Modelling, Simulation, and Model Checking of Large-Scale Biological Regulatory Networks

Loïc Paulevé, **Morgan Magnin**, **Olivier Roux**

{loic.pauleve,morgan.magnin,olivier.roux}@irccyn.ec-nantes.fr

http://www.irccyn.ec-nantes.fr/~pauleve

IRCCyN / MeForBio team, Nantes (France)

LIX / AMIB team - 1st June 2011

IRCCyN laboratory, École Centrale de Nantes (France) - MeForBio Team

Loïc Paulevé

Morgan Magnin

Olivier Roux

Other people I work with

Andrew Phillips
Microsoft Research, Cambridge, UK
Stochastic simulation

Adrien Richard
I3S & CNRS, Nice, France
Biological Regulatory Networks

Overview

Computer science for systems biology

- Models for dynamical concurrent systems.
- Validation of the model / control of the system.
- We focus on Biological Regulatory Networks (BRNs).
- We introduce a new modelling framework: the Process Hitting.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Overview

Computer science for systems biology

- Models for dynamical concurrent systems.
- Validation of the model / control of the system.
- We focus on Biological Regulatory Networks (BRNs).
- We introduce a new modelling framework: the Process Hitting.

The Process Hitting [Paulevé, Magnin, Roux in TCSB 2011]

- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Stochastic and Time dimensions (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

Overview

Computer science for systems biology

- Models for dynamical concurrent systems.
- Validation of the model / control of the system.
- We focus on Biological Regulatory Networks (BRNs).
- We introduce a new modelling framework: the Process Hitting.

The Process Hitting [Paulevé, Magnin, Roux in TCSB 2011]

- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Stochastic and Time dimensions (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

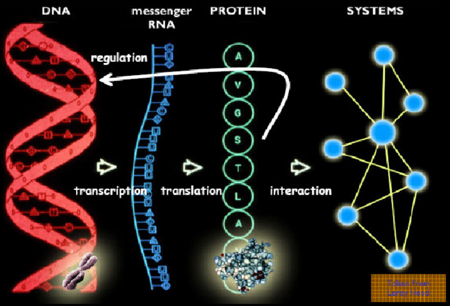
Large-scale model checking (dynamical properties)

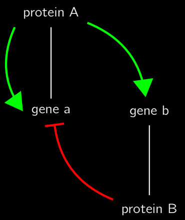
- Cope with state space explosion.
- Our approach: Static Analysis of the model.
- Static analysis by Abstract Interpretation.

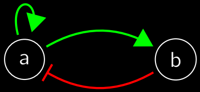
Outline

Biological Regulatory Networks (BRNs)

The interaction graph



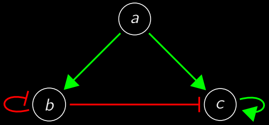




Loïc Paulevé, Morgan Magnin, Olivier Roux

Discrete Networks (BRNs)

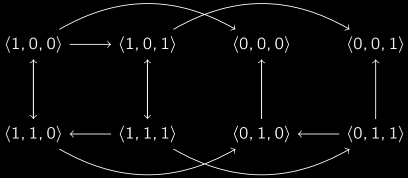
- Each component has a finite set of qualitative levels; e.g. $\{0, 1, 2\}$;
- may be seen as a quantization of the concentration of the component.

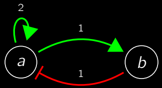


$f^a(x) = 0$

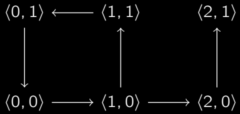
$f^b(x) = x[a] \wedge \neg x[b]$

$f^c(x) = \neg x[b] \wedge (x[a] \vee x[c])$





+ K



[René Thomas in Journal of Theoritical Biology, 1973] [A. Richard, J.-P. Comet, G. Bernot in Modern Formal Methods and Applications, 2006]

Hybrid Modelling

Continuous features governing discrete transitions

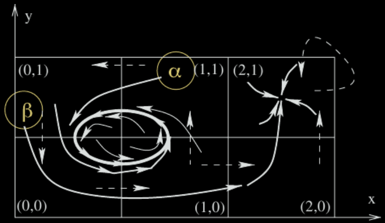
Introduce delays to actions

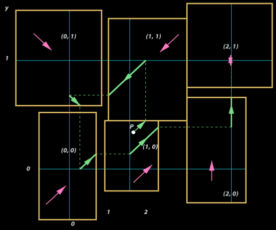
Stochastic Models

- Delays are random variables (generally exponential, i.e Markovian);
- ⇒ compute probabilities for observing behaviours.

Stochastic Petri Nets / π-calculus, etc.

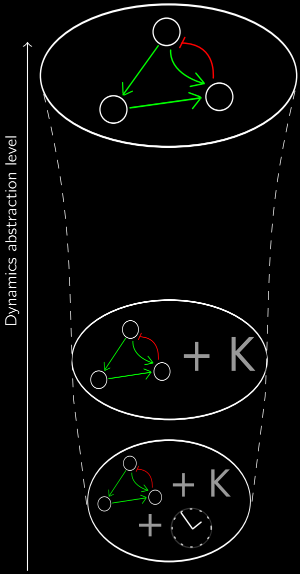
Timed Models





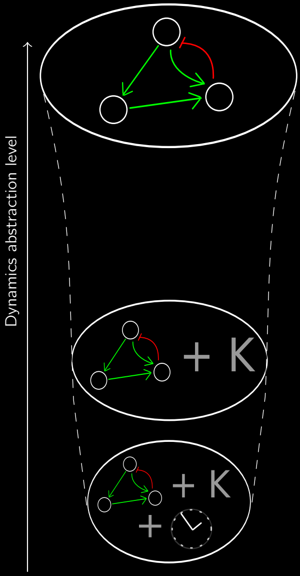
Timed / Hybrid Automata

Summary and Contribution



Loïc Paulevé, Morgan Magnin, Olivier Roux

Summary and Contribution



General Properties: Bounds on fix points #; sustained oscillations?; functionnalities; ...

Fix points (steady states)

Successive level reachabilities

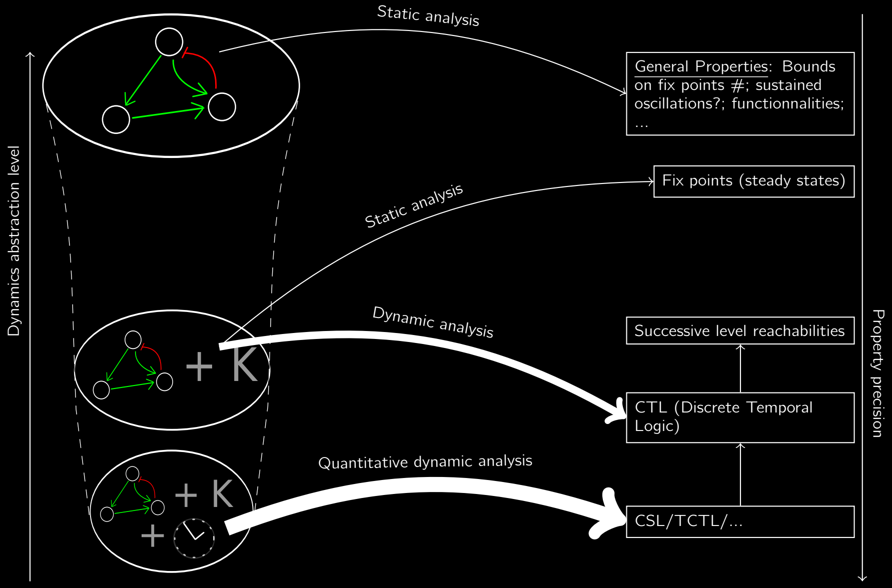
CTL (Discrete Temporal Logic)

CSL/TCTL/...

