Abstract Modelling and Analysis of Large Biological Regulatory Networks

ETH Zurich - BISON Seminar - 4th April 2012

Loïc Paulevé

LIX, École Polytechnique, France
pauleve@lix.polytechnique.fr
http://loicpauleve.name

Joint work with Morgan Magnin and Olivier Roux
IRCCyN, École Centrale de Nantes, France (MeForBio team)

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.

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]

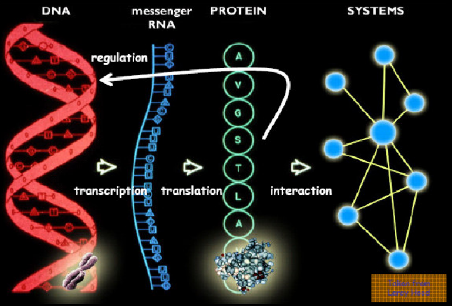
- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Stochastic and Time dimensions (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

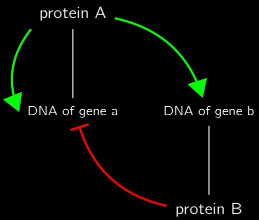
Large-scale model checking (discrete dynamical properties)

- Cope with state space explosion.
- Static Analysis by Abstract Interpretation
- Main result: efficient reachability properties approximation + clues for control.

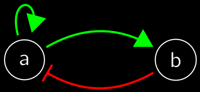
Biological Regulatory Networks (BRNs)

The interaction graph





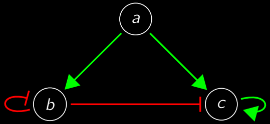
Interaction graph



Qualitative Networks

- Each component has a finite set of qualitative levels ($\{0, 1, 2\}$).
- Functions associate the next level given the state of the regulators.

Boolean example:

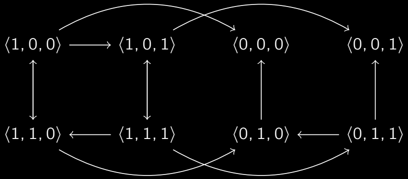


+K

$f^a(x) = 0$

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

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



[René Thomas in Journal of Theoretical Biology, 1973]
[Richard, Comet, Bernot in Modern Formal Methods and App., 2006]

Hybrid Modelling

Continuous features governing discrete transitions

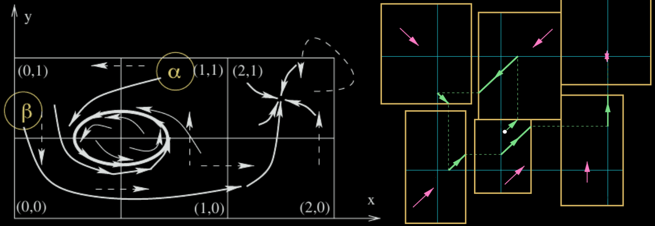
Introduce delays to actions

Stochastic Models

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

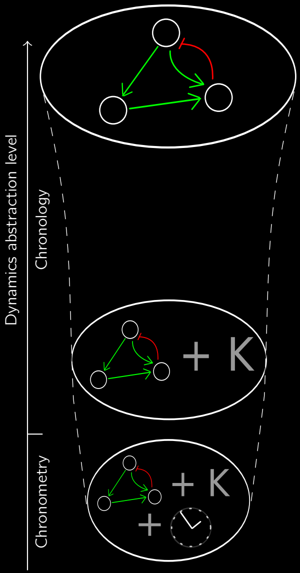
Stochastic Petri Nets / π-calculus, etc. [Heiner, Regev, Priami, Phillips, etc.]

Timed Models

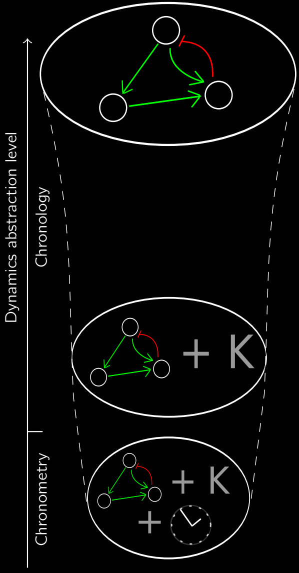


Timed / Hybrid Automata [Ahmad, Roux, Batt, Bockmayr, etc.]

