Modelling and Checking 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 / MOVES Team, Nantes (France)

LIF / MoVe Working Group
18th November 2010

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.

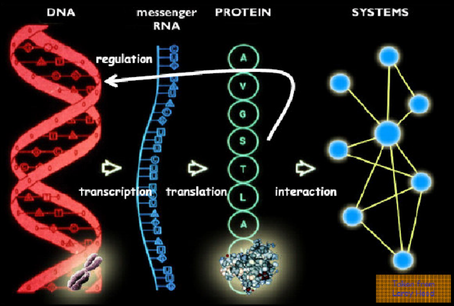
Large scale model checking (dynamical properties)

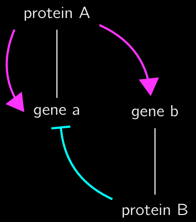
- Cope with state space explosion.
- Our approach: static analysis of the model.

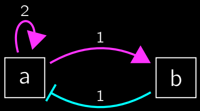
Outline

1. Introduction to BRNs; brief SotA.
2. The Process Hitting + static listing of fix points of dynamics.
3. Abstract interpretation and static analysis of reachability properties.
4. Applications to large BRNs + on-going work.

Biological Regulatory Networks (BRNs)





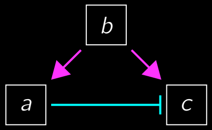


Loïc Paulevé, Morgan Magnin, Olivier Roux

Biological Regulatory Networks (BRNs)

- Relates components with actions of activation and inhibition;
- each component has a finite number of ordered states;
- the next state of a component is function of its activators/inhibitors.

Interaction graph



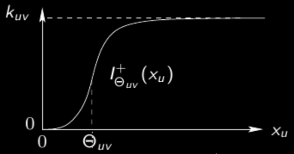
Cooperations

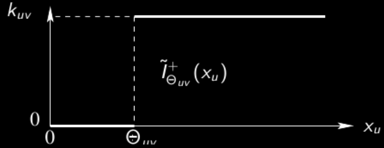
For instance (boolean case): $c = \neg a \wedge b$.

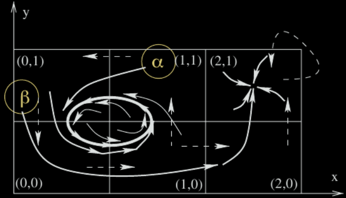
[René Thomas in Journal of Theoritical Biology, 1973] [A. Richard, J.-P. Comet, G. Bernot in Modern Formal Methods and Applications, 2006]

Loïc Paulevé, Morgan Magnin, Olivier Roux

Temporal features





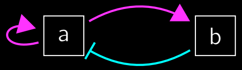


Some current work

- Time(d), Stochastic Petri Nets [Heiner],
- Continuous-time Markov Chains (PRISM) [Kwiatkowska, Parker],
- Biocham [Fages], Kappa [Danos, Feret, Fontana, Krivine],
- Timed Automata [Siebert, Bockmayr],
- Linear Hybrid Automata [Ahmad, Roux].

Dynamics Properties from Interaction Graph

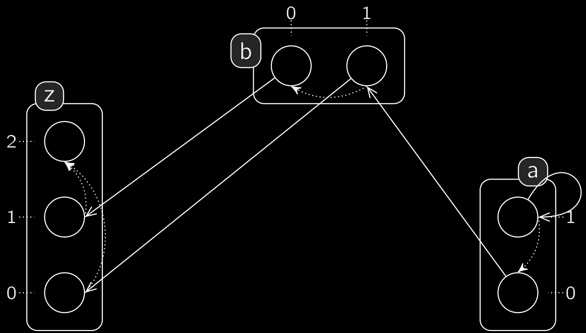
An interaction graph can describe a large set of different dynamics (think to different cooperations between components).



Mathematical work on relationship between the interaction graph and dynamical properties:

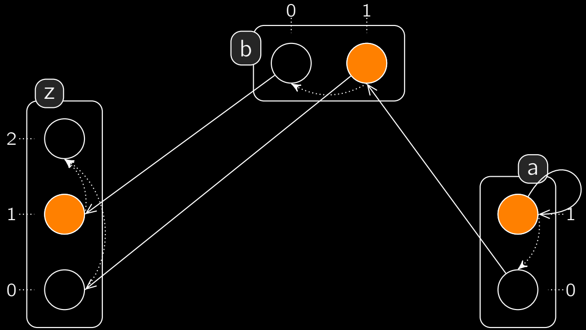
- Multi-stationnarity requires a positive circuit (René Thomas conjecture) [Soule in ComPlexUs, 2003] [Richard, Comet in Discrete Appl. Math., 2007].
- Sustained oscillations requires a negative circuit (René Thomas conjecture) [Remy, *et al.* in Adv. Appl. Math., 2008] [Richard in Adv. Appl. Math., 2010].
- The maximum number of fixed points can be characterized [Aracena in Bul. of Mathematical Biology, 2008]; [Richard in Discrete Appl. Math., 2009].
- Topological Fixed Points [Paulevé, Richard in CRAS 2010].

