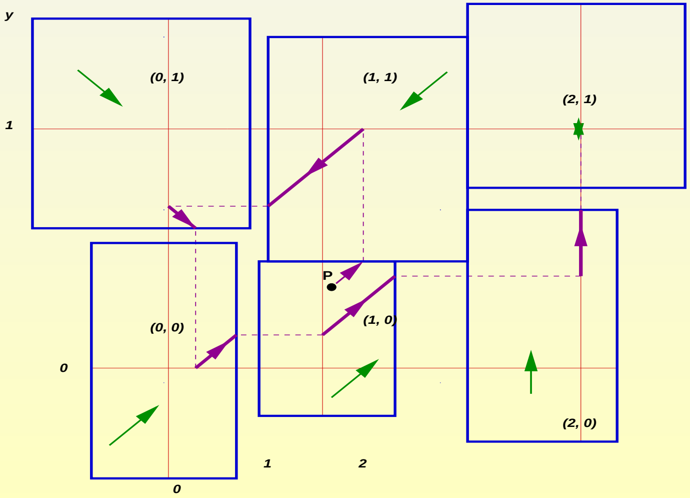


Figure 16 : Pseudomonas æruginosa

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

Trajectories

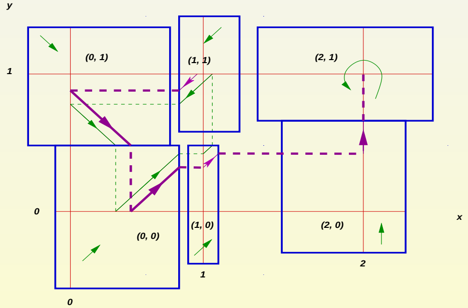
Sequence of alternating continuous and discrete transitions



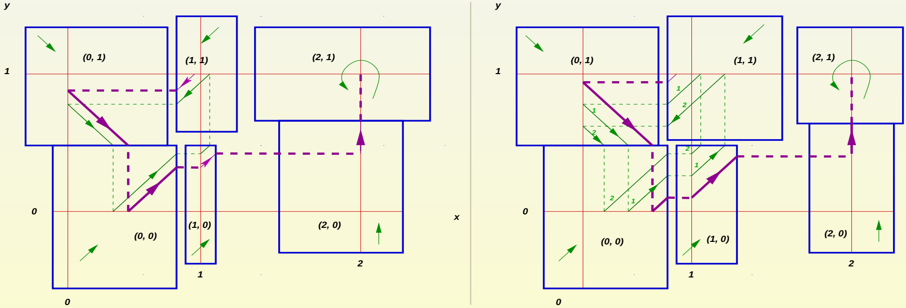
Analysis, first verification results and discussion

- ▷ Evolution (analysis of the trajectories)
- ▷ Results
- ▷ Limitations

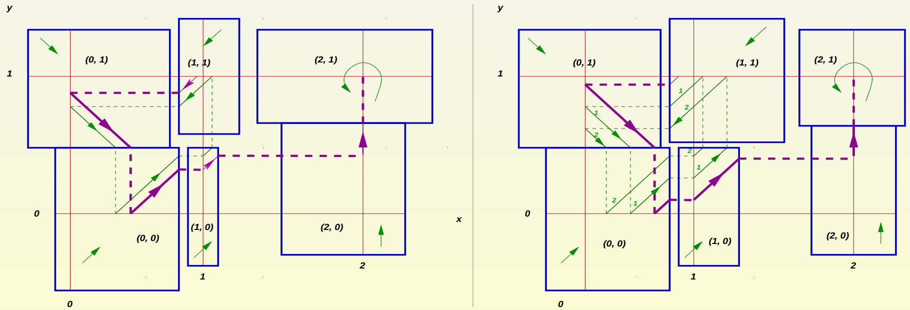
Divergences and Invariance Kernels



Divergences and Invariance Kernels



Divergences and Invariance Kernels



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"*.

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

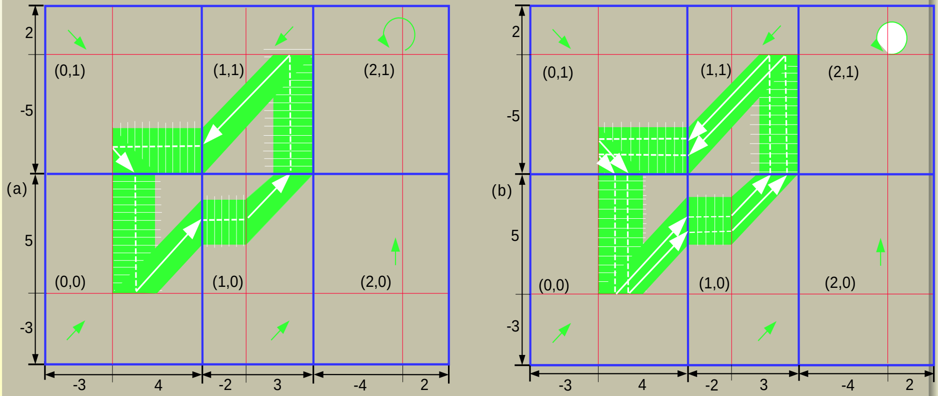
Solution:

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

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.

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

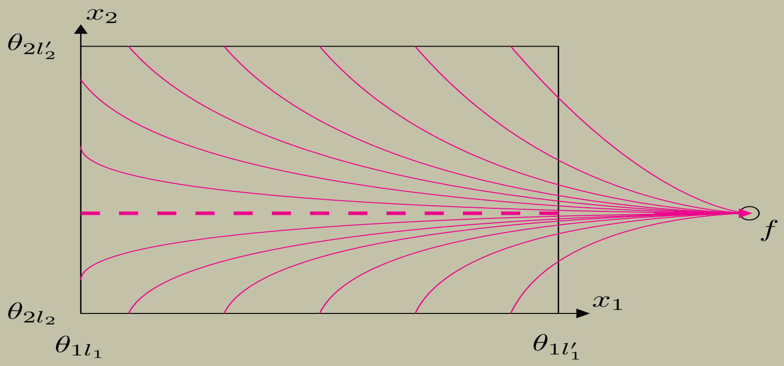
but

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

$\rightsquigarrow$ Modeling Refinement ...

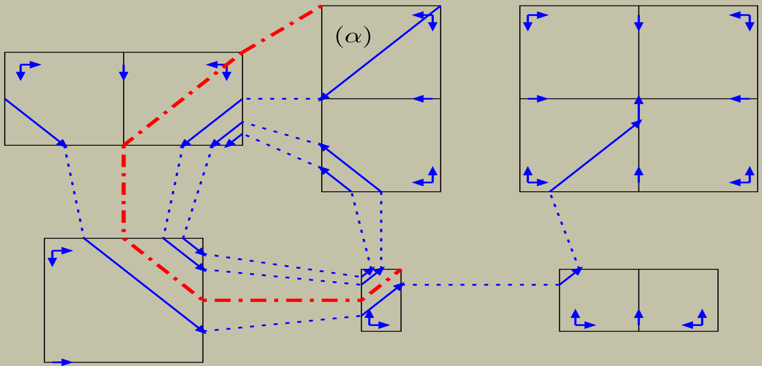
Modeling Refinement: *Asymptotic Evolution towards the focal point*

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



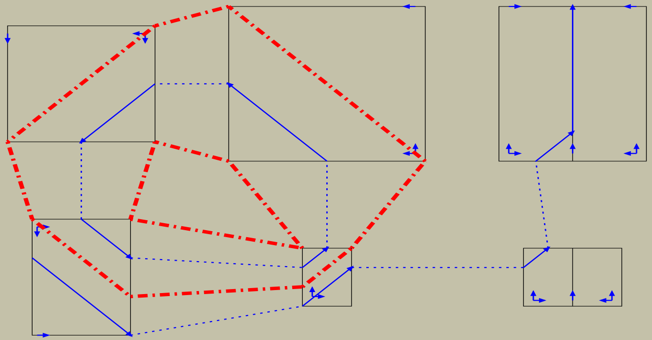
Modeling Refinement (cont'): idea and overview

Figure 19 : Hybrid dynamics with centripetal spirals



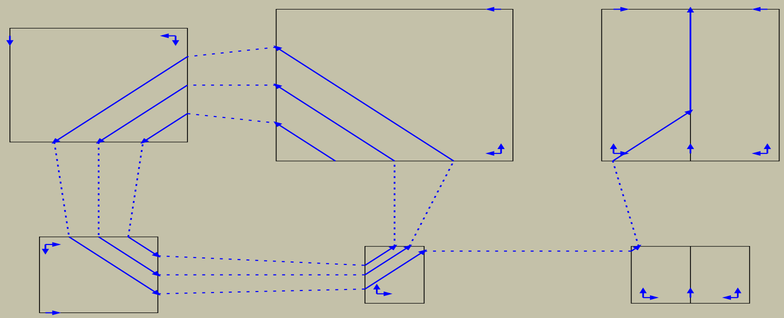
Modeling Refinement (cont'): idea and overview