State of the Art



State of the Art



General Properties: Bounds on attractors #; functionnal-ities; ...

Fixed points (steady states)

Successive level reachabilities

↑

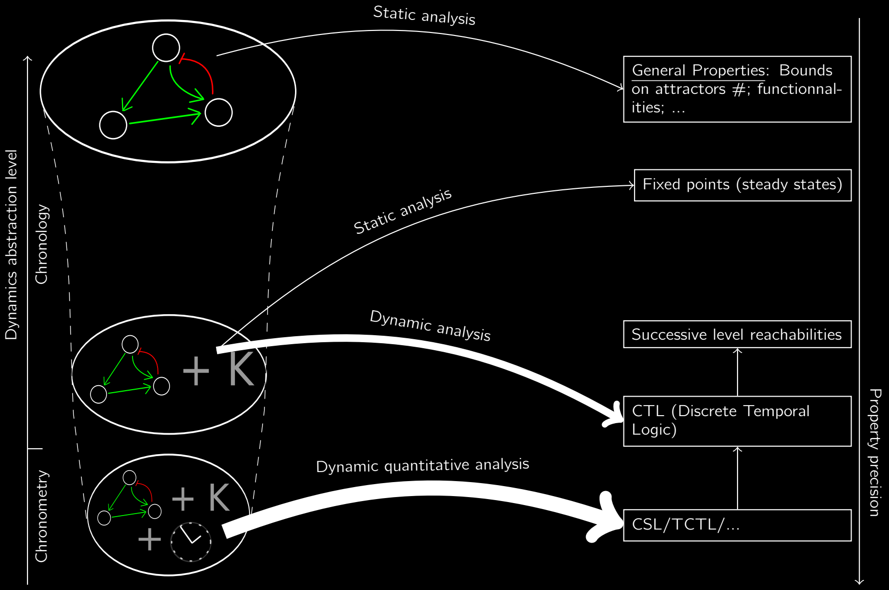
CTL (Discrete Temporal Logic)

↑

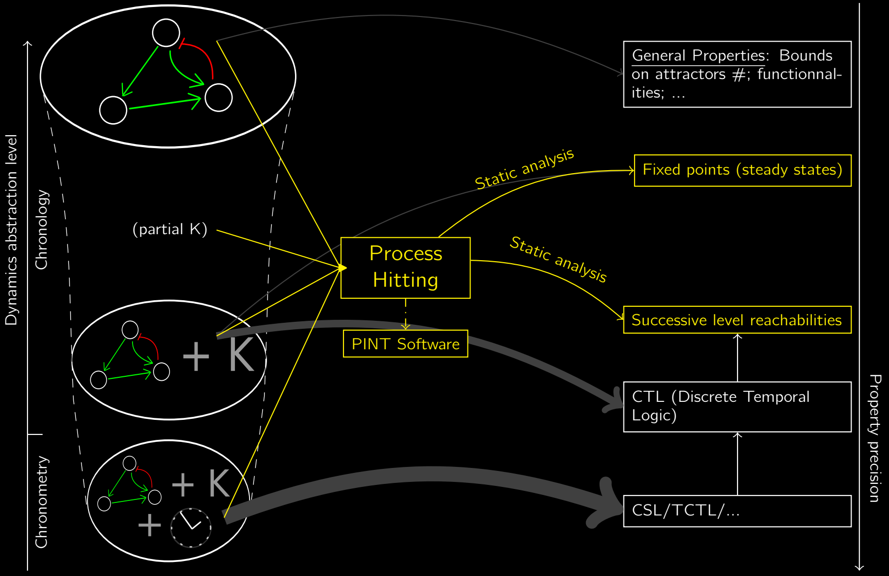
CSL/TCTL/...