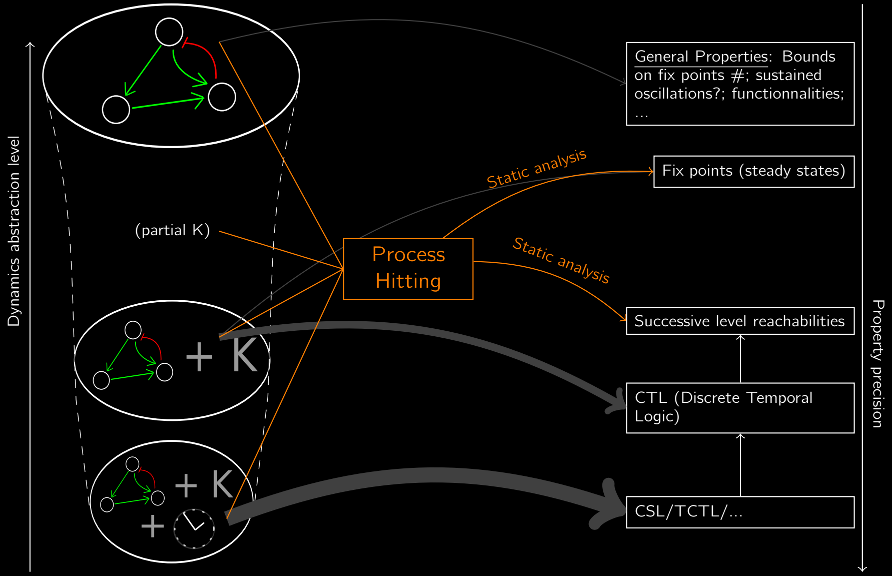
Property precision

Loïc Paulevé, Morgan Magnin, Olivier Roux

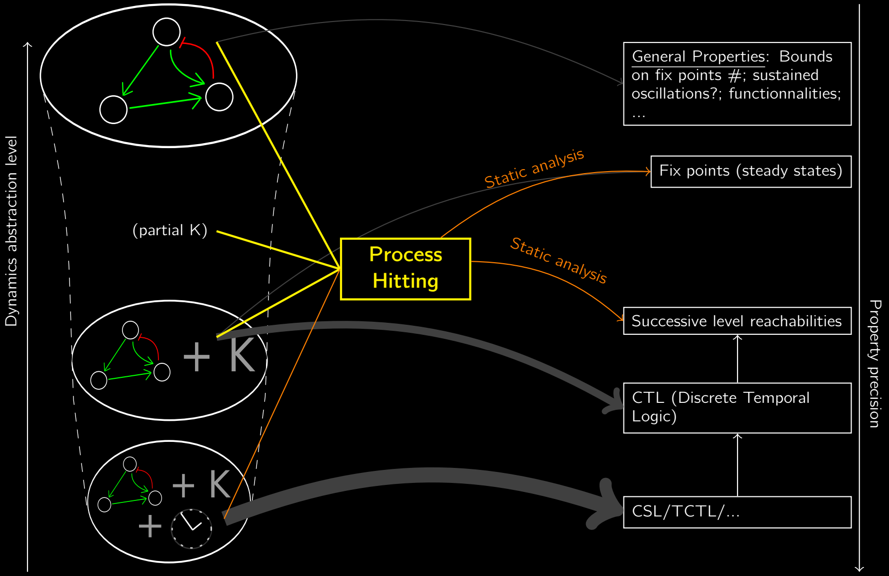
Summary and Contribution



Summary and Contribution

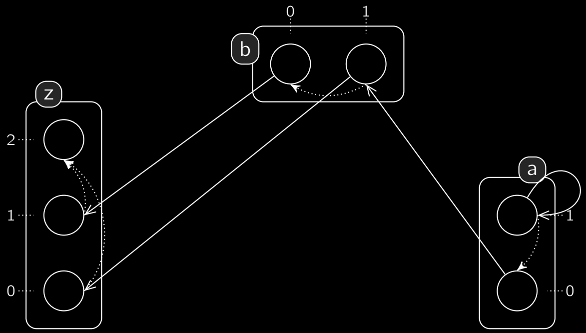


Outline


Dynamics abstraction level
(partial K)
+ K
+ K
+
Process Hitting
Static analysis
Static analysis
General Properties: Bounds on fix points #; sustained oscillations?; functionnalities; …
Fix points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/…
Property precision

The Process Hitting Framework

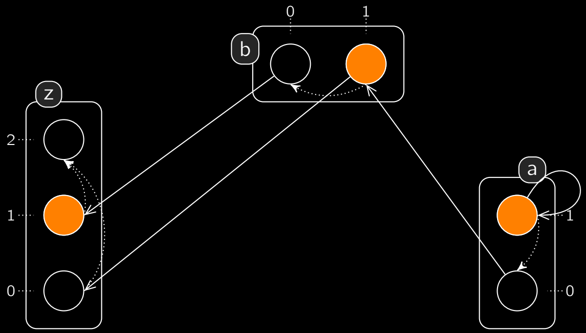
[Paulevé, Magnin, Roux in TCSB 2011]



- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

The Process Hitting Framework

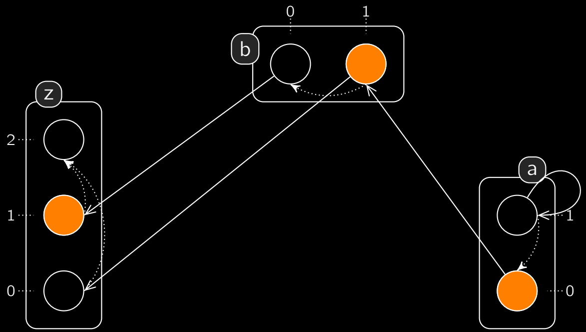
[Paulevé, Magnin, Roux in TCSB 2011]



- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

The Process Hitting Framework

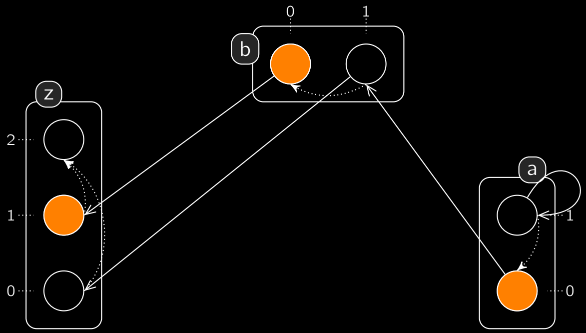
[Paulevé, Magnin, Roux in TCSB 2011]



- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

The Process Hitting Framework

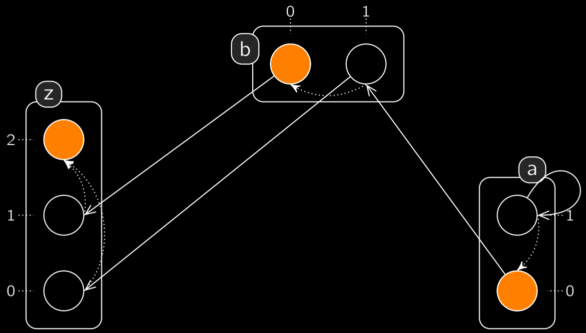
[Paulevé, Magnin, Roux in TCSB 2011]



- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines (CFSM).