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



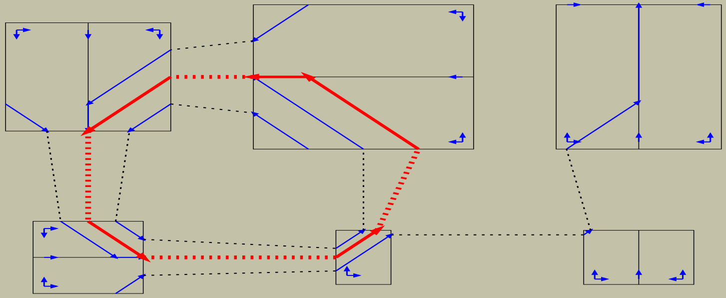
Modeling Refinement (cont'): idea and overview

Figure 21 : Hybrid dynamics with divergent spirals



Modeling Refinement (cont'): idea and overview

Figure 22 : Hybrid dynamics with a limit cycle (in thick red line)



Part:

Algebraic Modeling. The "Process Hitting" (from the π-calculus):

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

$\rightsquigarrow$ Towards an algebraic approach

(yet with timie concerns)

The π-calculus [R. Milner 1992]

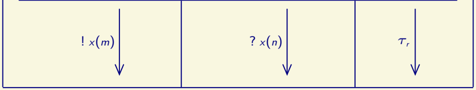
We use only a fragment

```
Dec ::= new x {@r} : t                          Channel Declaration
      | type n = t                              Type Declaration
      | val m = v                               Value Declaration
      | run P                                   Process Declaration
      | let D1 and ... and DN                   Definitions, N >= 1
 D   ::= X (m1, ... , mN ) = P                  Definition, N >= 0
 P   ::= ()                                     Null Process
      | (P1 | ... | PM)                         Parallel, M >= 2
      | X (v1, ... , vN)                        Instantiation, N >= 0
      | a{; P }                                 Action
      | do a1{; P1} or ... or aM{; PM}          Choice, M >= 2
      | (Dec1 . . . DecN P)                     Declarations, N >= 0
 aI ::= !x {(v1, ... , vN)}                     Output, N >= 0
      | ?x {(m1, ... , mN)}                     Input, N >= 0
      | delay @ r                               Delay
```

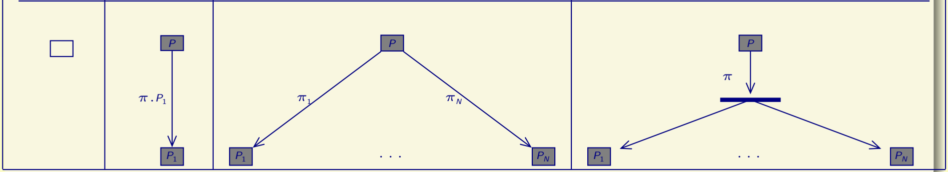
rate: $\mathtt{r} \rightsquigarrow$ mean duration: $1/\mathtt{r}$, variance: $1/\mathtt{r}^2$

The π-calculus: *Fragment*

Action π =::		
$!x(m)$	$?x(n)$	τ_r
sends value m on channel x	Receives value n on channel x	Delay at rate r



Process P=::			
0	$\pi . P_1$	$\pi_1 . P_1 + \ldots + \pi_N . P_N$	$\pi . (P_1 \mid \ldots \mid P_N)$
Nul process	makes an action	Chooses between actions	Parallel composition of processes



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].

Related works (cont')

... 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 π-calculus):

A new formalism for dynamic systems

Idea: split structure and dynamics ...

...avoiding the whole transitions relation:

$$\begin{array}{ccc} f_0 c_0 a_0 & \longrightarrow & f_0 c_0 a_1 \\ & & \\ & \dots & \\ f_0 c_1 a_1 & \longrightarrow & f_0 c_1 a_0 \\ f_1 c_1 a_1 & \longrightarrow & f_1 c_1 a_0 \\ & \dots & \end{array}$$

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 ...

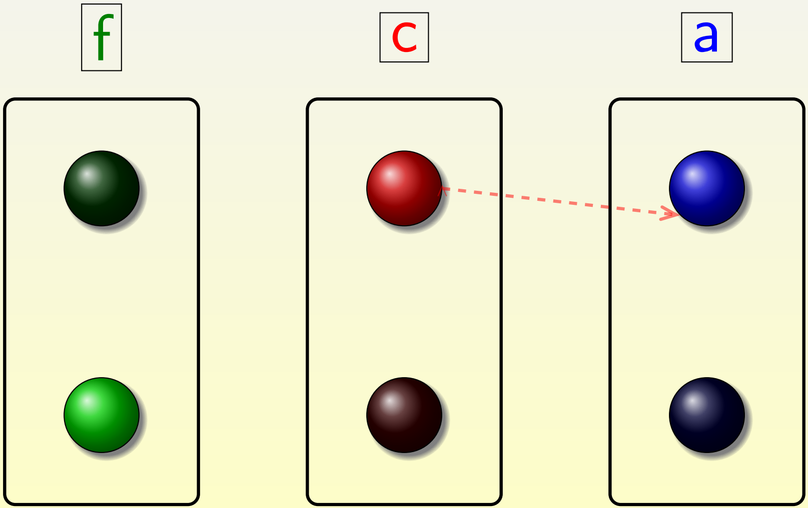
...avoiding the whole transitions relation:

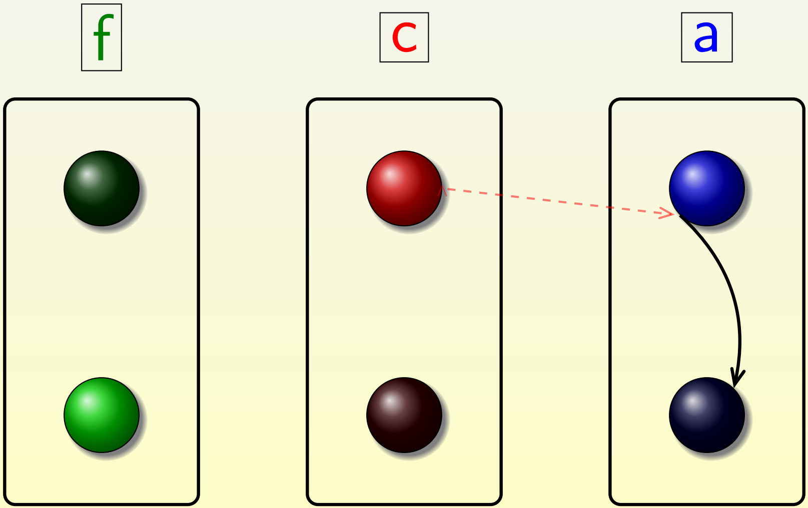
$$
\begin{array}{ccc}
f_0 c_0 a_0 & \longrightarrow & f_0 c_0 a_1 \\
 & \ldots & \\
f_0 c_1 a_1 & \longrightarrow & f_0 c_1 a_0 \\
f_1 c_1 a_1 & \longrightarrow & f_1 c_1 a_0 \\
 & \ldots &
\end{array}
$$

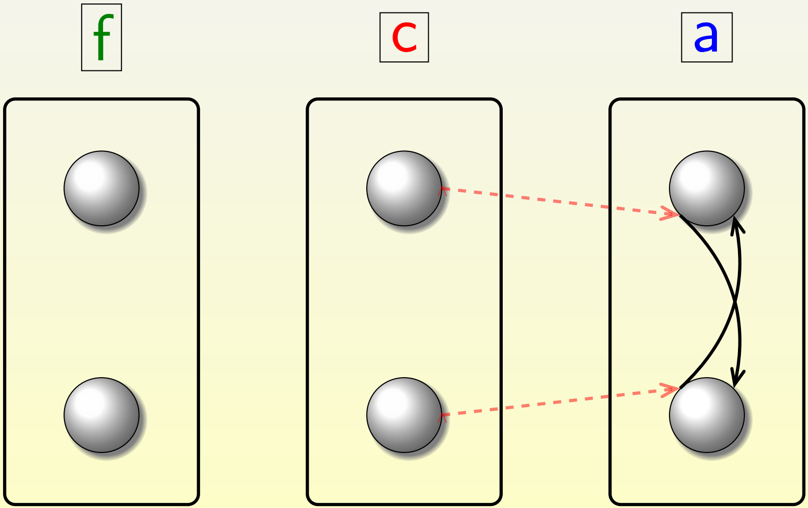
in a set of 2^n states...