Property precision

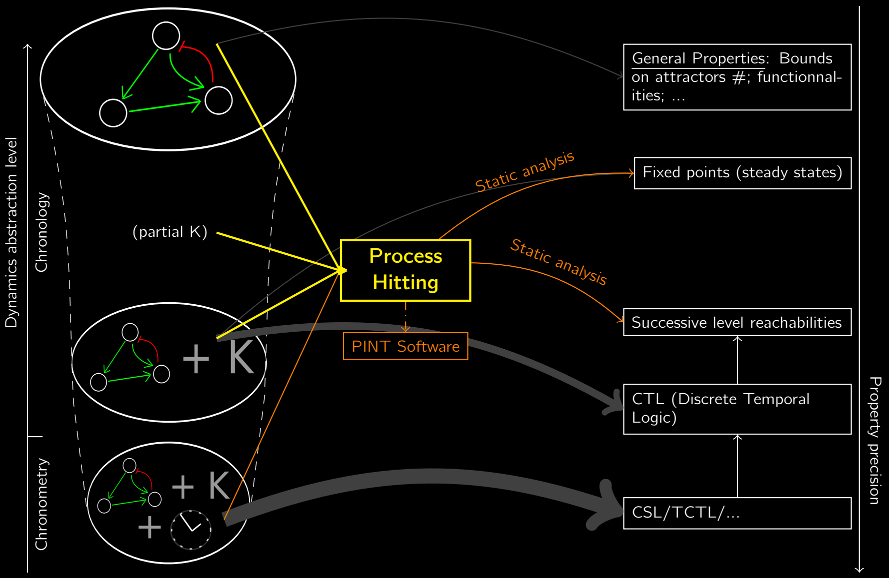
State of the Art



Contributions

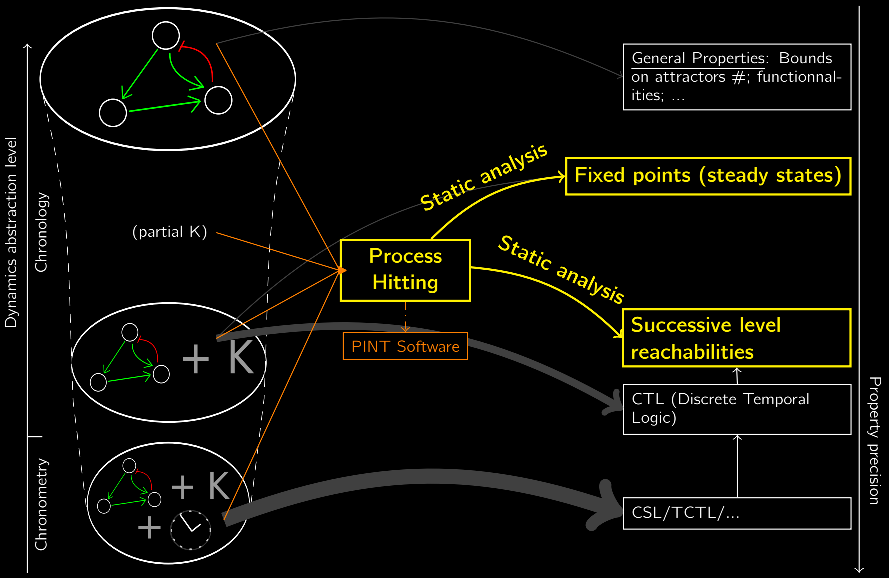


Outline



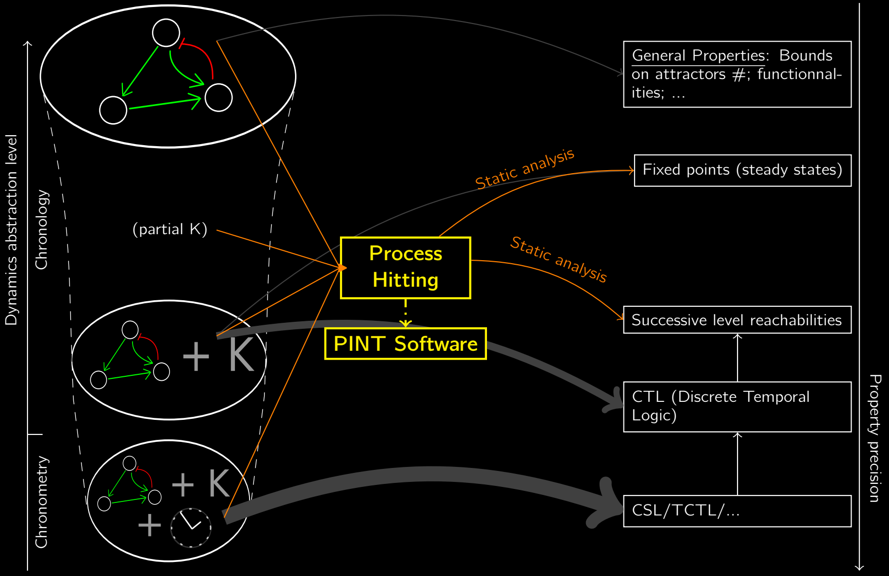
Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis
General Properties: Bounds on attractors #; functionnal-ities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