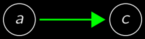
The Process Hitting Framework

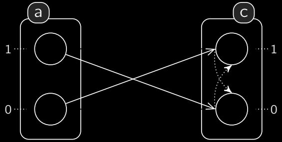
[Paulevé, Magnin, Roux in TCSB 2011]



- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

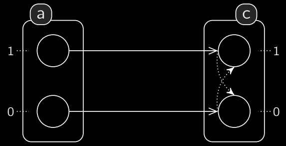
From BRNs to Process Hittings





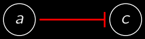
[Paulevé, Magnin, Roux in Trans. in Computational Systems Biology, 2011]

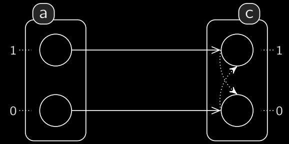
From BRNs to Process Hittings

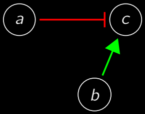


[Paulevé, Magnin, Roux in Trans. in Computational Systems Biology, 2011]

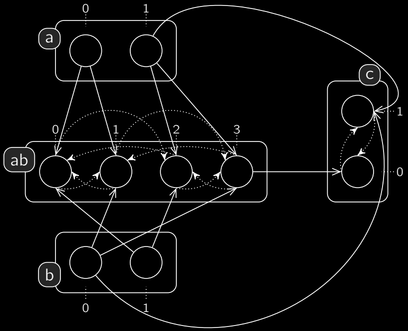
From BRNs to Process Hittings





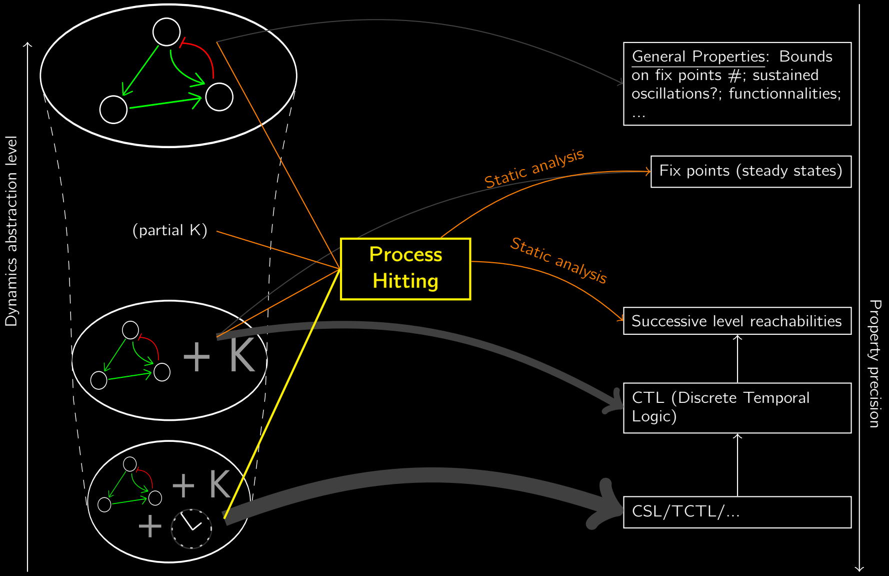


$c = \neg a \wedge b$



[Paulevé, Magnin, Roux in Trans. in Computational Systems Biology, 2011]

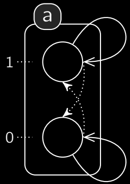
Outline

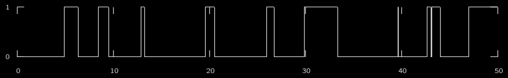

Dynamics abstraction level
(partial K)
+ K
+ K
+
Process Hitting
Static analysis
Static analysis
General Properties: Bounds on fix points #; sustained oscillations?; functionnalities; ...
Fix points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

Stochasticity Absorption Factor

[Paulevé, Magnin, Roux in IEEE TSE, 2010]

- Stochastic dimension is prominent... but no precise time features:
- (Markov) Exponential distribution: mean r^{-1}; variance r^{-2}.
- At our level of abstraction, we need to tune time features.



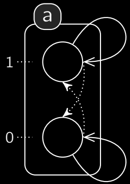


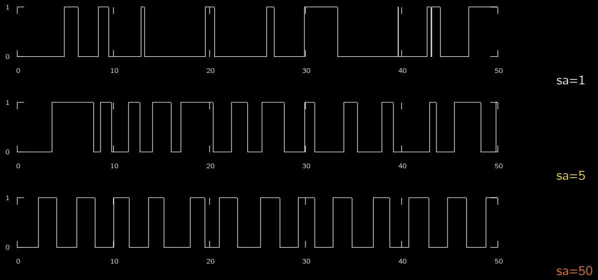
sa=1

Stochasticity Absorption Factor

[Paulevé, Magnin, Roux in IEEE TSE, 2010]

- Stochastic dimension is prominent... but no precise time features:
- (Markov) Exponential distribution: mean r^{-1}; variance r^{-2}.
- At our level of abstraction, we need to tune time features.
- Provide a stochasticity absorption factor sa:
- duration follows the sum of sa exponentials of rate $r.sa$;
- mean r^{-1}; variance $r^{-2}sa^{-1}$ (Erlang distribution).

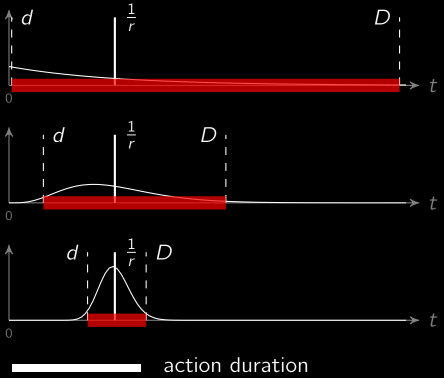




Stochastic and Time Parameters

[Paulevé, Magnin, Roux in IEEE TSE, 2010]

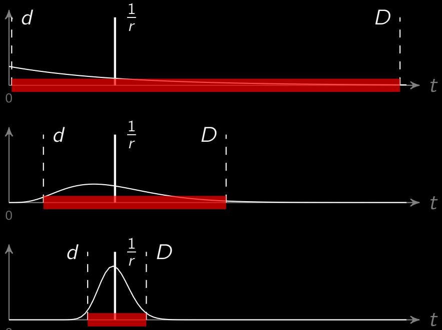
- Specify either (r, sa), or its firing interval $[d; D]$,
- which is the confidence interval at confidence coefficient $1 - \alpha$.
- We have estimators to translate $[d; D]$ to (r, sa).



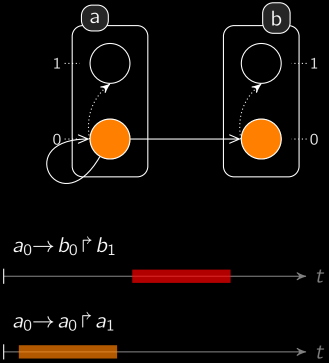
Stochastic and Time Parameters

[Paulevé, Magnin, Roux in IEEE TSE, 2010]

- Specify either (r, sa), or its firing interval $[d; D]$,
- which is the confidence interval at confidence coefficient $1 - \alpha$.
- We have estimators to translate $[d; D]$ to (r, sa).



action duration



$\Rightarrow$ b_1 is reached at a very low probability.

Simulation and Model Checking

Stochastic Model Checking

- Translation from the Erlang stochastic π-calculus to PRISM [Paulevé, Magnin, Roux in IEEE TSE, 2010].
- Applies to the Process Hitting as well.
- Not tractable with large stochasticiy absorption factors;
- but there is hope in symmetry reductions, or abstractions of sequences of transitions, or ...

Simulation

- Non-Markovian simulation using the
- Generic abstract machine for stochastic process calculi [Paulevé, Youssef, Lakin, Phillips at CMSB 2010].
- Process Hitting simulator implemented in PINT.

Challenge: scalable inference of stochastic and time parameters...