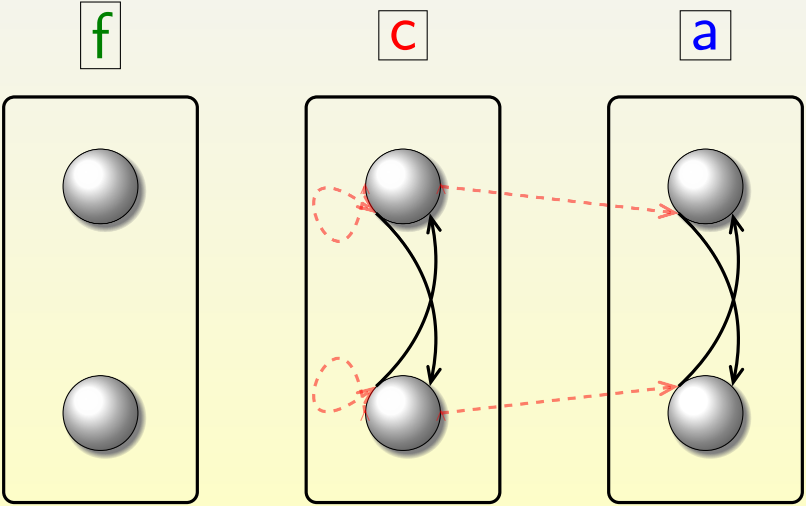
n *"sorts"* 1 and only 1 *"processus"* is alive at each time

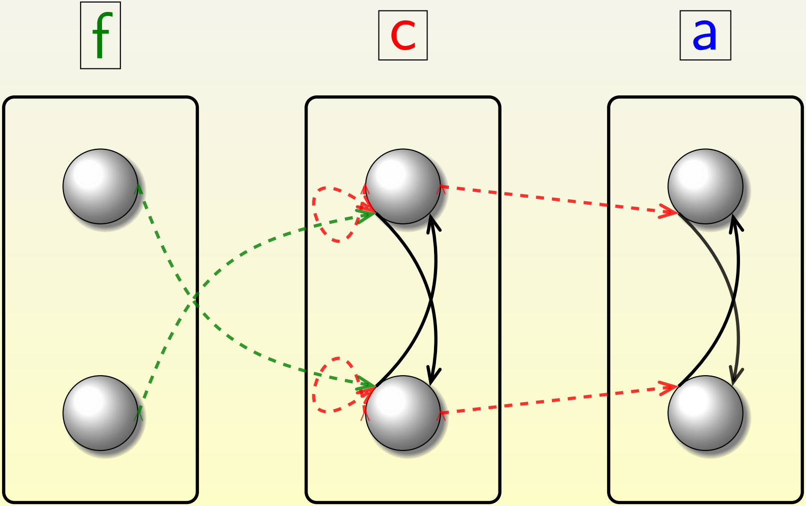
n *"sorts"* 1 and only 1 *"processus"* is alive at each time

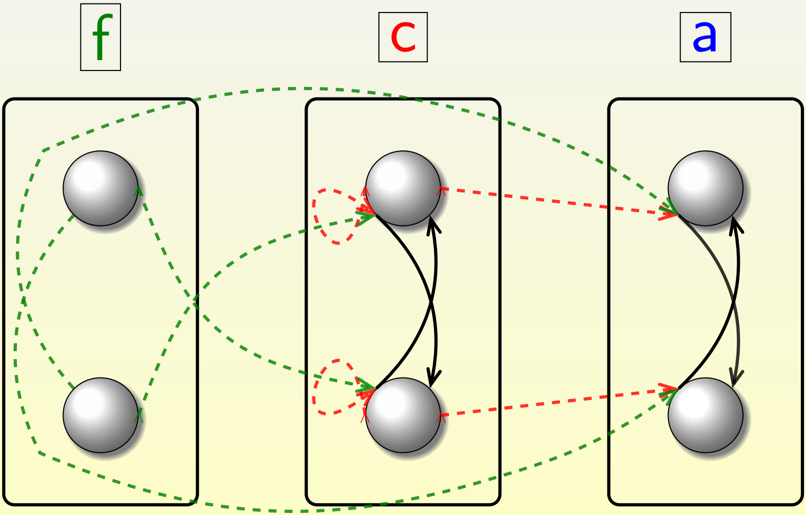

f
c
a


f
c
a


f
c
a

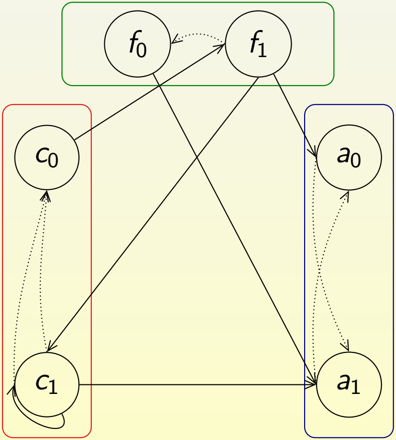

f
c
a


f
c
a


f
c
a

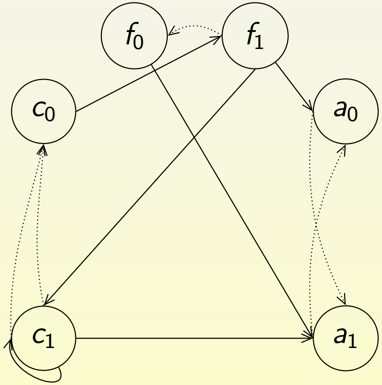
Overview of the part: Algebraic Modeling. The "Process Hitting" (from the π-calculus):

Verification of structural properties: stable states

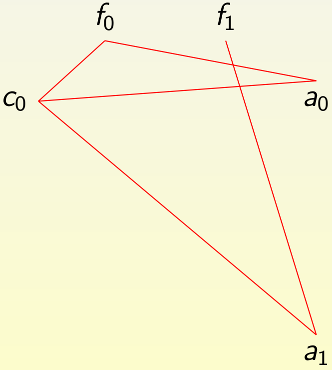


Hypergraph of the *"Process Hitting"*

Verification of structural properties: stable states

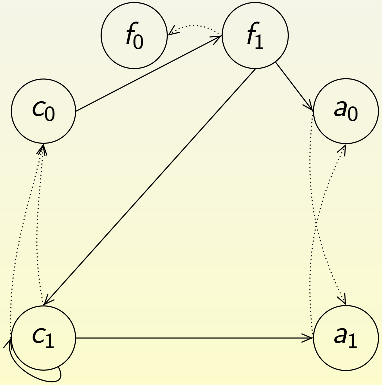


Hypergraph of the "Process Hitting"

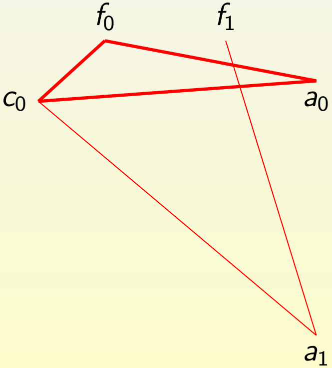


"Hitless" Graph

Verification of structural properties: stable states



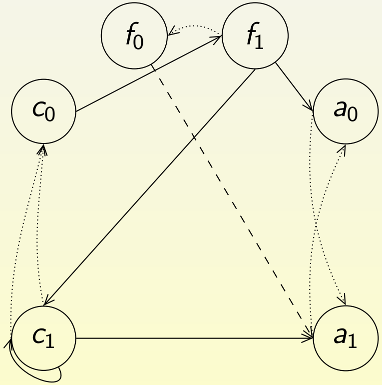
Hypergraph of the "Process Hitting"

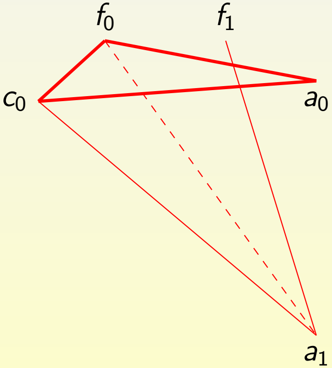


"Hitless" Graph

The *n*-cliques <u>are</u> the stable states: $f_0 c_0 a_0$

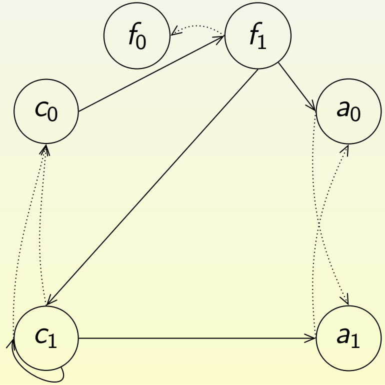
Verification of structural properties: stable states