Outline

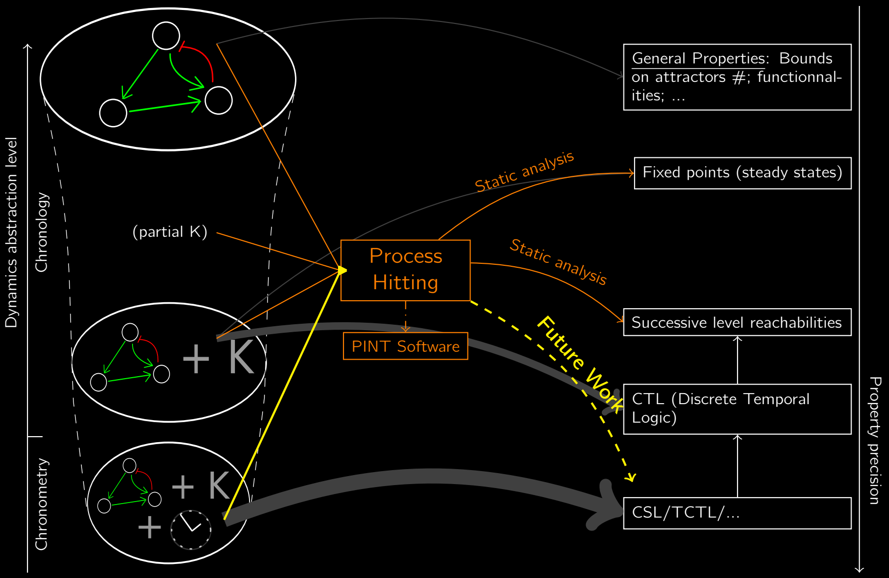


Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
General Properties: Bounds on attractors #; functionnal-ities; ...
Static analysis
Fixed points (steady states)
Process Hitting
Static analysis
PINT Software
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

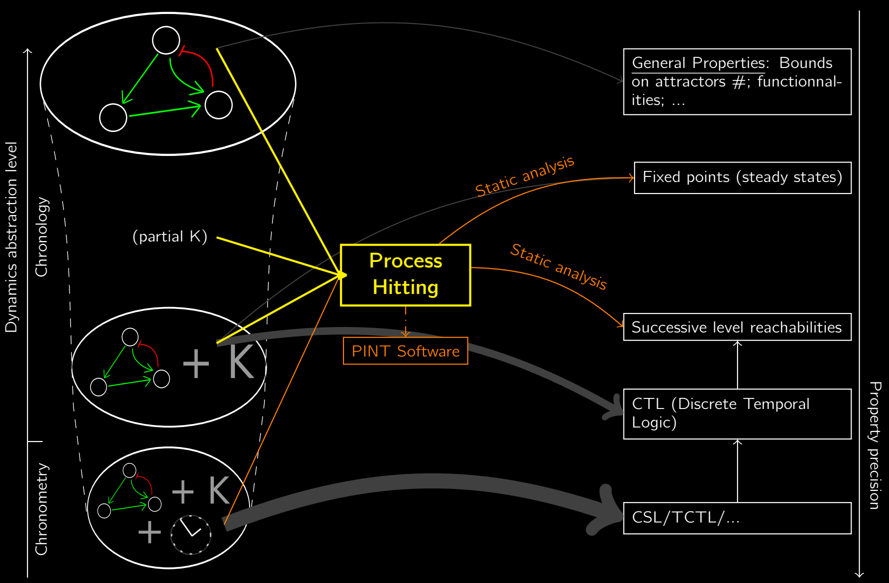
Outline


Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

Outline

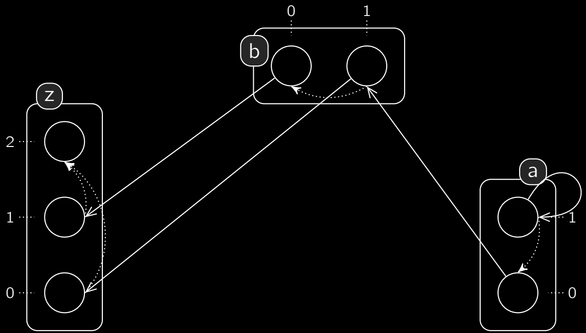

Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis
Future Work
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