The Process Hitting Framework



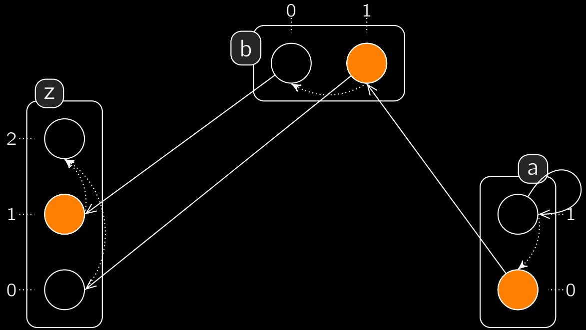
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

The Process Hitting Framework



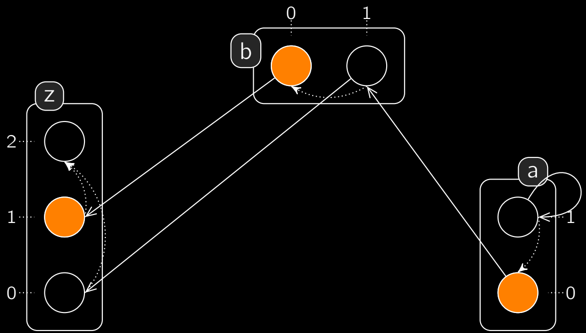
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

The Process Hitting Framework



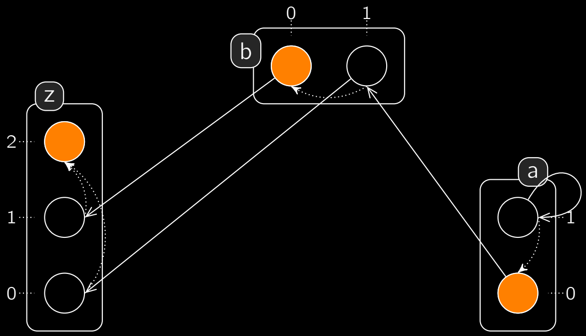
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

The Process Hitting Framework



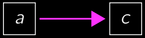
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

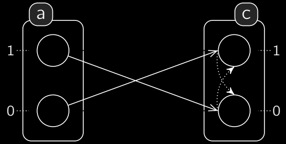
The Process Hitting Framework

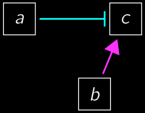


- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, . . . ;
- Restriction of Communicating Finite-State Machines (CFSM).

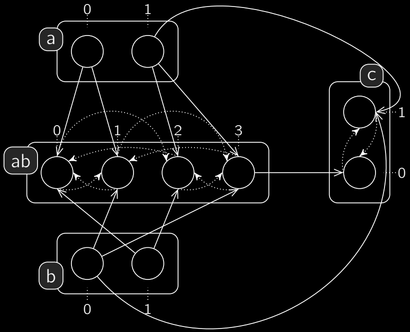
From BRNs to Process Hittings





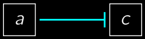


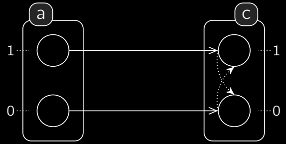
$c = \neg a \wedge b$

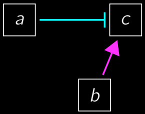


[Paulevé, Magnin, Roux in Trans. in Computational Systems Biology, 2010]

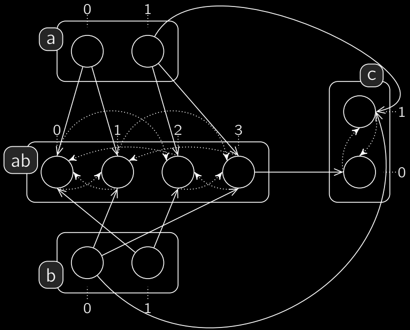
From BRNs to Process Hittings







$c = \neg a \wedge b$

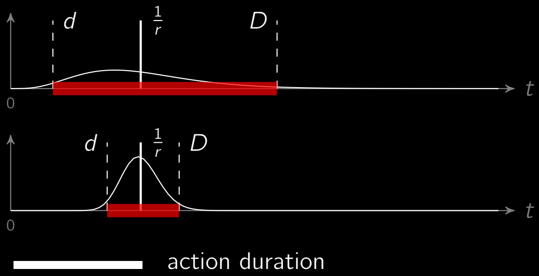


[Paulevé, Magnin, Roux in Trans. in Computational Systems Biology, 2010]

Stochastic and Temporal Parameters

- Duration of an action follows a random variable;
- Parameters: mean ($1/r$) and stochasticity absorption factor.
- Generalisation to the stochastic π-calculus, model-checking, parameters estimator (Erlang) [Paulevé, Magnin, Roux in IEEE TSE, 2010].
- Generic simulation of stochastic process calculi [Paulevé, Youssef, Lakin, Phillips at CMSB 2010].

Firing interval (confidence coefficient $1-\alpha$):



Scalable Dynamics Analysis

Objectives

- Analyse dynamics of large BRNs.
- Don't get lost in the state space.
- SotA: 20-30 components (we address easily 100 components).

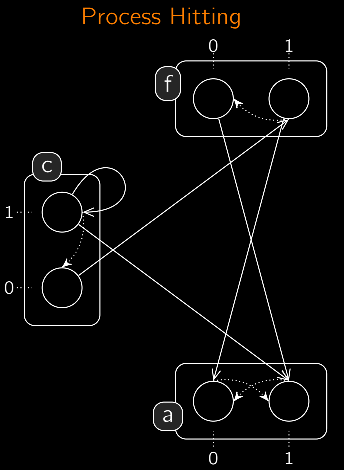
Methods

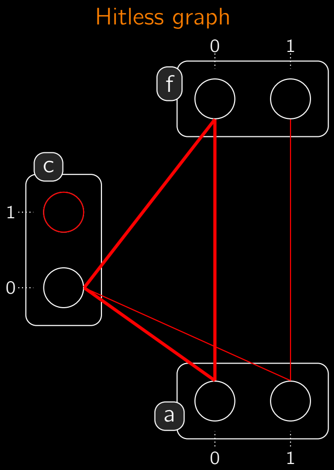
- Take advantage of Process Hitting simple structures;
- Static analysis + Abstract interpretation.
- Only look at the source code, no (explicit/symbolic) state space construction.

Stay tuned

- A first simple static analysis: stable states (fix points).
- Process reachability.