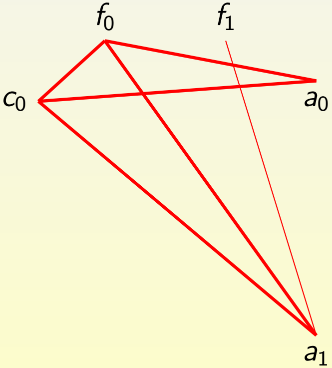




The *n*-cliques <u>are</u> the stable states: $f_0 c_0 a_0$

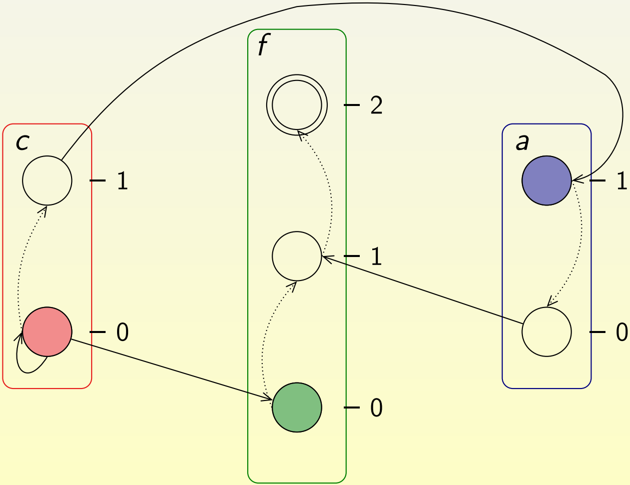
Verification of structural properties: stable states





The *n*-cliques are the stable states: $f_0 c_0 a_0$ and $f_0 c_0 a_1$

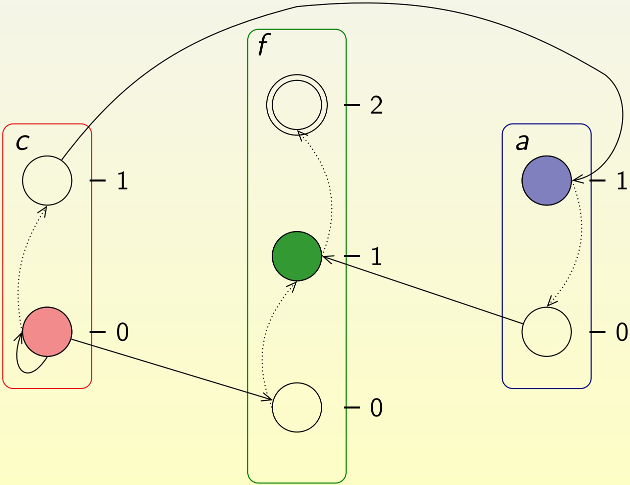
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

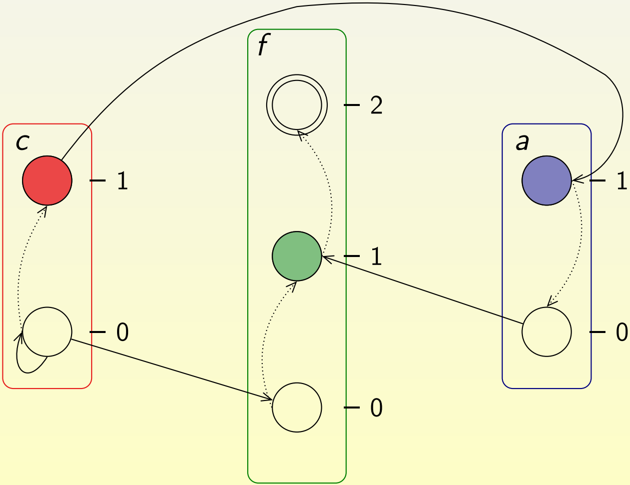
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

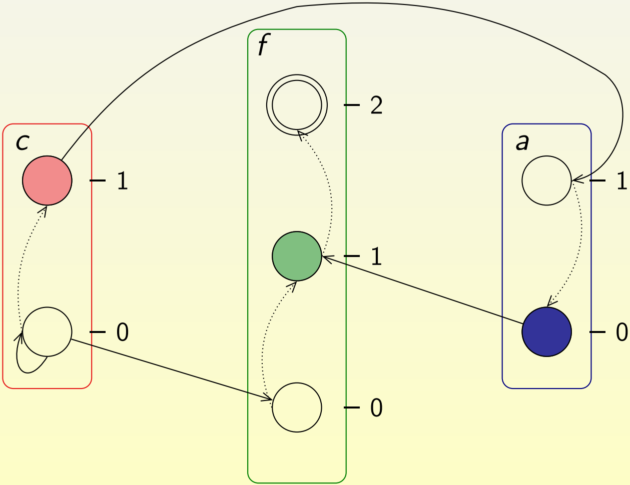
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

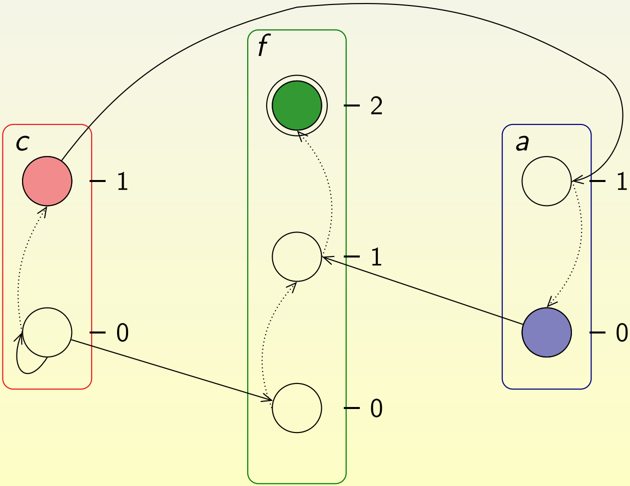
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

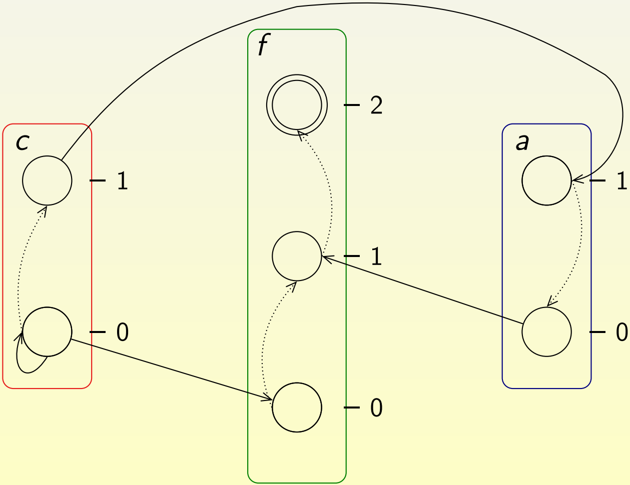
Verification of dynamic properties: reachability



*f*_2 accessible!

retour

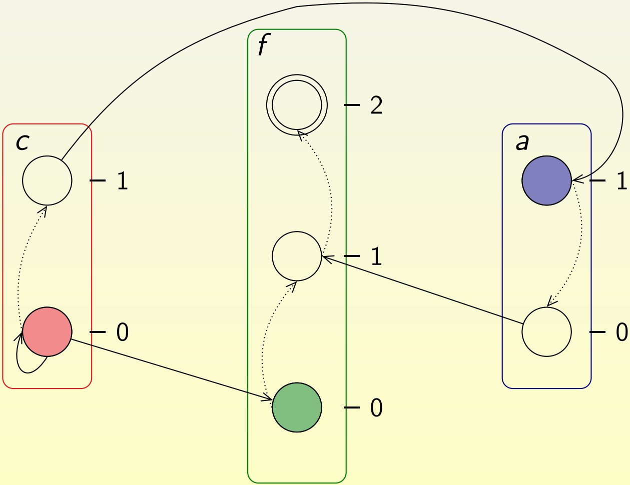
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

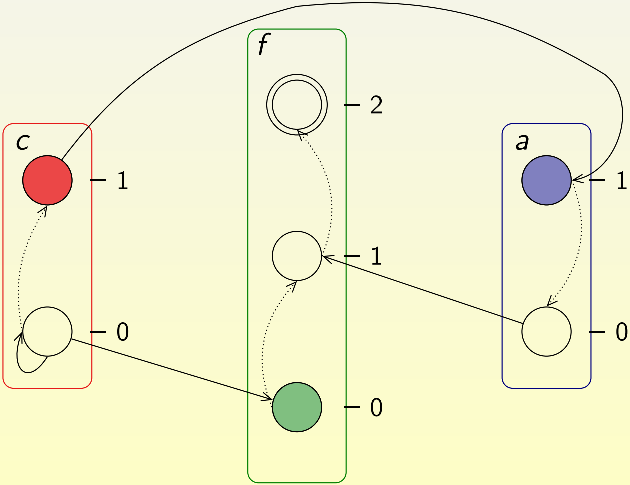
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