Outline


Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

The Process Hitting Framework

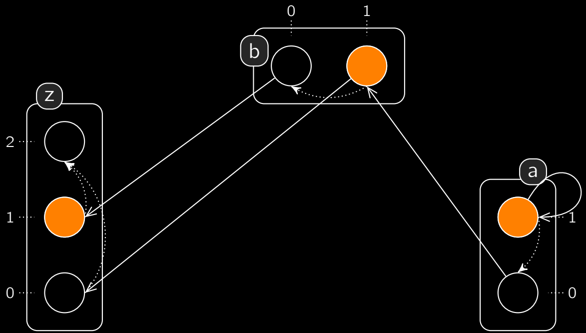
[Paulevé, Magnin, Roux in TCSB 2011]



- 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

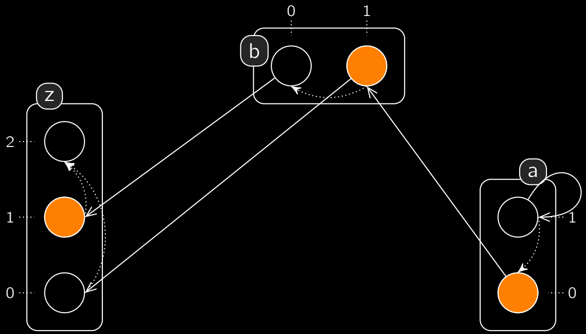
[Paulevé, Magnin, Roux in TCSB 2011]



- 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

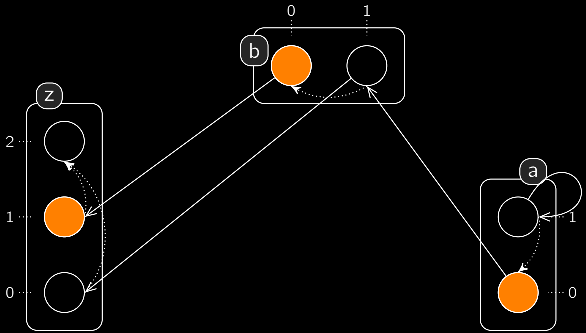
[Paulevé, Magnin, Roux in TCSB 2011]



- 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

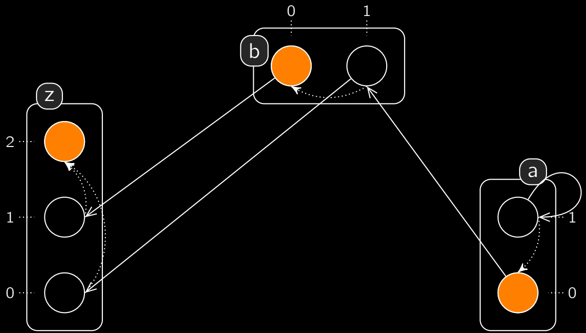
[Paulevé, Magnin, Roux in TCSB 2011]



- 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]

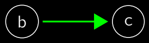


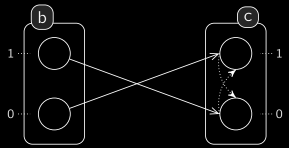
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines (CFSM).

Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

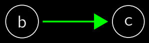


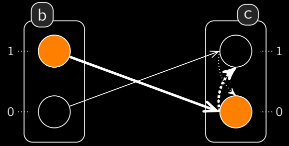


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

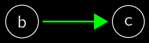


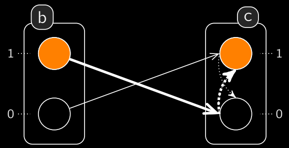


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

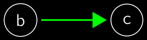


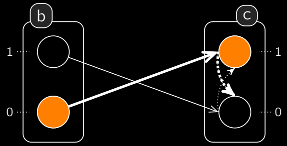


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

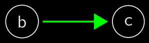


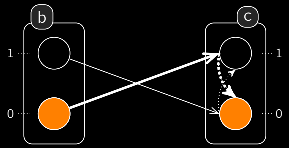


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

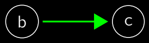


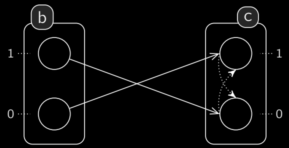


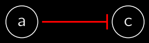
Generalized Dynamics of BRNs

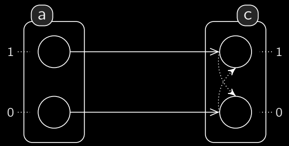
- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:



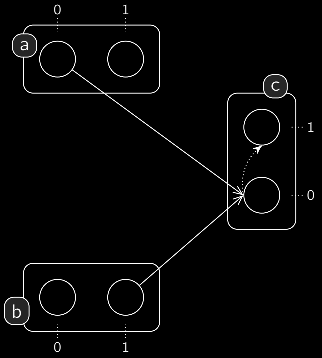






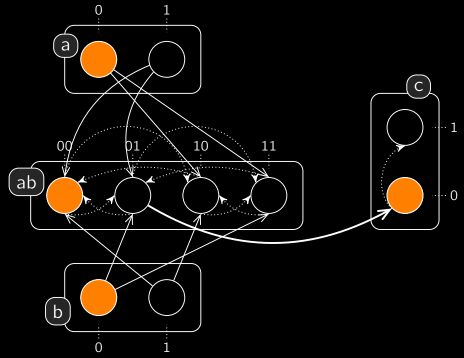
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative sort reflecting the state of the sorts a and b.



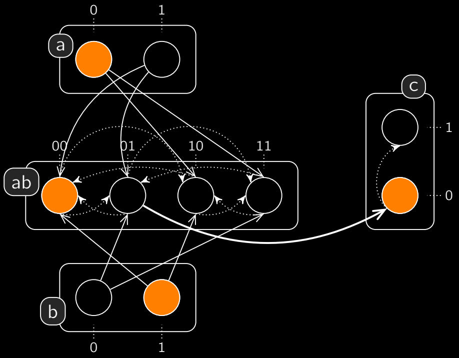
Refining with Cooperation

- Idea: $c_0 \upharpoonright c_1$ when a_0 <u>and</u> b_1 are present.
- Introduction of a cooperative sort reflecting the state of the sorts a and b.



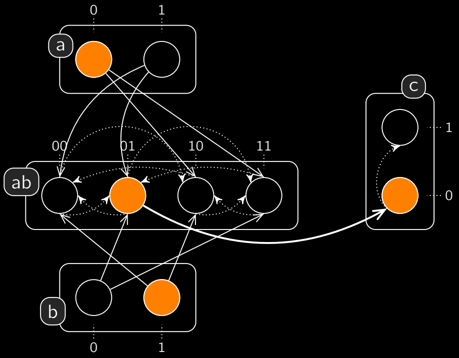
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative sort reflecting the state of the sorts a and b.



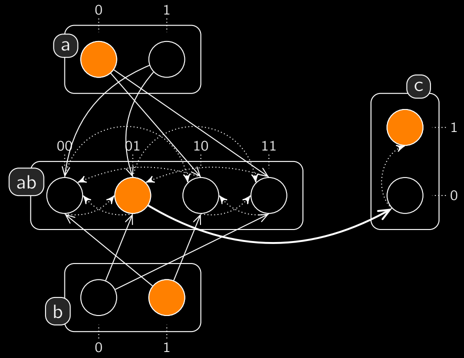
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative sort reflecting the state of the sorts a and b.



Refining with Cooperation

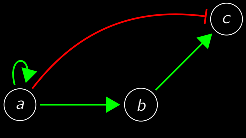
- Idea: $c_0 \upharpoonright c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative sort reflecting the state of the sorts a and b.