Simulation and Model Checking

Stochastic Model Checking

- Translation from the Erlang stochastic π-calculus to PRISM [Paulevé, Magnin, Roux in IEEE TSE, 2010].
- Applies to the Process Hitting as well.
- Not tractable with large stochasticiy absorption factors;
- but there is hope in symmetry reductions, or abstractions of sequences of transitions, or ...

Simulation

- Non-Markovian simulation using the
- Generic abstract machine for stochastic process calculi [Paulevé, Youssef, Lakin, Phillips at CMSB 2010].
- Process Hitting simulator implemented in PINT.

Challenge: scalable inference of stochastic and time parameters...
... still open; prior need for scalable qualitative analyses.

Static Analysis of Process Hitting

- Static analysis: derive dynamical properties without executing the model.
- Aim at coping with the combinatorial explosion of the dynamics.
- Several approaches: topological analysis, control flow analysis, constraints, abstractions, etc.

BRNs analysis

- Topological analyses of the interaction graph: bounds on fix points #; possibility of observing sustained oscillations, etc.
- Decision diagrams to derive fix points (from full BRN specification).

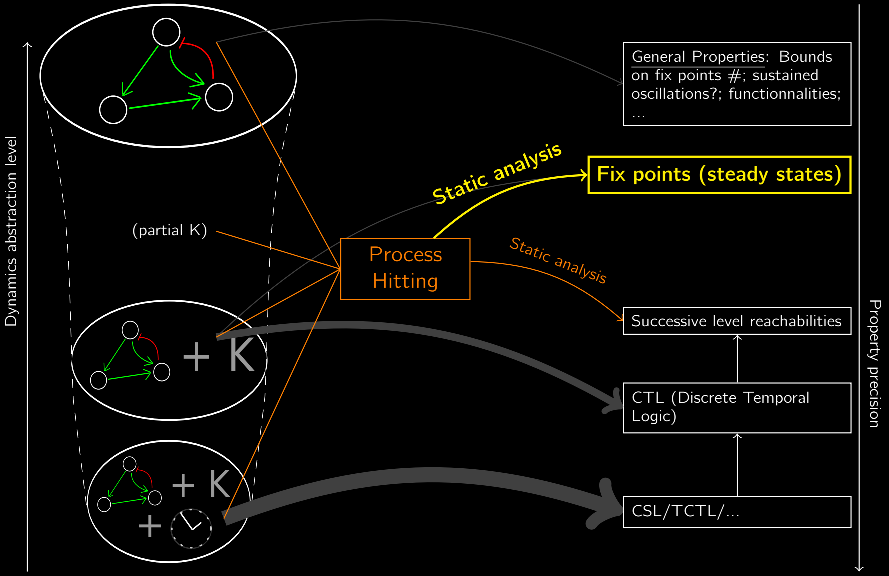
Our Contributions using Process Hitting

- Topological analysis: complete characterisation of fix points.
- Abstract interpretation: over- and under-approximations of reachability properties.

$\Longrightarrow$ brings new insight for deriving efficiently more precise dynamical properties from BRNs.

Loïc Paulevé, Morgan Magnin, Olivier Roux

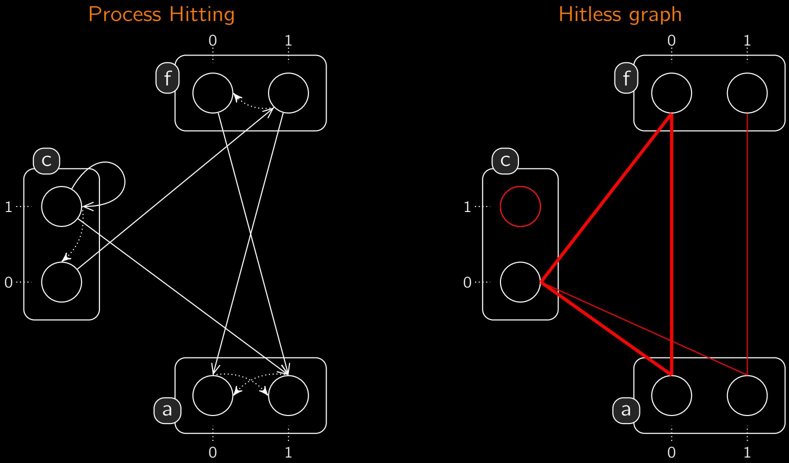
Outline



Loïc Paulevé, Morgan Magnin, Olivier Roux

Fix Points

[Paulevé, Magnin, Roux in TCSB 2011]

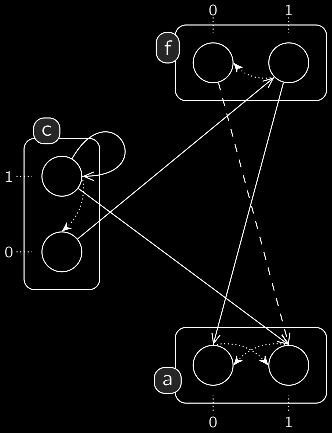


n-cliques are fix points

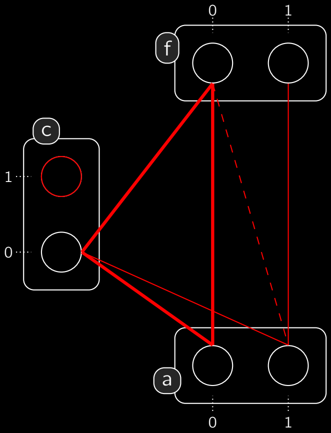
Fix Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



Hitless graph

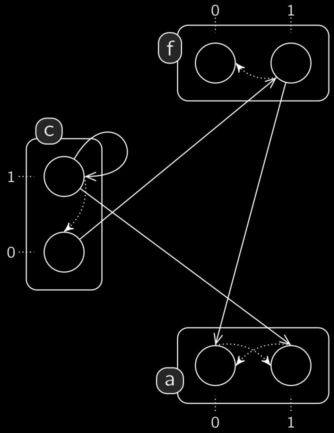


n-cliques <u>are</u> fix points

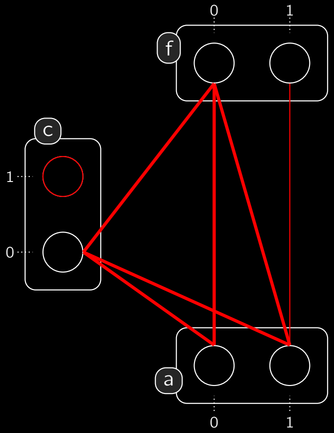
Fix Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting

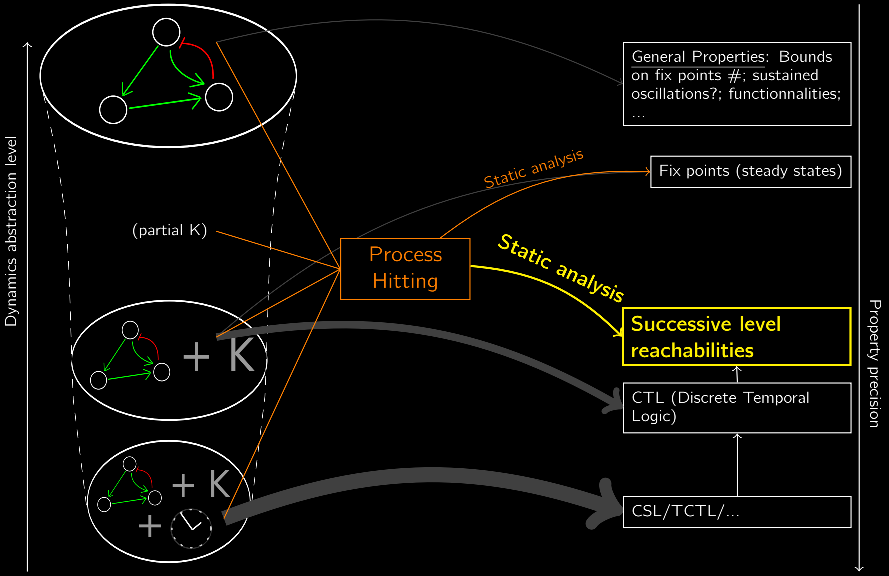


Hitless graph



n-cliques are fix points

Outline



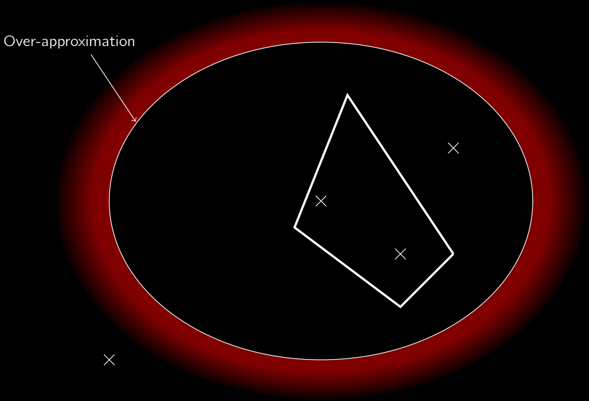
Loïc Paulevé, Morgan Magnin, Olivier Roux

Approximation of Reachability Properties

- Successive reachability of processes (reach a_0 then b_1 then ...);
- Approach using abstract interpretation techniques;
- Results in both over- and under-approximations;
- Limited complexity at the cost of potentially being inconclusive.

Approximation of Reachability Properties

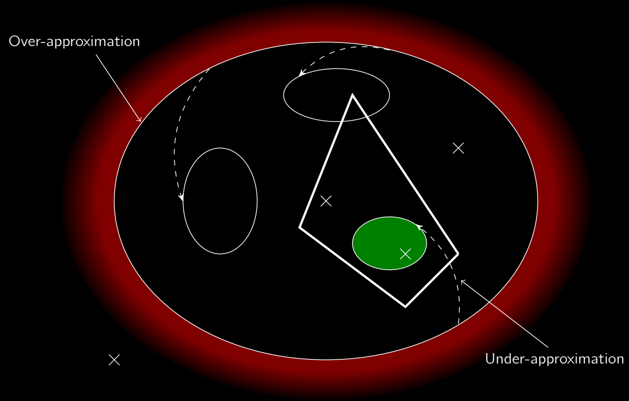
- Successive reachability of processes (reach a_0 then b_1 then ...);
- Approach using abstract interpretation techniques;
- Results in both over- and under-approximations;
- Limited complexity at the cost of potentially being inconclusive.



[Paulevé, Magnin, Roux at SASB 2010 + MSCS submitted]

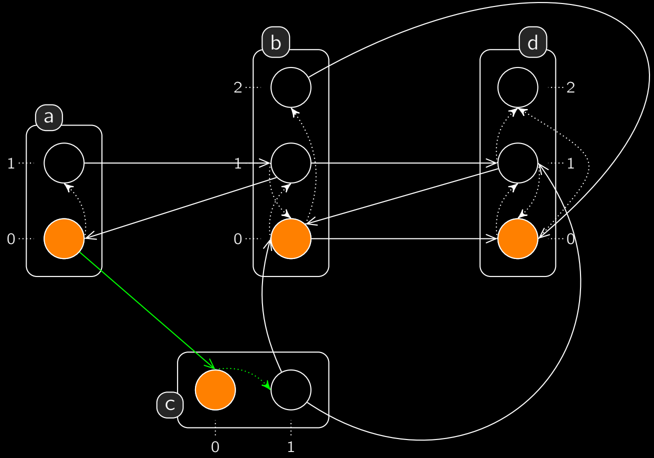
Approximation of Reachability Properties

- Successive reachability of processes (reach a_0 then b_1 then ...);
- Approach using abstract interpretation techniques;
- Results in both over- and under-approximations;
- Limited complexity at the cost of potentially being inconclusive.



[Paulevé, Magnin, Roux at SASB 2010 + MSCS submitted]

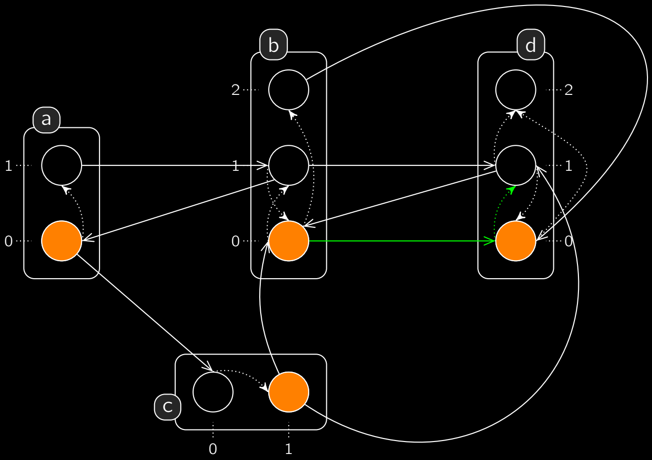
Scenarios



$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$

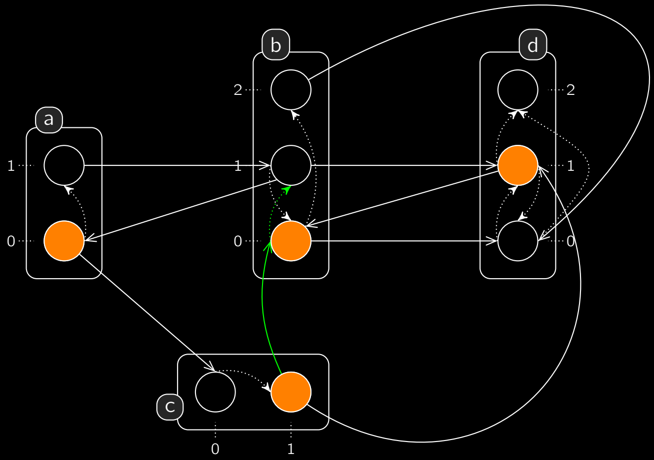
Loïc Paulevé, Morgan Magnin, Olivier Roux

Scenarios



$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$

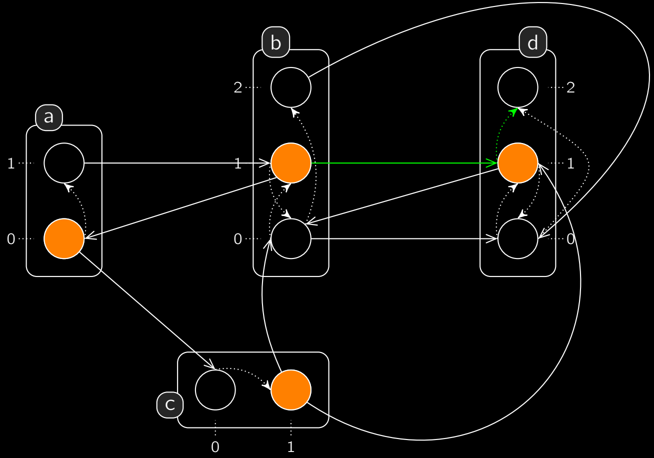
Scenarios



$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$

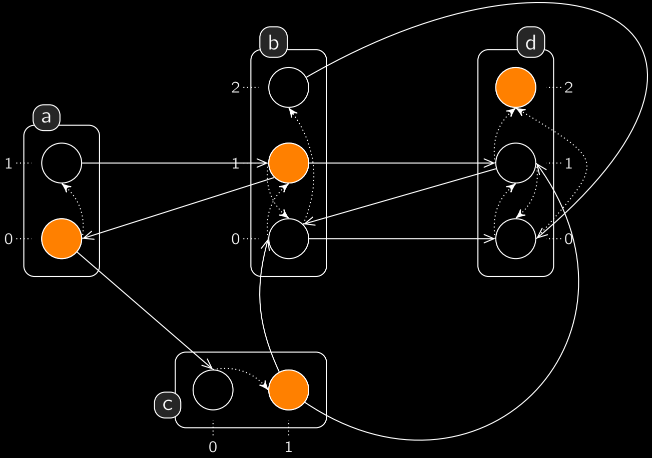
Loïc Paulevé, Morgan Magnin, Olivier Roux

Scenarios



$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$

Scenarios



$$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$$

Abstract Interpretation of Scenarios

[Paulevé, Magnin, Roux at SASB 2010 + MSCS submitted]

Scenarios – Successively playable actions.