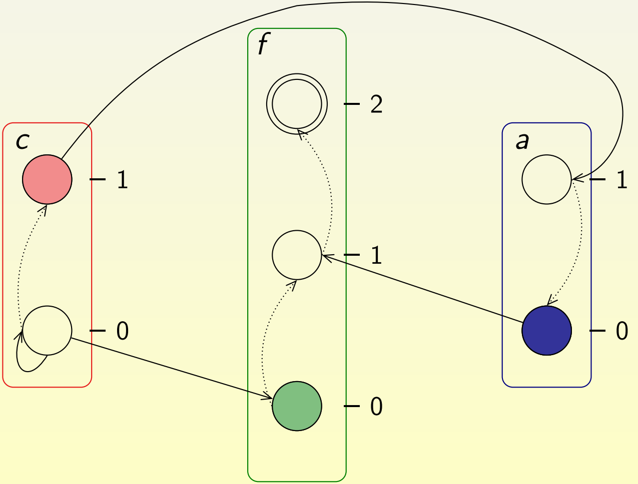
Verification of dynamic properties: reachability



Accessibility of f_2?

retour

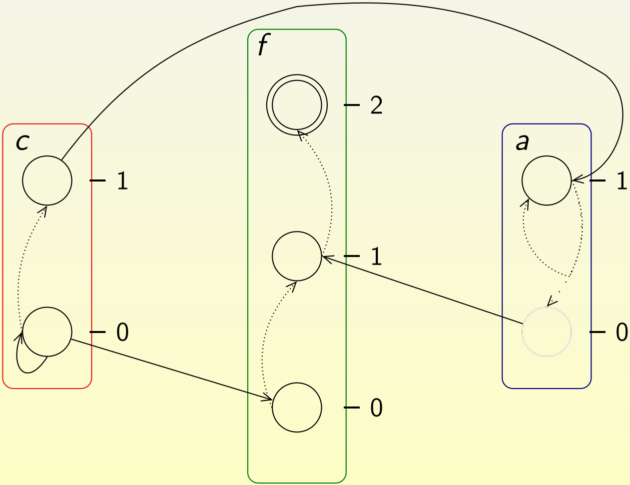
Verification of dynamic properties: reachability



Lock: f_2 non accessible!

retour

Verification of dynamic properties: control



Control of f_2-unreachability: prevent from a_0

retour

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

Translations

- in the π-calcul

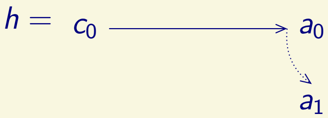
d then

- in SPIM $\rightsquigarrow$ simulations
- in PRISM $\rightsquigarrow$ probabilistic model-checking

and

- the inference of temporal parameters

Stochastic Framework



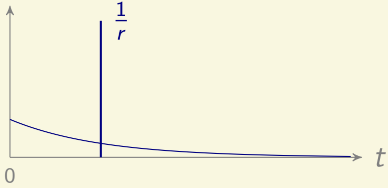
$$\equiv \begin{cases} C_0 ::= \cdots + !\gamma_h . C_0 \\ A_0 ::= \cdots + ?\gamma_h . A_1 \end{cases}$$

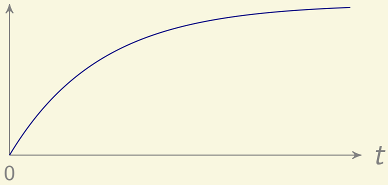
- straightforward translation in the stochastic π-calculus
- to each channel γ_h: use rate r_h
- natural introduction of stochastic parameters into the Process Hitting framework
- Gillespie: reaction duration follows an exponential law
- average duration of an action with use rate r: $\frac{1}{r}$

Tuning stochasticity: temporal parameter synthesis

Example: *self-hitting* process:

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

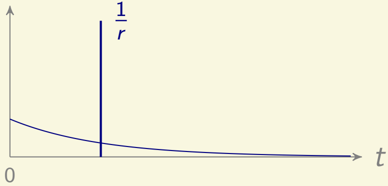


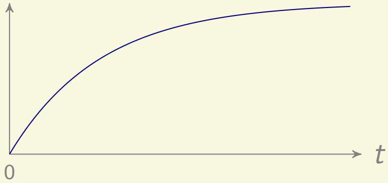


Tuning stochasticity: temporal parameter synthesis

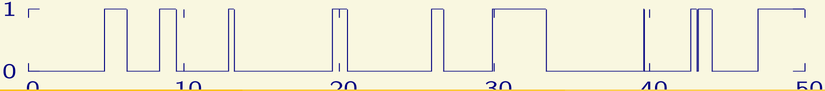
Example: *self-hitting* process:

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





Simulation through SPIM [A. Phillips]:



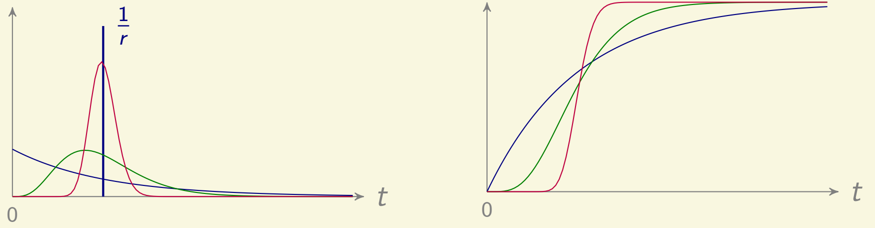
Stochasticity absorption factor

"Duration follows one exponential random variable of rate r"

becomes

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

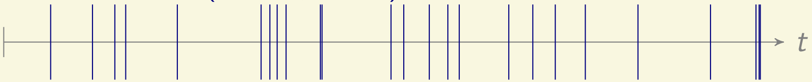
Erlang distribution (particular Gamma) of shape sa and rate $r.sa$:



$sa = 1, \quad sa = 5, \quad sa = 50$

Stochasticity Absorption

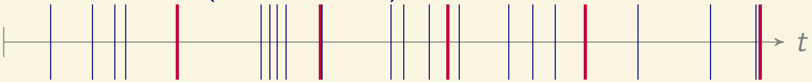
Use rate $r = 1$ ($m = 1, v = 1$)



- $\hat{m} = 1.144, \hat{v} = 0.764$
- $\hat{m} = 5.72, \hat{v} = 0.539$
- Stochasticity absorption factor $sa = 5$ ($r = 1 * 5$):
- $\hat{m} = 1.004, \hat{v} = 0.344$

Stochasticity Absorption

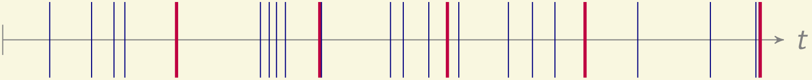
Use rate $r = 1$ ($m = 1, v = 1$)



- $\hat{m} = 1.144, \hat{v} = 0.764$
- $\hat{m} = 5.72, \hat{v} = 0.539$
- Stochasticity absorption factor $sa = 5$ ($r = 1 * 5$):
- $\hat{m} = 1.004, \hat{v} = 0.344$

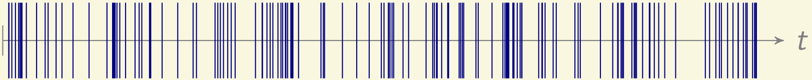
Stochasticity Absorption

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



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

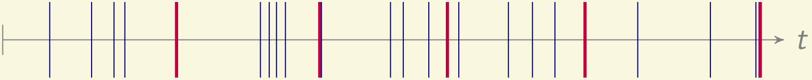
- Stochasticity absorption factor $sa = 5$ ($r = 1 * 5$):



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

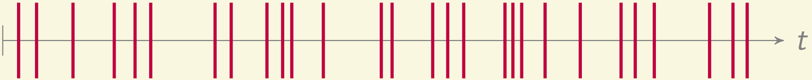
Stochasticity Absorption

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 $sa = 5$ $(r = 1 * 5)$:

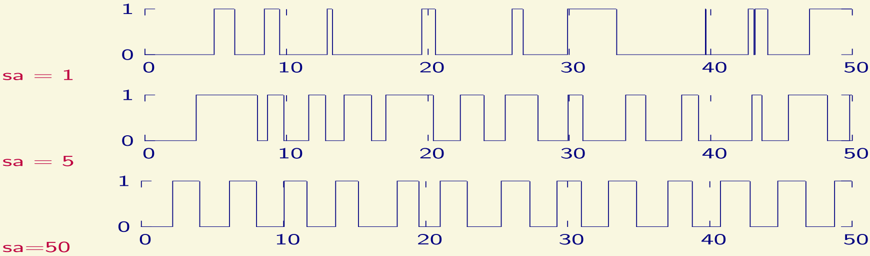


- $\hat{m} = 1.004, \hat{v} = 0.344$

Temporal and stochastic parameters

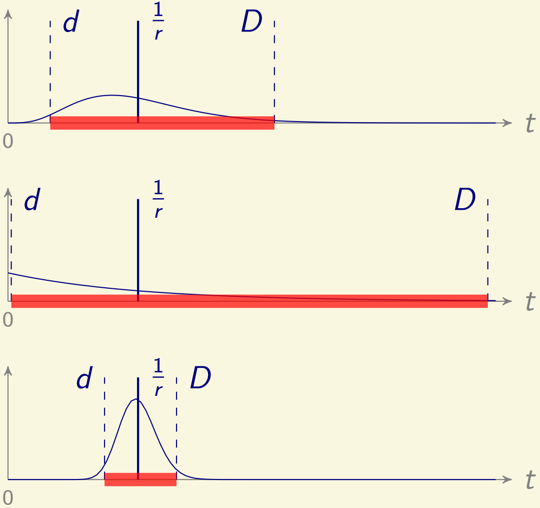
Example: *self-hitting* process *(again)*:

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



Firing intervals

Given a confidence $1 - \alpha$:



$(r, sa) \rightleftarrows$

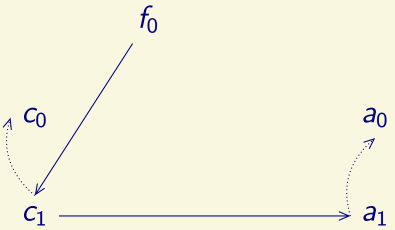
lower sa

higher sa

Further results: behaviour avoidance

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

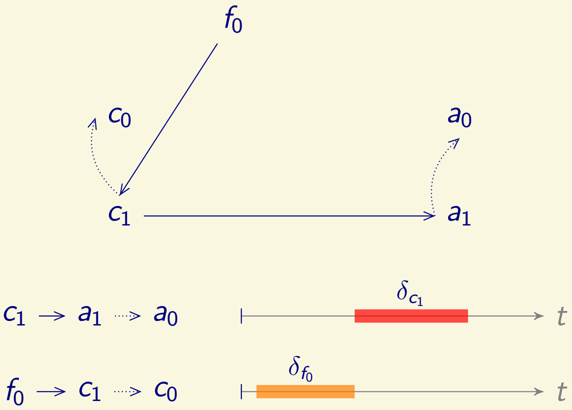
Prevent a from being bounced from 1 to 0



Further results: behaviour avoidance

Example: $f_0 \longrightarrow c_1 \upharpoonright c_0$ and $c_1 \longrightarrow a_1 \upharpoonright a_0$

Prevent a from being bounced from 1 to 0



Translation in PRISM (example)

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

```
ctmc                                          // continuous-time markov chains
module proc_c
c:  [0..1] init 1;                            // states c0, c1
[h_0] c=1 -> 0.25:  (true);                   // delay δc1 = 4
[h_1] c=1 ->         (c'=0);
endmodule
module proc_f
f:  [0..1] init 0;                            // states f0, f1
     [h_1] f=0 − > 2.0:  (true);        // delay δf0 = 1/2  endmodule
module proc_a
a:  [0..1] init 1;                            // states a0, a1
     [h_0] a=1 − >         (a'=0);
endmodule
```

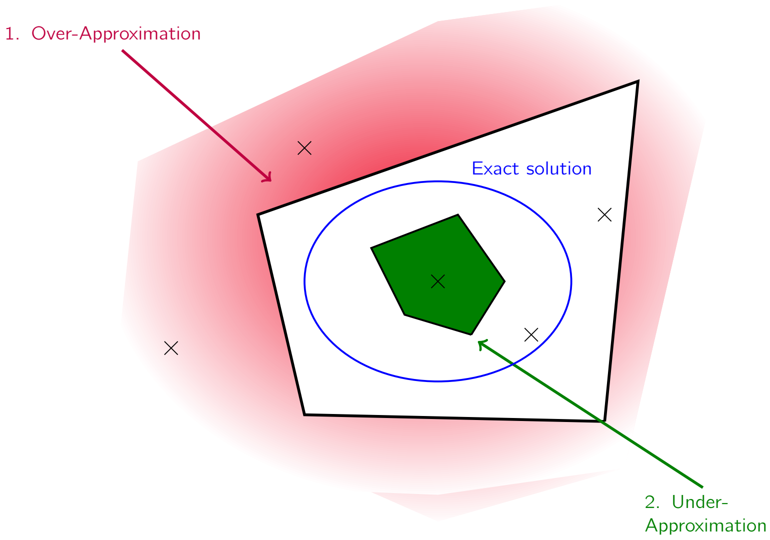
PRISM model-checking and simulation

without and with stochasticity absorption factor:

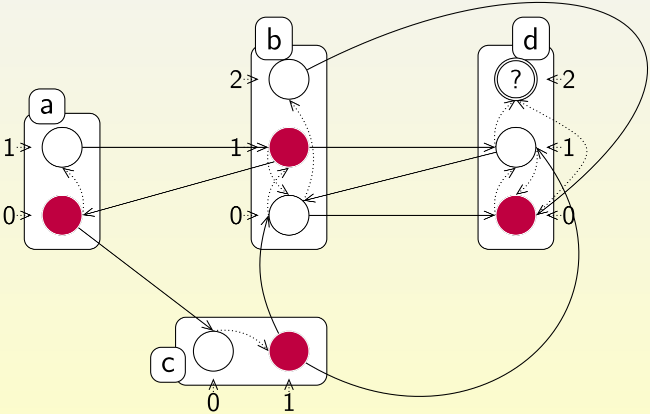
```
Model checking: P = ?  [ F (a=0) ]
Time for model checking:  0.0010 seconds.
Result (probability):  0.1111111111111111

-----------------------------
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 π-calculus):

Over- and Under-Approximation



The Process Reachability Problem: Running Example

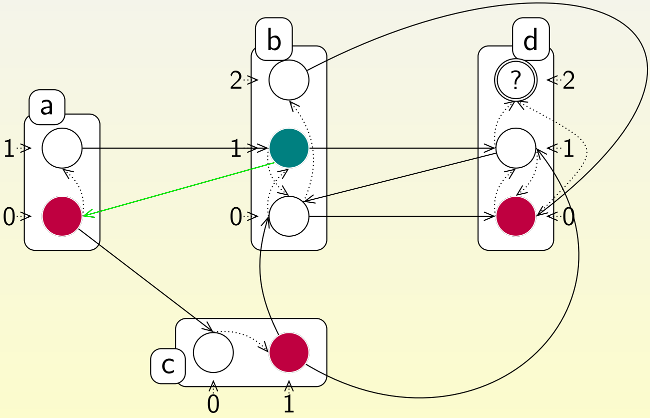


Is d_2 reachable from d_0 in $< a_0, b_1, c_1, d_0 >$?

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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

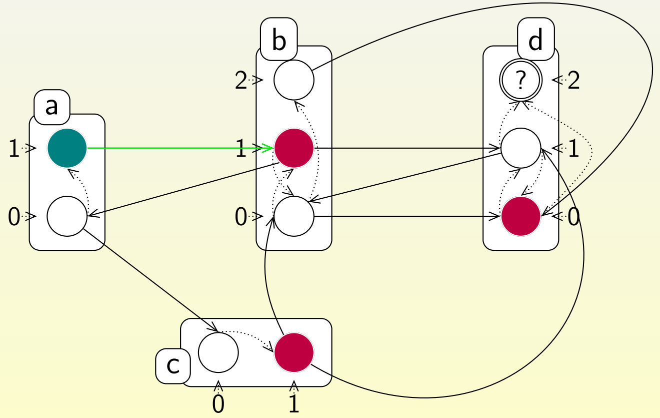
The Process Reachability Problem: Running Example



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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

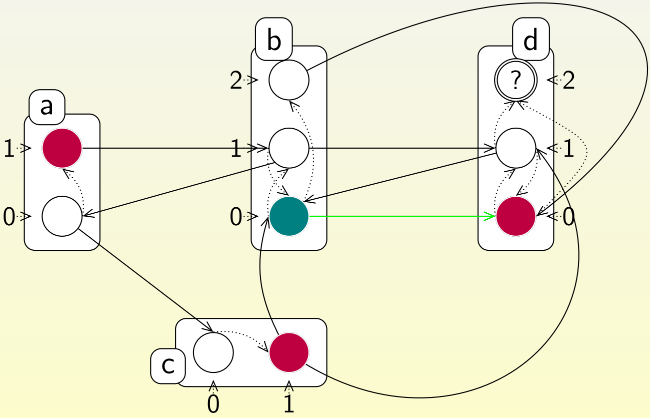
The Process Reachability Problem: Running Example



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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

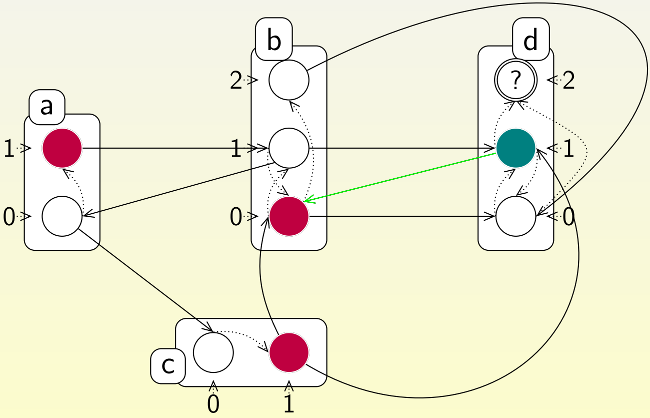
The Process Reachability Problem: Running Example



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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

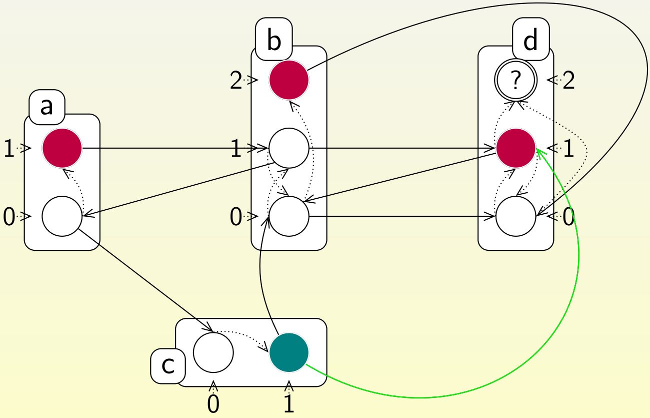
The Process Reachability Problem: Running Example



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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

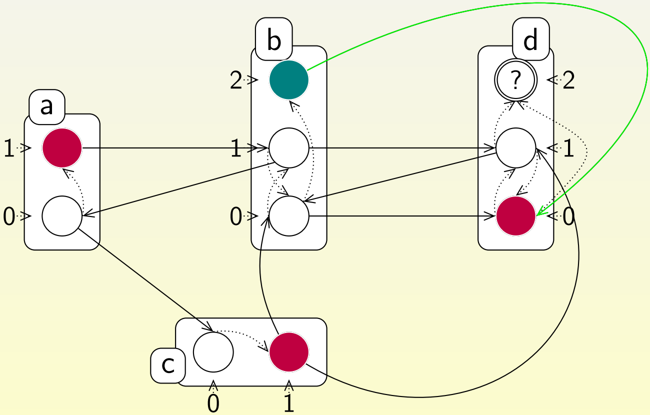
The Process Reachability Problem: Running Example



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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

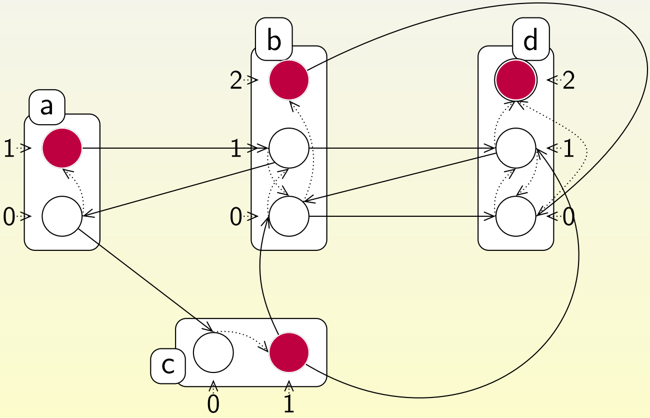
The Process Reachability Problem: Running Example



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

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

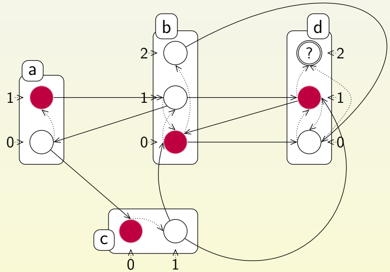
The Process Reachability Problem: Running Example



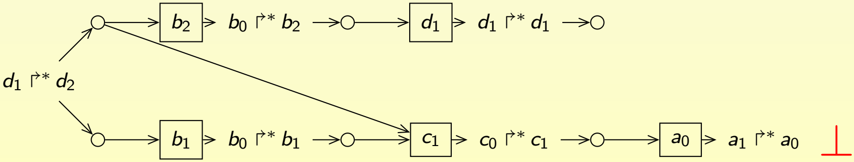
Is d_2 reachable from d_0 in $< a_0, b_1, c_1, d_0 >$? e.g. initial state $< a_0, b_1, c_1, d_0 >$

$b_1 \rightarrow a_0 \upharpoonright a_1, a_1 \rightarrow b_1 \upharpoonright b_0, b_0 \rightarrow d_0 \upharpoonright d_1, d_1 \rightarrow b_0 \upharpoonright b_2, c_1 \rightarrow d_1 \upharpoonright d_0, b_2 \rightarrow d_0 \upharpoonright d_2$

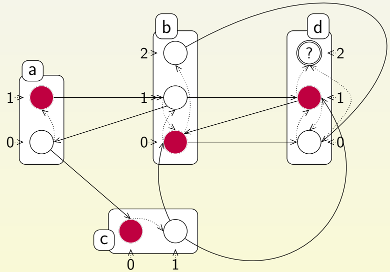
Over-approximation of Process Reachability



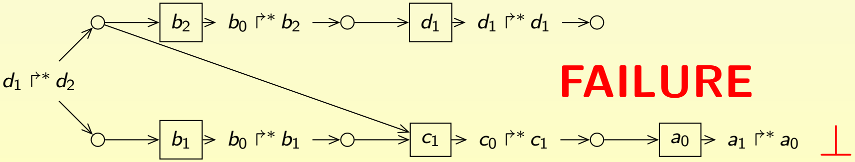
- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailled);
- Necessary condition: there always exists a solution ending with a trivial objective.



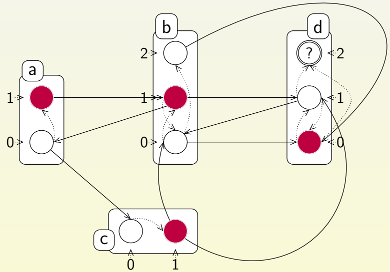
Over-approximation of Process Reachability



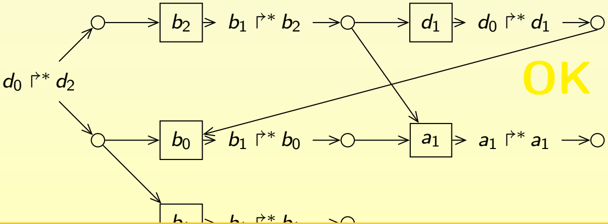
- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailled);
- Necessary condition: there always exists a solution ending with a trivial objective.



Over-approximation of Process Reachability

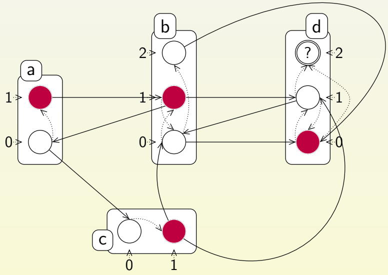


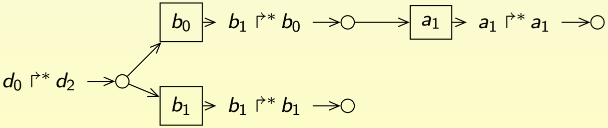
- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailled);
- Necessary condition: there always exists a solution ending with a trivial objective.



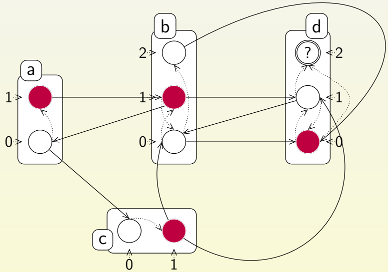
OK (inconclusive)

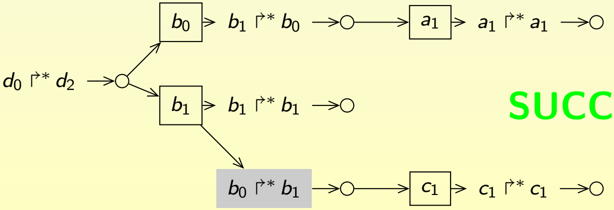
Under-approximation of Process Reachability





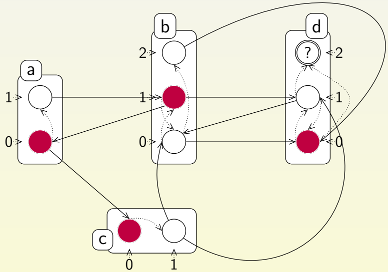
Under-approximation of Process Reachability

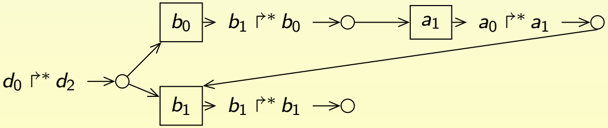




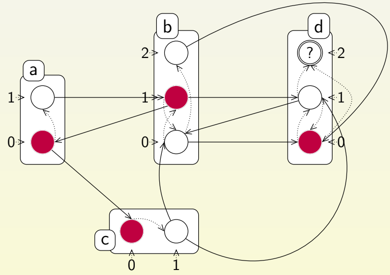
SUCCESS

Under-approximation of Process Reachability

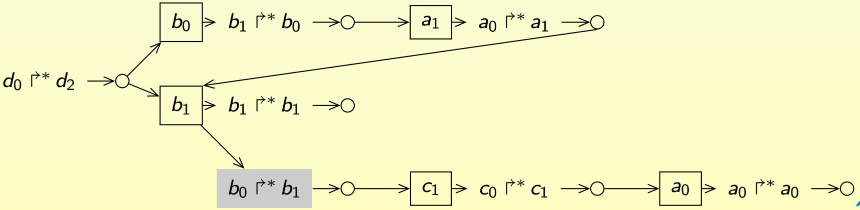




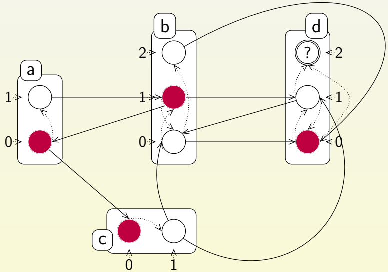
Under-approximation of Process Reachability

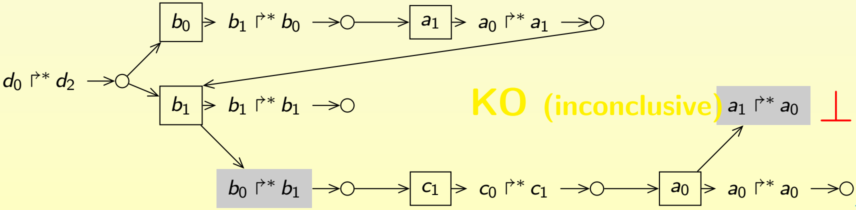


Initial: $\langle a_0, b_1, c_0, d_0 \rangle$

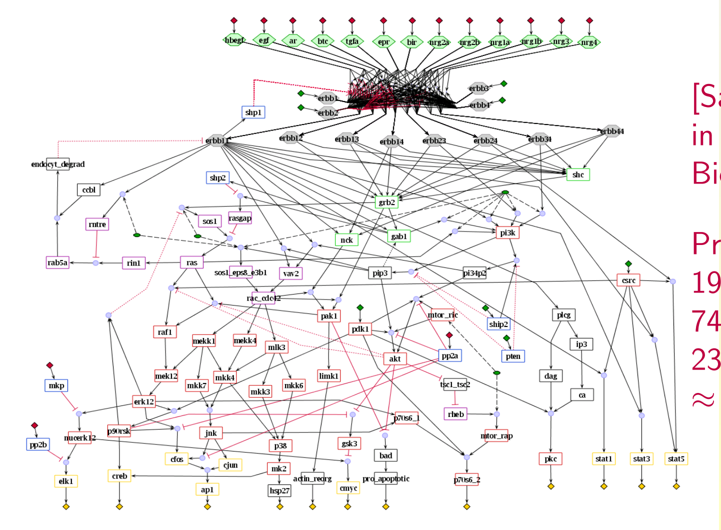


Under-approximation of Process Reachability





EGFR/ErbB Signalling Network
(104 composants)



[Samaga, *et al.* in PLoS Comput Biol, 2009]

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

Temps d'exécution

Pour différentess analyses d'accessibilité:

Model	sorts	procs	actions	states	Biocham[1]	libddd[2]	PINT
egfr20	35	196	670	2^{64}	[3s-KO]	[1s-150s]	0.007s
tcrsig40	54	156	301	2^{73}	[1s-KO]	[0.6s-KO]	0.004s
tcrsig94	133	448	1124	2^{194}	KO	KO	0.030s
egfr104	193	748	2356	2^{320}	KO	KO	0.050s

1 [Inria Paris-Rocquencourt/Contraintes]
2 [LIP6/Move]

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