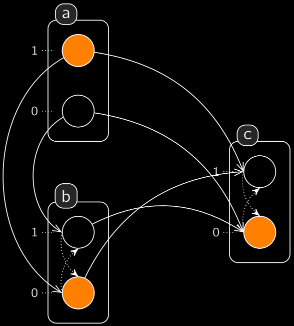


⇒ introduce a temporal shift; similar to complexes.

Toy example

Incoherent feed-forward loop





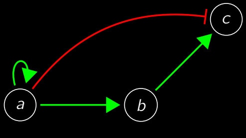
$\langle a_1, b_0, c_0 \rangle$

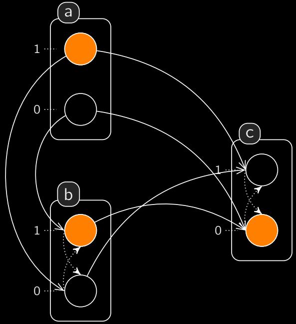
$\downarrow$

$\langle a_1, b_1, c_0 \rangle \longleftrightarrow \langle a_1, b_1, \mathbf{c_1} \rangle$

Toy example

Incoherent feed-forward loop





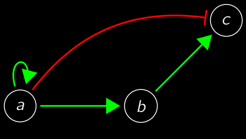
$\langle a_1, b_0, c_0 \rangle$

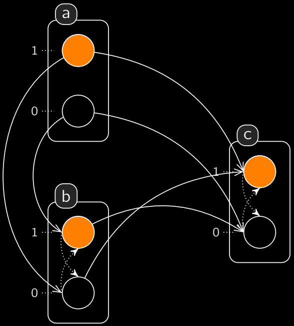
$\downarrow$

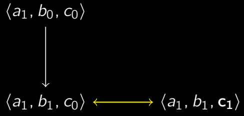
$\langle a_1, b_1, c_0 \rangle \longleftrightarrow \langle a_1, b_1, \mathbf{c_1} \rangle$

Toy example

Incoherent feed-forward loop

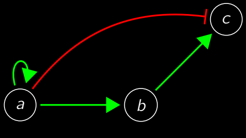


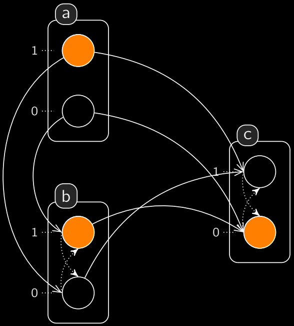




Toy example

Incoherent feed-forward loop





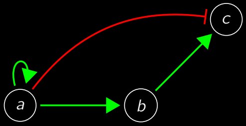
$\langle a_1, b_0, c_0 \rangle$

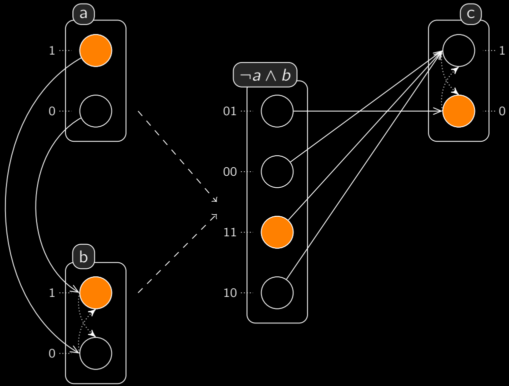
$\downarrow$

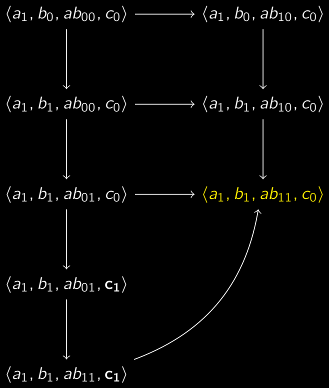
$\langle a_1, b_1, c_0 \rangle \longleftrightarrow \langle a_1, b_1, \mathbf{c_1} \rangle$