Stable states

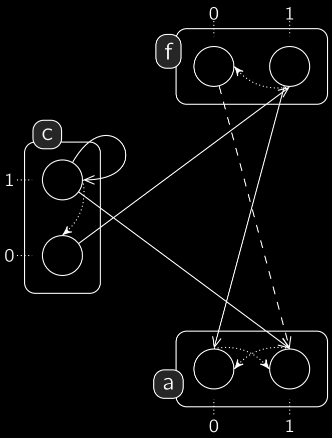




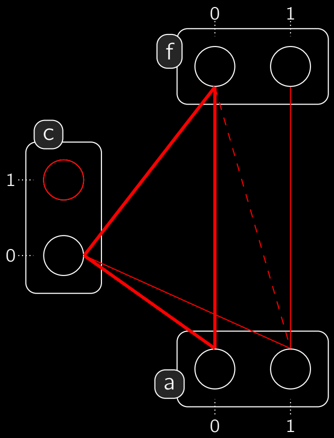
n-cliques are stable states

Stable states

Process Hitting



Hitless graph

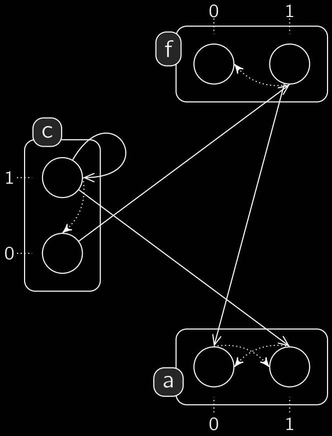


n-cliques are stable states

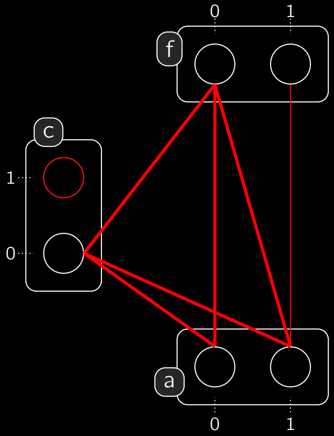
Loïc Paulevé, Morgan Magnin, Olivier Roux

Stable states

Process Hitting



Hitless graph



n-cliques are stable states

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1 , a_1 \rightarrow b_1 \upharpoonright b_0 , b_0 \rightarrow d_0 \upharpoonright d_1 , c_0 \rightarrow d_1 \upharpoonright d_2$$

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1,\ a_1 \rightarrow b_1 \upharpoonright b_0,\ b_0 \rightarrow d_0 \upharpoonright d_1,\ c_0 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by objective sequences

- $a_0 \upharpoonright^* a_1,\ b_1 \upharpoonright^* b_0,\ d_0 \upharpoonright^* d_1,\ d_1 \upharpoonright^* d_2$

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1 \,,\, a_1 \rightarrow b_1 \upharpoonright b_0 \,,\, b_0 \rightarrow d_0 \upharpoonright d_1 \,,\, c_0 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by objective sequences

- $a_0 \upharpoonright^* a_1 \,,\, b_1 \upharpoonright^* b_0 \,,\, d_0 \upharpoonright^* d_1 \,,\, d_1 \upharpoonright^* d_2$
- $b_1 \upharpoonright^* b_0 \,,\, d_0 \upharpoonright^* d_2$

Loïc Paulevé, Morgan Magnin, Olivier Roux

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1 \,,\, a_1 \rightarrow b_1 \upharpoonright b_0 \,,\, b_0 \rightarrow d_0 \upharpoonright d_1 \,,\, c_0 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by objective sequences

- $a_0 \upharpoonright^* a_1 \,,\, b_1 \upharpoonright^* b_0 \,,\, d_0 \upharpoonright^* d_1 \,,\, d_1 \upharpoonright^* d_2$
- $b_1 \upharpoonright^* b_0 \,,\, d_0 \upharpoonright^* d_2$
- $d_0 \upharpoonright^* d_2$

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1 \, , \, a_1 \rightarrow b_1 \upharpoonright b_0 \, , \, b_0 \rightarrow d_0 \upharpoonright d_1 \, , \, c_0 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by objective sequences

- $a_0 \upharpoonright^* a_1 \, , \, b_1 \upharpoonright^* b_0 \, , \, d_0 \upharpoonright^* d_1 \, , \, d_1 \upharpoonright^* d_2$
- $b_1 \upharpoonright^* b_0 \, , \, d_0 \upharpoonright^* d_2$
- $d_0 \upharpoonright^* d_2$
- ...

Abstraction by bounce sequences

- $a_1 \rightarrow b_1 \upharpoonright b_0 \; (b_1 \upharpoonright^* b_0)$

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1 \,,\, a_1 \rightarrow b_1 \upharpoonright b_0 \,,\, b_0 \rightarrow d_0 \upharpoonright d_1 \,,\, c_0 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by objective sequences

- $a_0 \upharpoonright^* a_1 \,,\, b_1 \upharpoonright^* b_0 \,,\, d_0 \upharpoonright^* d_1 \,,\, d_1 \upharpoonright^* d_2$
- $b_1 \upharpoonright^* b_0 \,,\, d_0 \upharpoonright^* d_2$
- $d_0 \upharpoonright^* d_2$
- ...

Abstraction by bounce sequences

- $a_1 \rightarrow b_1 \upharpoonright b_0 \; (b_1 \upharpoonright^* b_0)$
- $b_0 \rightarrow d_0 \upharpoonright d_1 \,,\, c_0 \rightarrow d_1 \upharpoonright d_2 \; (d_0 \upharpoonright^* d_2)$

The Process Reachability Problem

Scenario: sequence of playable actions.

$$b_1 \rightarrow a_0 \upharpoonright a_1 , a_1 \rightarrow b_1 \upharpoonright b_0 , b_0 \rightarrow d_0 \upharpoonright d_1 , c_0 \rightarrow d_1 \upharpoonright d_2$$

Abstraction by objective sequences

- $a_0 \upharpoonright^* a_1 , b_1 \upharpoonright^* b_0 , d_0 \upharpoonright^* d_1 , d_1 \upharpoonright^* d_2$
- $b_1 \upharpoonright^* b_0 , d_0 \upharpoonright^* d_2$
- $d_0 \upharpoonright^* d_2$
- ...