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

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 $\rightsquigarrow$ detection $\rightsquigarrow$ 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 $\rightsquigarrow$ detection $\rightsquigarrow$ 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.
In *IMA 134: Mathematical systems theory in biology, communication, and finance. Springer-Verleg, New York.*, page 12, 2002.

J. Ahmad, A. Richard, G. Bernot, J.-P. Comet, and O. Roux.
Delays in biological regulatory networks (BRN).
In *Proceedings of IWBRA 2006, LNCS*, page to appear, 2006.

Anil Aswani and Claire Tomlin.
Reachability algorithm for biological piecewise-affine hybrid systems.
In *HSCC'07: Proceedings of the 10th international conference on Hybrid systems*, pages 633–636, Berlin, Heidelberg, 2007. Springer-Verlag.

Grégory Batt, Calin Belta, and Ron Weiss.
Model checking genetic regulatory networks with parameter uncertainty.
In *HSCC'07: Proceedings of the 10th international conference on Hybrid systems*, pages 61–75, Berlin, Heidelberg, 2007. Springer-Verlag.

Grégory Batt, Calin Belta, and Ron Weiss.
Model checking liveness properties of genetic regulatory networks.
In *TACAS'07: Proceedings of the 13th international conference on Tools and algorithms for the construction and analysis of systems*, pages 323–338, Berlin, Heidelberg, 2007. Springer-Verlag.

G. Bernot, F. Cassez, J.-P. Comet, F. Delaplace, C. Müller, O. Roux, and O.(H) Roux.
Semantics of biological regulatory networks.
In Vincent Danos and Cosimo Laneve, editors, *BioConcur 2003*, Marseille (France), 2003. Elsevier's ENTCS series.

Ralf Blossey, Luca Cardelli, and Andrew Phillips.
Compositionality, stochasticity and cooperativity in dynamic models of gene regulation.
HFSP Journal, 2(1):17–28, February 2008.

L. Bortolussi and A. Policriti.
Hybrid systems and biology. continuous and discrete modeling for systems biology.
In P. Degano M. Bernardo and G. Zavattaro, editors, *Formal Methods For Computational System Biology*, volume 5016 of *LNCS*, pages 424–448. Springer, 2006.

Federica Ciocchetta and Jane Hillston.
Bio-pepa: A framework for the modelling and analysis of biological systems.
Theor. Comput. Sci., 410(33-34):3065–3084, 2009.

Claudine Chaouiya, Elisabeth Remy, and Denis Thieffry.
Petri net modelling of biological regulatory networks.
J. of Discrete Algorithms, 6(2):165–177, 2008.

Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine.
Rule-based modelling of cellular signalling.
In *CONCUR*, pages 17–41, 2007.

Lorenzo Dematté, Corrado Priami, and Alessandro Romanel.

Modelling and simulation of biological processes in blenx.
SIGMETRICS Perform. Eval. Rev., 35(4):32–39, 2008.

François Fages and Sylvain Soliman.
Abstract interpretation and types for systems biology.
Theor. Comput. Sci., 403(1):52–70, 2008.

T-A. Henzinger, P-H. Ho, and H. Wong-Toi.
HYTECH: A model checker for hybrid systems.
International Journal on Software Tools for Technology Transfer, 1(1–2):110–122, 1997.

John Heath, Marta Kwiatkowska, Gethin Norman, David Parker, and Oksana Tymchyshyn.
Probabilistic model checking of complex biological pathways.
Theoretical Computer Science, 391(3):239–257, 2008.

Paola Lecca and Corrado Priami.
Cell cycle control in eukaryotes: A biospi model.
Electr. Notes Theor. Comput. Sci., 180(3):51–63, 2007.

P. Lincoln and A. Tiwari.
Symbolic systems biology: Hybrid modeling and analysis of biological networks.
In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control HSCC*, volume 2993 of *LNCS*, pages 660–672. Springer, March 2004.

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.
In *CMSB '08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology*, pages 63–82, Berlin, Heidelberg, 2008. Springer-Verlag.

Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman.
Application of a stochastic name-passing calculus to representation and simulation of molecular processes.
Inf. Process. Lett., 80(1):25–31, 2001.

Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman.
On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology.
In Monika Heiner and Adeline Uhrmacher, editors, *CMSB'08: Proceedings of the fourth international conference on Computational Methods in Systems Biology*, volume 5307 of *Lecture Notes in Computer Science*, pages 251–268. Springer-Verlag, October 2008.