- E.g. $\delta = a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$.

Context — For each sort, subset of initial processes.

- E.g. $\varsigma = \langle a_0, b_0, b_2, c_0, d_0 \rangle$.

Abstract Interpretation of Scenarios

[Paulevé, Magnin, Roux at SASB 2010 + MSCS submitted]

Scenarios – Successively playable actions.

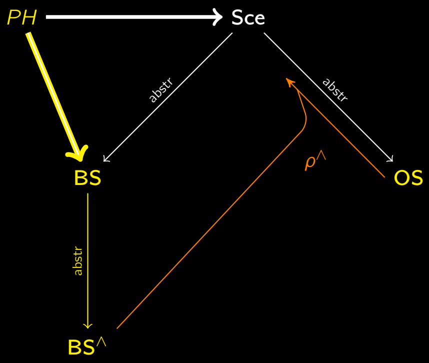
- E.g. $\delta = a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$.

Context — For each sort, subset of initial processes.

- E.g. $\varsigma = \langle a_0, b_0, b_2, c_0, d_0 \rangle$.

Overall approach

- 2 complementary abstractions;
- Bounce Sequences **BS**;
- Objective Sequences **OS**;
- Concretization: $\gamma_\varsigma : \mathbf{OS} \mapsto \wp(\mathbf{Sce})$;
- Refinements: $\rho : \mathbf{OS} \mapsto \wp(\mathbf{OS})$;
- $\gamma_\varsigma(\omega) = \gamma_\varsigma(\rho(\omega))$.



Two Complementary Abstractions

$$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by Objective Sequences

- $c_0 \upharpoonright^* c_1 :: d_0 \upharpoonright^* d_1 :: b_0 \upharpoonright^* b_1 :: d_1 \upharpoonright^* d_2$;

Two Complementary Abstractions

$$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$$

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$

Two Complementary Abstractions

$$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by Objective Sequences

- $c_0 \upharpoonright^* c_1 :: d_0 \upharpoonright^* d_1 :: b_0 \upharpoonright^* b_1 :: d_1 \upharpoonright^* d_2$;
- $b_0 \upharpoonright^* b_1 :: d_0 \upharpoonright^* d_2$
- $d_0 \upharpoonright^* d_2$, ...

Loïc Paulevé, Morgan Magnin, Olivier Roux

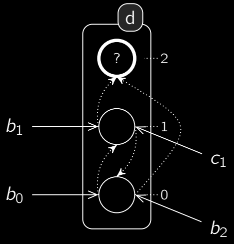
Two Complementary Abstractions

$$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by Objective Sequences

- $c_0 \upharpoonright^* c_1 :: d_0 \upharpoonright^* d_1 :: b_0 \upharpoonright^* b_1 :: d_1 \upharpoonright^* d_2$;
- $b_0 \upharpoonright^* b_1 :: d_0 \upharpoonright^* d_2$
- $d_0 \upharpoonright^* d_2$, ...

Abstraction by Bounce Sequences



E.g.: $b_0 \rightarrow d_0 \upharpoonright d_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$ ($d_0 \upharpoonright^* d_2$)

Loïc Paulevé, Morgan Magnin, Olivier Roux

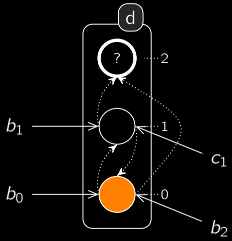
Two Complementary Abstractions

$$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$$

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$
- $d_0 \Rsh^* d_2$, ...

Abstraction by Bounce Sequences



E.g.: $b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2$ $(d_0 \Rsh^* d_2)$
$\Rightarrow$ can be computed off-line:

- $\mathbf{BS}(d_0 \Rsh^* d_2) = \{b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2, b_2 \rightarrow d_0 \Rsh d_2]\}$;
- $\mathbf{BS}^{\wedge}(d_0 \Rsh^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.

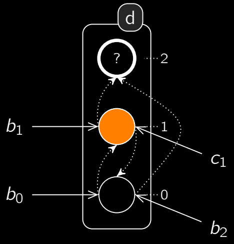
Two Complementary Abstractions

$$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$$

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$
- $d_0 \Rsh^* d_2$, ...

Abstraction by Bounce Sequences



E.g.: $b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2$ ($d_0 \Rsh^* d_2$)
⇒ can be computed off-line:

- $\mathbf{BS}(d_0 \Rsh^* d_2) = \{b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2,$
  $b_2 \rightarrow d_0 \Rsh d_2]\}$;
- $\mathbf{BS}^\wedge(d_0 \Rsh^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.
- $\mathbf{BS}(d_1 \Rsh^* d_2) = \{b_1 \rightarrow d_1 \Rsh d_2,$
  $c_1 \rightarrow d_1 \Rsh d_0 :: b_2 \rightarrow d_0 \Rsh d_2\}$;
- $\mathbf{BS}^\wedge(d_1 \Rsh^* d_2) = \{\{b_1\}, \{b_2, c_1\}\}$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Objective Sequence Refinements

$$\gamma_\varsigma(\omega) = \{\delta \in \textbf{Sce} \mid \omega \text{ abstracts } \delta \wedge \mathrm{support}(\delta) \subseteq \varsigma\}.$$

Idea: the more details we know, the better $\gamma_\varsigma(\omega)$ should be understood.

Objective Sequence Refinements

$$\gamma_\varsigma(\omega) = \{\delta \in \textbf{Sce} \mid \omega \text{ abstracts } \delta \wedge \mathrm{support}(\delta) \subseteq \varsigma\}.$$

Idea: the more details we know, the better $\gamma_\varsigma(\omega)$ should be understood.

Objective Refinement by $\mathbf{BS}^\wedge$: $\rho^\wedge$

$\mathbf{Obj} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$d_0 \Rsh^* d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \Rsh^* b_0 :: b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_1 :: b_1 \Rsh^* b_0 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_2 :: d_0 \Rsh^* d_2$
$\gamma_\varsigma(d_0 \Rsh^* d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0 \Rsh^* d_2, \mathbf{BS}^\wedge(d_0 \Rsh^* d_2)))$

Objective Sequence Refinements

$$\gamma_\varsigma(\omega) = \{\delta \in \textbf{Sce} \mid \omega \text{ abstracts } \delta \wedge \mathrm{support}(\delta) \subseteq \varsigma\}.$$

Idea: the more details we know, the better $\gamma_\varsigma(\omega)$ should be understood.