Abstraction by bounce sequences

- $a_1 \rightarrow b_1 \upharpoonright b_0 \ (b_1 \upharpoonright^* b_0)$
- $b_0 \rightarrow d_0 \upharpoonright d_1 , c_0 \rightarrow d_1 \upharpoonright d_2 \ (d_0 \upharpoonright^* d_2)$

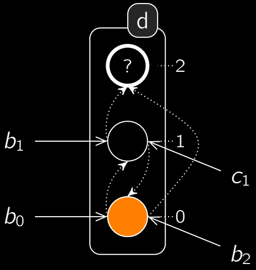
Process Reachability

Is an objective sequence $a_i \upharpoonright^* a_j , \dots , z_k \upharpoonright^* z_l$ concretizable in a state s?

$$EF(s_a = a_j \wedge EF(\cdots \wedge EF(s_z = z_l) \dots))$$

[Paulevé, Magnin, Roux at SASB 2010 + MSCS in prep.]

(Abstracted) Bounce Sequences



Objectives

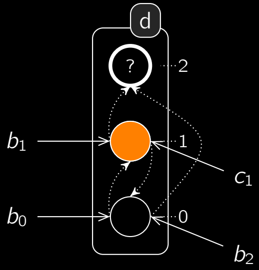
$d_0 \Rsh^* d_2,\ d_1 \Rsh^* d_2,\ \ldots$

Bounce Sequences

Only minimal sequences are kept.

$$\mathcal{BS}(d_0 \Rsh^* d_2) = \{[b_0 \rightarrow d_0 \Rsh d_1;\ b_1 \rightarrow d_1 \Rsh d_2],$$
$$[b_2 \rightarrow d_0 \Rsh d_2]\}$$
$$\mathcal{BS}(d_1 \Rsh^* d_2) = \{[b_1 \rightarrow d_1 \Rsh d_2],$$
$$[c_1 \rightarrow d_1 \Rsh d_0;\ b_2 \rightarrow d_0 \Rsh d_2]\}$$

(Abstracted) Bounce Sequences



Objectives

$d_0 \Rsh^* d_2$, $d_1 \Rsh^* d_2$, ...

Bounce Sequences

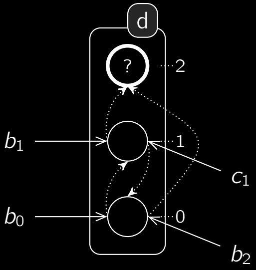
Only minimal sequences are kept.

$$\mathcal{BS}(d_0 \Rsh^* d_2) = \{[b_0 \rightarrow d_0 \Rsh d_1; b_1 \rightarrow d_1 \Rsh d_2], [b_2 \rightarrow d_0 \Rsh d_2]\}$$

$$\mathcal{BS}(d_1 \Rsh^* d_2) = \{[b_1 \rightarrow d_1 \Rsh d_2], [c_1 \rightarrow d_1 \Rsh d_0; b_2 \rightarrow d_0 \Rsh d_2]\}$$

Loïc Paulevé, Morgan Magnin, Olivier Roux

(Abstracted) Bounce Sequences



Objectives

$d_0 \Rsh^* d_2,\ d_1 \Rsh^* d_2, \ldots$

Bounce Sequences

Only minimal sequences are kept.

$$\mathcal{BS}(d_0 \Rsh^* d_2) = \{[b_0 \rightarrow d_0 \Rsh d_1; b_1 \rightarrow d_1 \Rsh d_2], [b_2 \rightarrow d_0 \Rsh d_2]\}$$

$$\mathcal{BS}(d_1 \Rsh^* d_2) = \{[b_1 \rightarrow d_1 \Rsh d_2], [c_1 \rightarrow d_1 \Rsh d_0; b_2 \rightarrow d_0 \Rsh d_2]\}$$

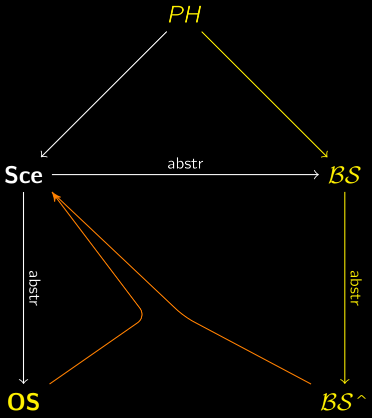
Abstracted Bounce Sequences

Only hitters are kept, and the order is forgotten.

$$\mathcal{BS}^\wedge(d_0 \Rsh^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$$

$$\mathcal{BS}^\wedge(d_1 \Rsh^* d_2) = \{\{b_1\}, \{b_2, c_1\}\}$$

Overall Approach



Idea: merge bounce sequences and objective sequences to form a scenario.

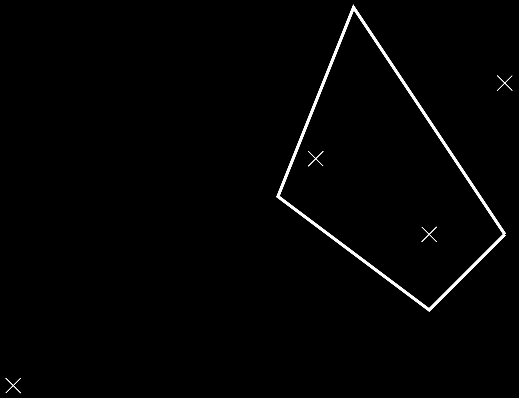
$\omega = a_0 \upharpoonright^* a_1, d_0 \upharpoonright^* d_2$

$\mathcal{BS}(a_0 \upharpoonright^* a_1) = \{[b_1 \to a_0 \upharpoonright a_1]\}$
$\mathcal{BS}(d_0 \upharpoonright^* d_2) = \{[a_1 \to d_0 \upharpoonright d_1; b_0 \to d_1 \upharpoonright d_2]\}$