Objective Refinement by $\mathbf{BS}^\wedge$: $\rho^\wedge$

$\mathbf{Obj} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$d_0 \Rsh^* d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \Rsh^* b_0 :: b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_1 :: b_1 \Rsh^* b_0 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_2 :: d_0 \Rsh^* d_2$
$\gamma_\varsigma(d_0 \Rsh^* d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0 \Rsh^* d_2, \mathbf{BS}^\wedge(d_0 \Rsh^* d_2)))$

Generalization to OS refinements: $\widetilde{\rho}$

$\mathbf{OS} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$\omega, \mathbf{BS}^\wedge$	$\mathrm{interleave}\begin{pmatrix}\omega' \\ \omega_{1..n-1}\end{pmatrix} :: \omega_{n..\lvert\omega\rvert}$ where $n \in \mathbb{I}^\omega$ and $\omega' :: \omega_n \in \rho^\wedge(\omega_n, \mathbf{BS}^\wedge(\omega_n))$
$\gamma_\varsigma(\omega)$	$= \gamma_\varsigma(\widetilde{\rho}(\omega, \mathbf{BS}^\wedge))$

Loïc Paulevé, Morgan Magnin, Olivier Roux

Objective Sequence Concretizability

a.k.a. Process Reachability Problem

$\gamma_s(a_i \Rsh^* a_j :: \ldots :: z_k \Rsh^* z_l) \neq \emptyset$?

CTL: $EF(s[a] = a_j \wedge EF(\cdots \wedge EF(s'[z] = z_l) \ldots))$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Objective Sequence Concretizability

a.k.a. Process Reachability Problem

$\gamma_\varsigma(a_i \upharpoonright^* a_j :: \ldots :: z_k \upharpoonright^* z_l) \neq \emptyset$?

CTL: $EF(s[a] = a_j \wedge EF(\cdots \wedge EF(s'[z] = z_l) \ldots))$.

Over-approximation

- Un-ordered: $\gamma_\varsigma(\omega) \neq \emptyset \Longrightarrow \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge$: **Obj** $\mapsto \wp(\mathbf{OS})$).

Objective Sequence Concretizability

a.k.a. Process Reachability Problem

$$\gamma_\varsigma(a_i \upharpoonright^* a_j :: \ldots :: z_k \upharpoonright^* z_l) \neq \emptyset?$$

CTL: $EF(s[a] = a_j \wedge EF(\cdots \wedge EF(s'[z] = z_l)\ldots))$.

Over-approximation

- Un-ordered: $\gamma_\varsigma(\omega) \neq \emptyset \implies \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge : \mathbf{Obj} \mapsto \wp(\mathbf{OS})$).
- Ordered: $\gamma_\varsigma(a_i \upharpoonright^* a_j :: \omega) \neq \emptyset \implies \gamma_{max\varsigma}(\omega) \neq \emptyset$.

Objective Sequence Concretizability

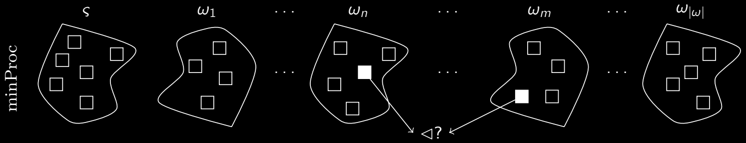
a.k.a. Process Reachability Problem

$\gamma_\varsigma(a_i \Rsh^* a_j :: \ldots :: z_k \Rsh^* z_l) \neq \emptyset?$

CTL: $EF(s[a] = a_j \wedge EF(\cdots \wedge EF(s'[z] = z_l) \ldots))$.

Over-approximation

- Un-ordered: $\gamma_\varsigma(\omega) \neq \emptyset \Longrightarrow \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge : \mathbf{Obj} \mapsto \wp(\mathbf{OS})$).
- Ordered: $\gamma_\varsigma(a_i \Rsh^* a_j :: \omega) \neq \emptyset \Longrightarrow \gamma_{max\varsigma}(\omega) \neq \emptyset$.
- Process occurrences order constaints ($a_i \lhd a_j$):



Objective Sequence Concretizability

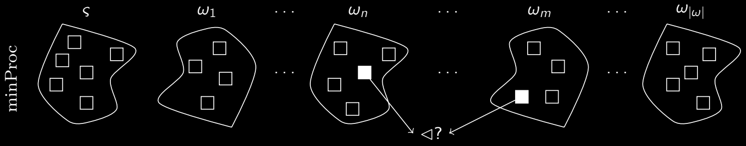
a.k.a. Process Reachability Problem

$\gamma_\varsigma(a_i \Rsh^* a_j :: \ldots :: z_k \Rsh^* z_l) \neq \emptyset$?

CTL: $EF(s[a] = a_j \wedge EF(\cdots \wedge EF(s'[z] = z_l) \ldots))$.

Over-approximation

- Un-ordered: $\gamma_\varsigma(\omega) \neq \emptyset \implies \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge$: **Obj** $\mapsto \wp(\mathbf{OS})$).
- Ordered: $\gamma_\varsigma(a_i \Rsh^* a_j :: \omega) \neq \emptyset \implies \gamma_{max\varsigma}(\omega) \neq \emptyset$.
- Process occurrences order constaints ($a_i \lhd a_j$):



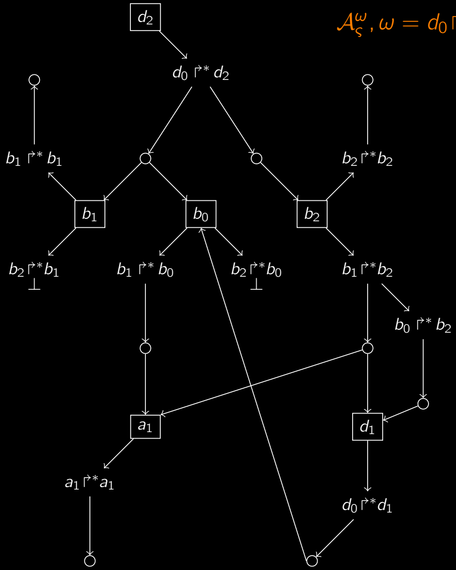
Under-approximation

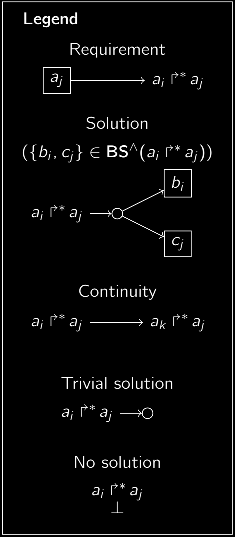
- Sufficient condition: impose a particular structure of scenarios.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Implementation of Over-approximations

$\mathcal{A}^{\omega}_{\varsigma}, \omega = d_0 \Rsh^* d_2, \varsigma = \langle a_1, b_1, b_2, c_1, d_0 \rangle$





Complexities

Data structures

- Computing **BS** is exponential in the number of processes in a single sort.
- Computing $\mathbf{BS}^{\wedge}$ is faster, but still exponential.
- The size of $\mathcal{A}_\varsigma^\omega$ is $\approx$ polynomial in the number of processes.

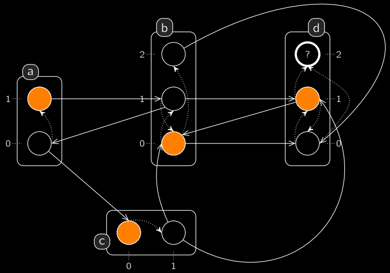
Analyses

- Over-approximations are polynomial in the size of $\mathcal{A}_\varsigma^\omega$.
- Under-approximation is polynomial in the size of $\mathcal{A}_\varsigma^\omega$; and can be applied a number of times exponential in the number of solutions of a single objective.

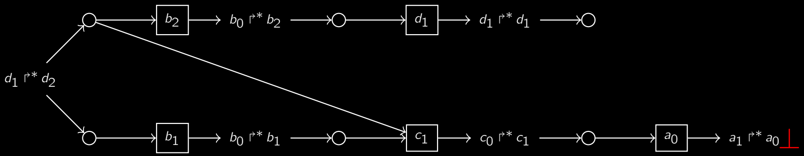
$\Longrightarrow$ efficient with a small number of processes per sort; while a very large number of sorts can be handled.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Un-ordered Over-approximation

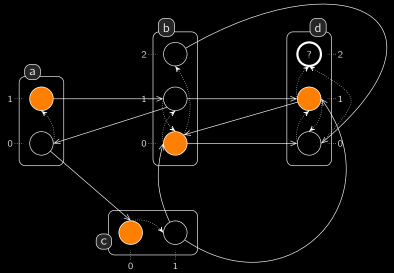


- $\gamma_\varsigma(\omega) \neq \emptyset \implies \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge : \mathbf{Obj} \mapsto \wp(\mathbf{OS})$).
- Necessary condition: there always exists a solution ending with a trivial objective.

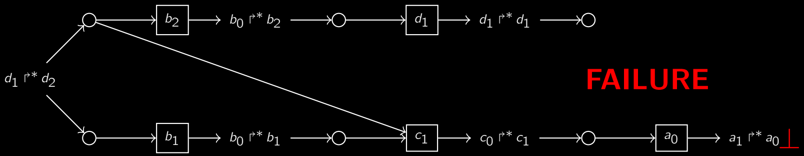


Loïc Paulevé, Morgan Magnin, Olivier Roux

Un-ordered Over-approximation

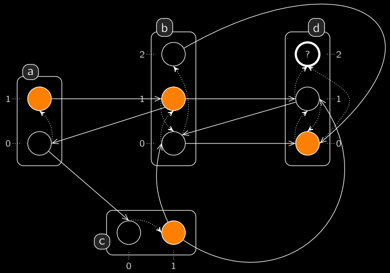


- $\gamma_\varsigma(\omega) \neq \emptyset \Longrightarrow \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge : \mathbf{Obj} \mapsto \wp(\mathbf{OS})$).
- Necessary condition: there always exists a solution ending with a trivial objective.

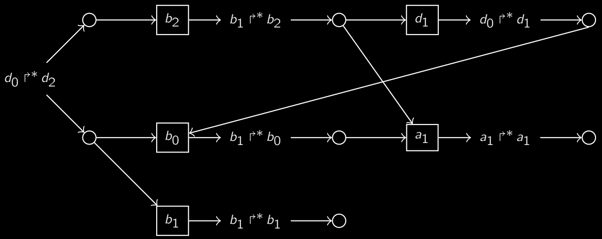


Loïc Paulevé, Morgan Magnin, Olivier Roux

Un-ordered Over-approximation



- $\gamma_\varsigma(\omega) \neq \emptyset \Longrightarrow \gamma_\varsigma(\omega_n) \neq \emptyset, \forall n \in \mathbb{I}^\omega$ (recursive reasoning with $\rho^\wedge : \mathbf{Obj} \mapsto \wp(\mathbf{OS})$).
- Necessary condition: there always exists a solution ending with a trivial objective.



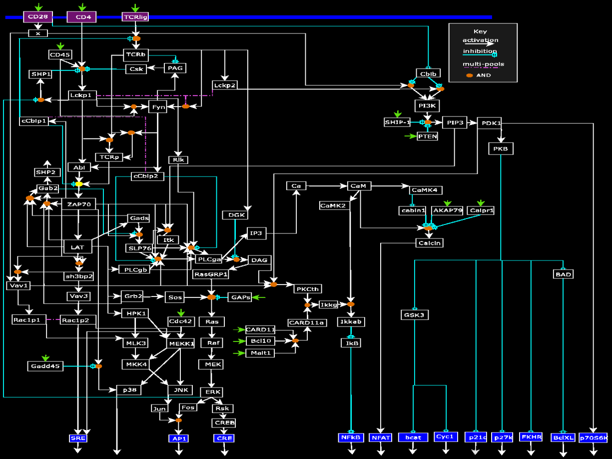
OK

Loïc Paulevé, Morgan Magnin, Olivier Roux

Outline

T-Cell Receptor Signalling Pathway

(94 components)



[Saez-Rodriguez, *et al.* in PLoS Comput Biol, 07]

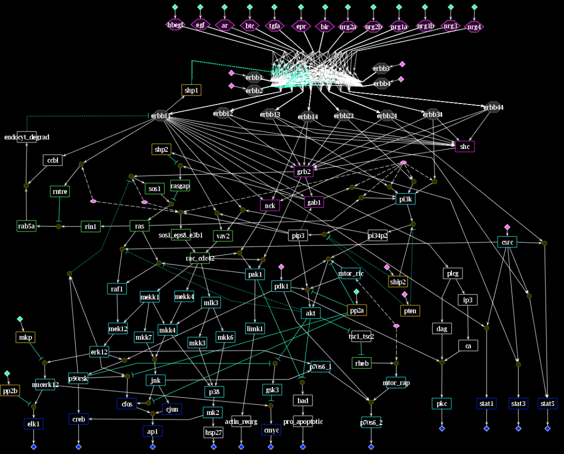
Process Hitting

133 sorts,
448 processes,
1124 actions:
$\approx 2 \cdot 10^{58}$ states.

Reachability analysis always conclusive; around 0.01s (compared to *libddd*: out of memory). [http://ddd.lip6.fr]

EGFR/ErbB Signalling

(104 components)



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

Process Hitting

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

Reachability analysis always conclusive; around 0.05*s* (compared to *libddd*: out of memory). [http://ddd.lip6.fr]

Outline

Conclusion

The Process Hitting

- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Generic tuning of time features within stochastic models (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

Conclusion

The Process Hitting

- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Generic tuning of time features within stochastic models (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

BRNs Modelling using Process Hitting

- Approach by abstraction refinements: from the generalized dynamics of the interation graph to the construction of cooperations;
- Support partially specified BRNs (largest dynamics used).
- Stochastic and temporal specifications: large-scale simulation + low-scale model checking.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Conclusion

The Process Hitting

- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Generic tuning of time features within stochastic models (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

BRNs Modelling using Process Hitting

- Approach by abstraction refinements: from the generalized dynamics of the interation graph to the construction of cooperations;
- Support partially specified BRNs (largest dynamics used).
- Stochastic and temporal specifications: large-scale simulation + low-scale model checking.

Static Analysis of Process Hitting

- Fix points by topological analysis;
- Very efficient over- and under-approximations of process reachability;
- Extract necessary processes for achieving reachabilities: towards control.
- Brings new insight to derive precise dynamical properties from BRNs.

Outlook

Derive more properties

- Characterisation of attractors;
- Verification for sustained oscillations;
- etc.

Outlook

Derive more properties

- Characterisation of attractors;
- Verification for sustained oscillations;
- etc.

Towards Quantitative Analysis

- Quantification of dynamics: reachability probabilities, expected times, etc.
- Extension to the Process Hitting with Priorities (e.g., two classes of actions: instantaneous and non-instantaneous);
- Idea: detect key concurrencies leading to the satisfication or non-satisfaction of a properties.

Outlook

Derive more properties

- Characterisation of attractors;
- Verification for sustained oscillations;
- etc.

Towards Quantitative Analysis

- Quantification of dynamics: reachability probabilities, expected times, etc.
- Extension to the Process Hitting with Priorities (e.g., two classes of actions: instantaneous and non-instantaneous);
- Idea: detect key concurrencies leading to the satisfication or non-satisfaction of a properties.

Application to BRNs

- Address bigger BRNs (E. Coli, etc.);
- Focus on properties of interest for BRNs analysis;
- Suggestions are very welcome.