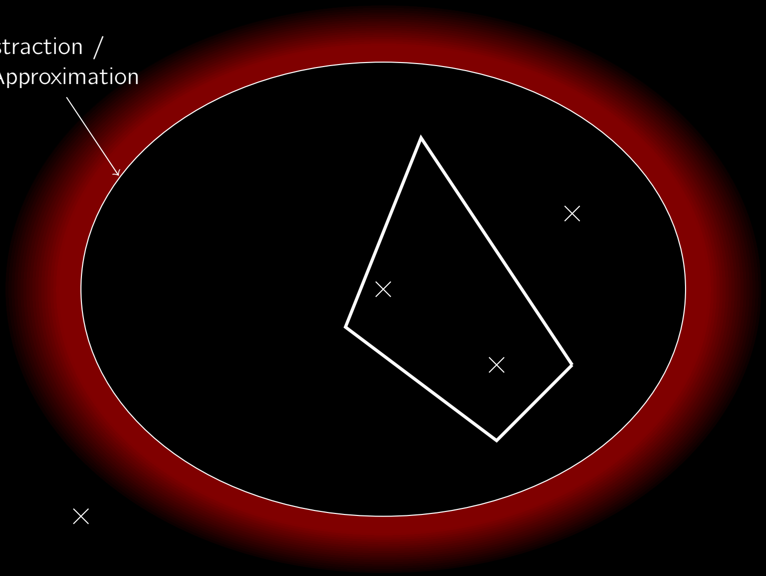
$\mathcal{BS}^\wedge(a_0 \upharpoonright^* a_1) = \{\{b_1\}\}$
$\mathcal{BS}^\wedge(d_0 \upharpoonright^* d_2) = \{\{a_1, b_0\}\}$

Use of an abstract structure relating dependencies between objectives and processes.

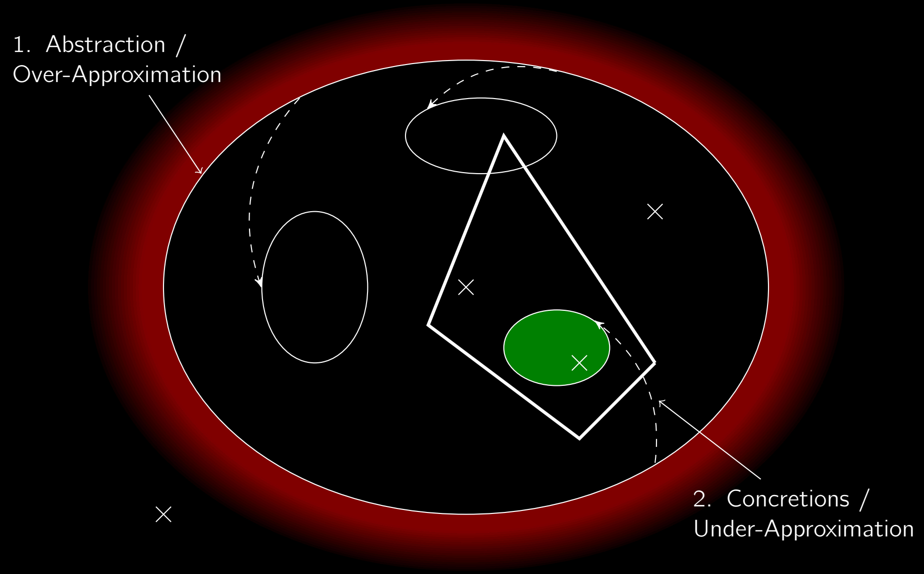
Method: Over- and Under-Approximation



Method: Over- and Under-Approximation

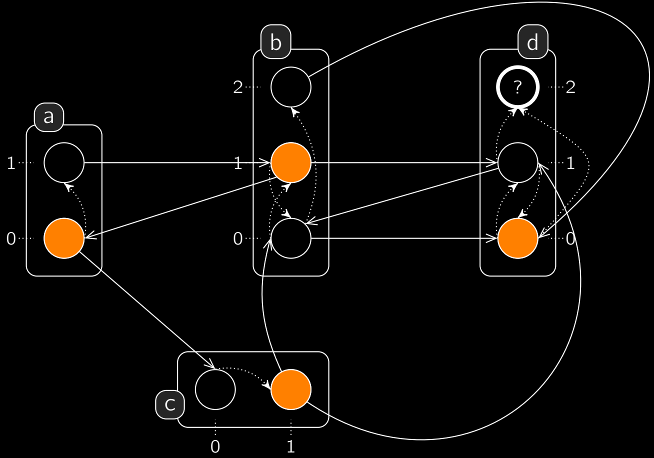


Method: Over- and Under-Approximation



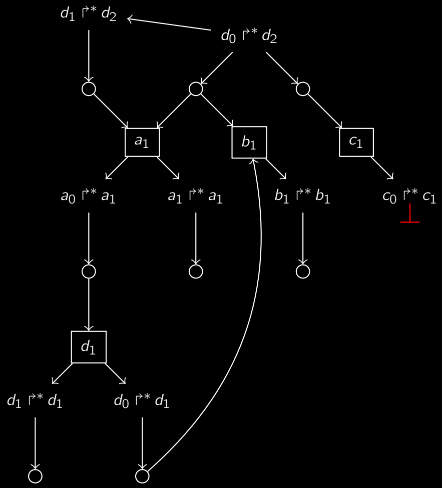
Loïc Paulevé, Morgan Magnin, Olivier Roux

Running Example

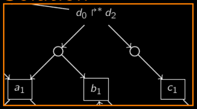


Is objective sequence $d_0 \upharpoonright^* d_2$ concretizable?

Process Hitting Abstract Structure

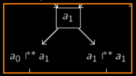


Solution



$\{\{a_1, b_1\}, \{c_1\}\} \subset \mathcal{BS}^\wedge(d_0 \Rsh^* d_2)$.

Requirement

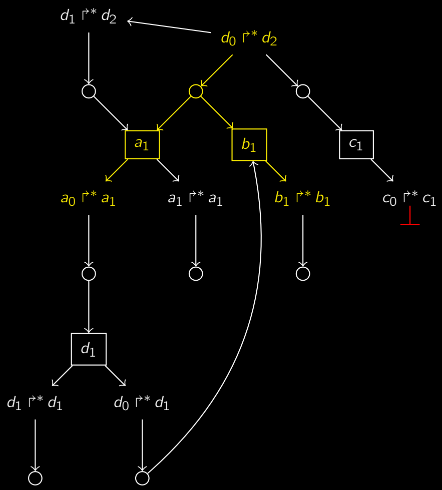


Objective to resolve (from current state).

Continuity

Objective resolution split.

Process Hitting Abstract Structure

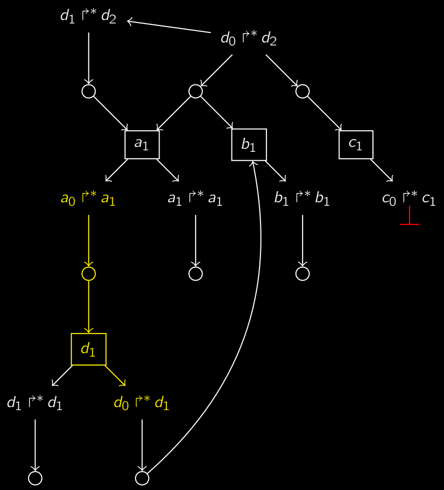


Starting with state $\langle a_0, b_1, c_0, d_0 \rangle$:

$$d_0 \Rsh^* d_2$$
$$\rightsquigarrow$$
$$\begin{cases} a_0 \Rsh^* a_1, b_1 \Rsh^* b_1, d_0 \Rsh^* d_2 \\ b_1 \Rsh^* b_1, a_0 \Rsh^* a_1, d_0 \Rsh^* d_2 \end{cases}$$

Loïc Paulevé, Morgan Magnin, Olivier Roux

Process Hitting Abstract Structure



Starting with state $\langle a_0, b_1, c_0, d_0 \rangle$:

$$d_0 \Rsh^* d_2$$

$$\rightsquigarrow$$

$$\begin{cases} a_0 \Rsh^* a_1, b_1 \Rsh^* b_1, d_0 \Rsh^* d_2 \\ b_1 \Rsh^* b_1, a_0 \Rsh^* a_1, d_0 \Rsh^* d_2 \end{cases}$$

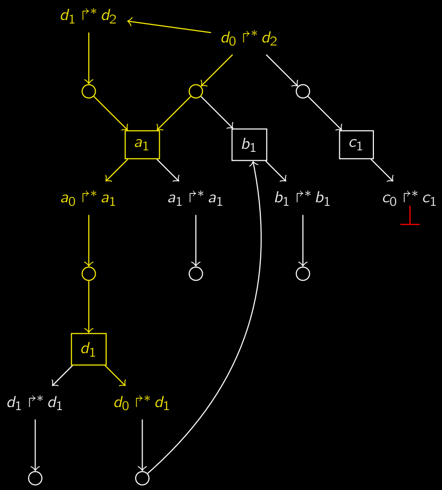
$$a_0 \Rsh^* a_1$$

$$\rightsquigarrow$$

$$d_0 \Rsh^* d_1, a_0 \Rsh^* a_1$$

Loïc Paulevé, Morgan Magnin, Olivier Roux

Process Hitting Abstract Structure



Starting with state $\langle a_0, b_1, c_0, d_0 \rangle$:

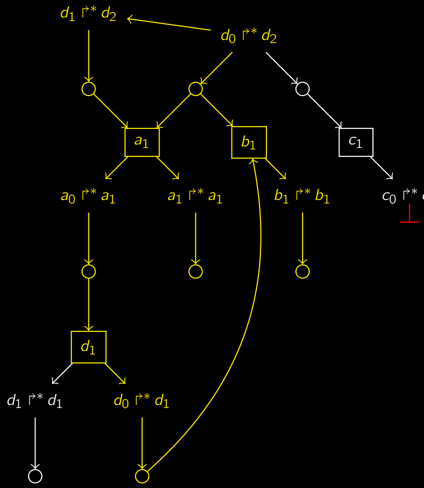
$$d_0 \Rsh^* d_2$$
$$\rightsquigarrow$$
$$\begin{cases} a_0 \Rsh^* a_1, b_1 \Rsh^* b_1, d_0 \Rsh^* d_2 \\ b_1 \Rsh^* b_1, a_0 \Rsh^* a_1, d_0 \Rsh^* d_2 \end{cases}$$

$$a_0 \Rsh^* a_1$$
$$\rightsquigarrow$$
$$d_0 \Rsh^* d_1, a_0 \Rsh^* a_1$$

$$a_0 \Rsh^* a_1, d_0 \Rsh^* d_2$$
$$\rightsquigarrow$$
$$d_0 \Rsh^* d_1, a_0 \Rsh^* a_1, d_1 \Rsh^* d_2$$

Loïc Paulevé, Morgan Magnin, Olivier Roux

Process Hitting Abstract Structure



Starting with state $\langle a_0, b_1, c_0, d_0 \rangle$:

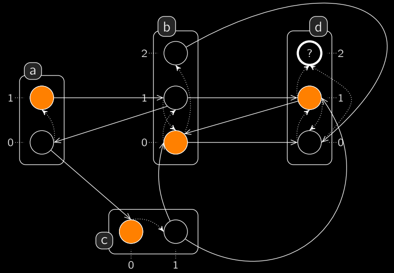
$$d_0 \Rsh^* d_2$$
$$\rightsquigarrow$$
$$\begin{cases} a_0 \Rsh^* a_1, b_1 \Rsh^* b_1, d_0 \Rsh^* d_2 \\ b_1 \Rsh^* b_1, a_0 \Rsh^* a_1, d_0 \Rsh^* d_2 \end{cases}$$

$$a_0 \Rsh^* a_1$$
$$\rightsquigarrow$$
$$d_0 \Rsh^* d_1, a_0 \Rsh^* a_1$$

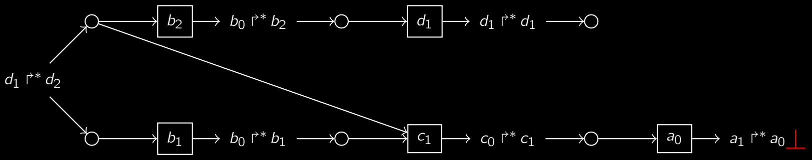
$$a_0 \Rsh^* a_1, d_0 \Rsh^* d_2$$
$$\rightsquigarrow$$
$$d_0 \Rsh^* d_1, a_0 \Rsh^* a_1, d_1 \Rsh^* d_2$$

$$d_0 \Rsh^* d_2$$
$$\rightsquigarrow$$
$$b_1 \Rsh^* b_1, d_0 \Rsh^* d_1, a_0 \Rsh^* a_1, d_1 \Rsh^* d_2$$

Over-approximation of Process Reachability

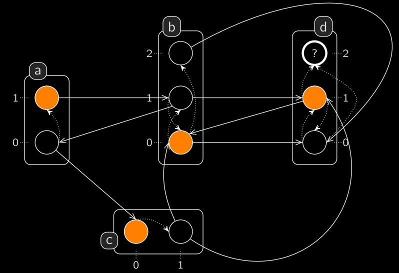


- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailled);
- Necessary condition: there always exists a solution ending with a trivial objective.

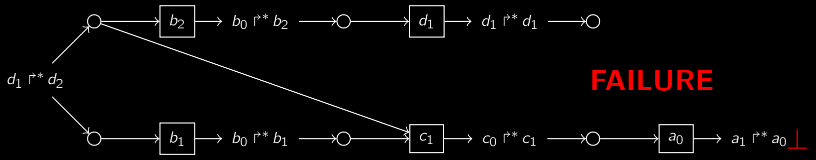


Loïc Paulevé, Morgan Magnin, Olivier Roux

Over-approximation of Process Reachability

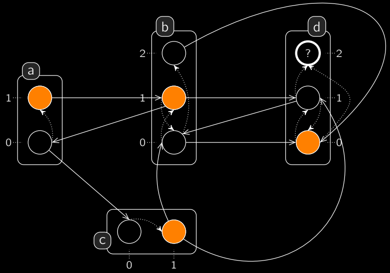


- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailled);
- Necessary condition: there always exists a solution ending with a trivial objective.

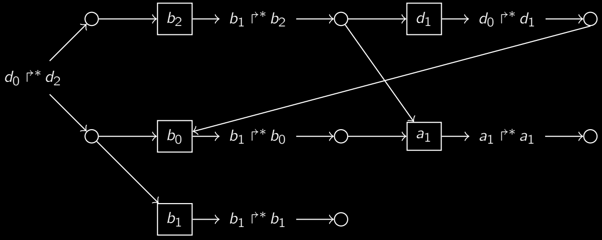


Loïc Paulevé, Morgan Magnin, Olivier Roux

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

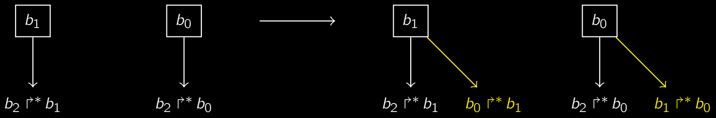
Under-Approximation of Process Reachability

Main idea: whatever the order of resolution, there is always a solution.

Conditions

- All objectives have at least one solution;
- No resolution cycles;
- Requirements saturation;
- Continuity saturation (not detailled).

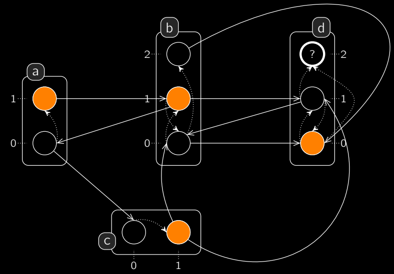
Requirements saturation

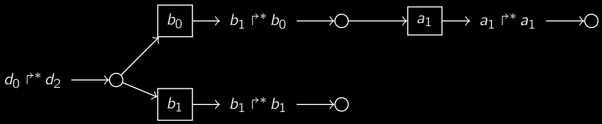


Remark

To increase conclusivness, one can arbitrarily select objective solutions.

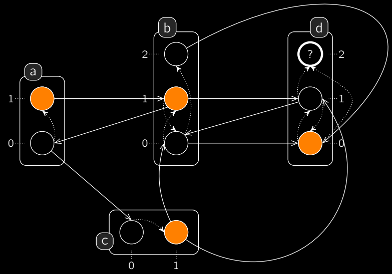
Under-approximation of Process Reachability

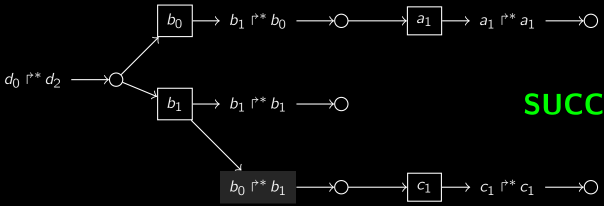




Loïc Paulevé, Morgan Magnin, Olivier Roux

Under-approximation of Process Reachability

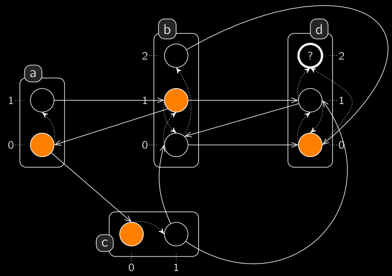


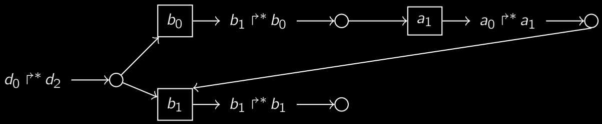


SUCCESS

Loïc Paulevé, Morgan Magnin, Olivier Roux

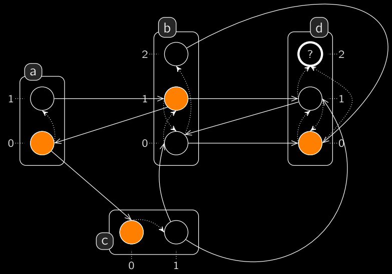
Under-approximation of Process Reachability

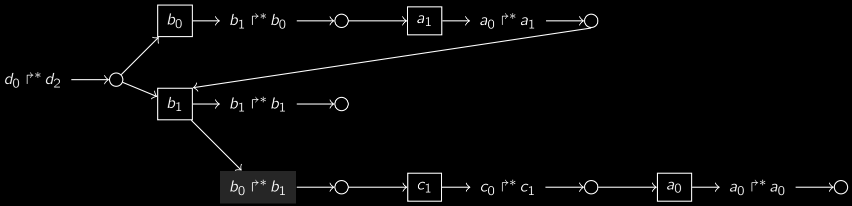




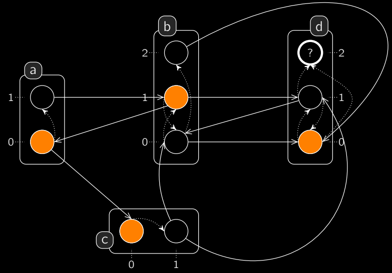
Loïc Paulevé, Morgan Magnin, Olivier Roux

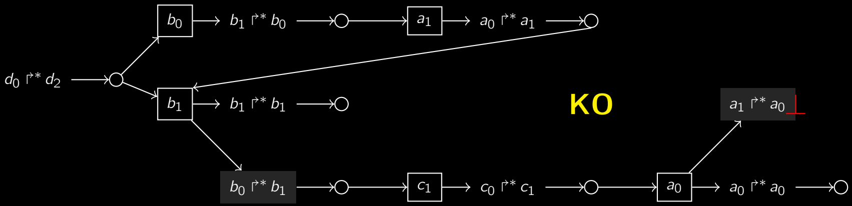
Under-approximation of Process Reachability





Under-approximation of Process Reachability





Pre-Conclusion

Comments

- Quite simple approximations from Process Hittings;
- prevent explicit state space exploration;
- abstract stucture provide information on required steps for reachability.

Complexities

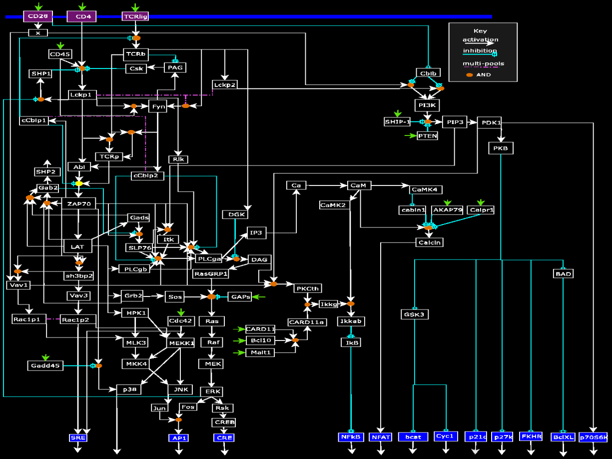
- Computing $\mathcal{BS}^\wedge$ is exponential in the size of the sort;
- other operations are $\approx$ polynomial in the number of sorts.

Take order into account (not detailled)

- $\mathcal{BS}^\wedge(a_1 \Rsh^* a_0) = \emptyset \Rightarrow$ a_0 cannot appear after a_1;
- refine over- and under-approximations with this knowledge:
- reduce inconclusivness.

T-Cell Receptor Signalling Pathway

(94 components)



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

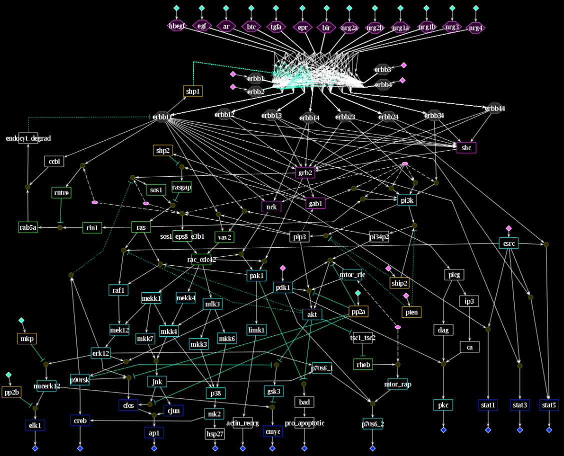
Process Hitting

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 0.05s (compared to *libddd*: out of memory). [http://ddd.lip6.fr]

Loïc Paulevé, Morgan Magnin, Olivier Roux

Conclusion and Outlook

Conclusion

- Efficient method to (semi-)decide process reachability;
- exploit particular structures of the Process Hitting;
- static analysis and abstract interpretation from the model;
- promising scalability for BRNs analysis.

Bleeding-edge results

- Reduce inconclusivness: exploit partial order of process appearance.
- Extract necessary processes for achieving reachabilities: towards control.
- Software (Pint) at `http://processhitting.wordpress.com`.

Future work

- How does the analysis apply to less constrained frameworks?
- Quantitative analysis (eg: probability of reaching a process within a time interval).