Toy example

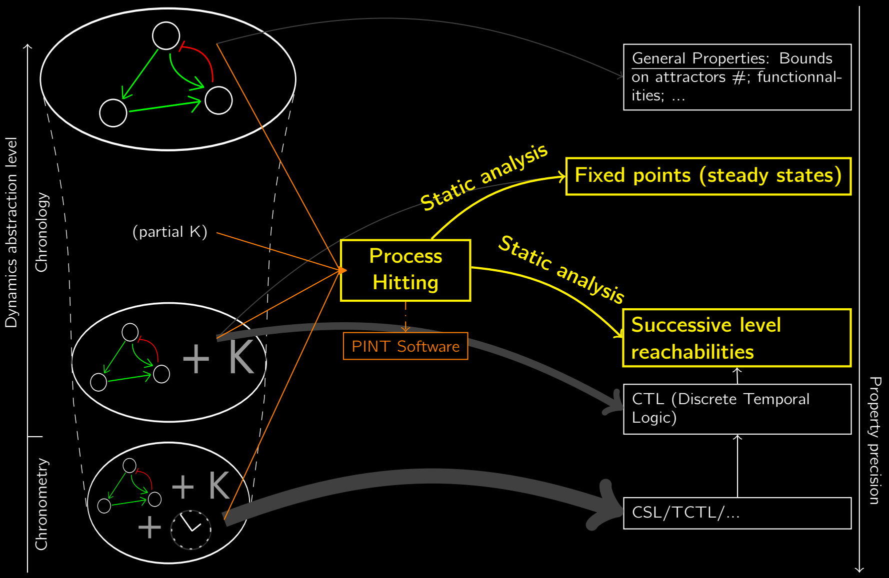
Incoherent feed-forward loop





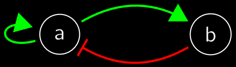


Outline


Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis
General Properties: Bounds on attractors #; functionnal-ities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

Static Analysis of BRNs using the Interaction Graph

An interaction graph can describe a large set of different dynamics.



Relationships 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 require 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].
- etc.

(See [Paulevé, Richard at SASB'11] for a short survey).

Static Analysis of Process Hittings

Intuition

- Simplicity of the Process Hitting $\Rightarrow$ models with simple structures.
- Efficient static derivation of dynamical properties.

Static Analysis of Process Hittings

Intuition

- Simplicity of the Process Hitting $\Rightarrow$ models with simple structures.
- Efficient static derivation of dynamical properties.

Fixed Points

- Complete enumeration of fixed points.
- Reduction to the n-cliques of an n-partite graph.

Static Analysis of Process Hittings

Intuition

- Simplicity of the Process Hitting ⇒ models with simple structures.
- Efficient static derivation of dynamical properties.

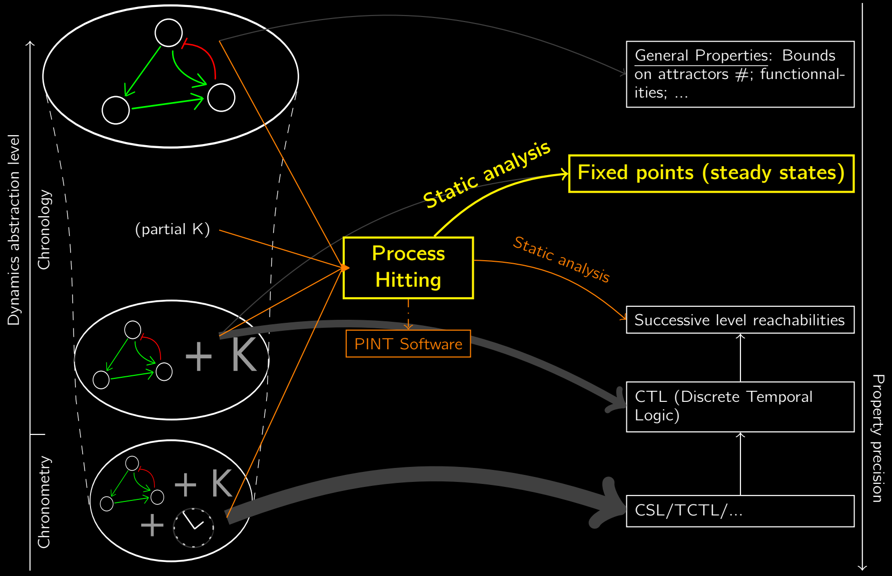
Fixed Points

- Complete enumeration of fixed points.
- Reduction to the n-cliques of an n-partite graph.

Successive reachability properties $\mathrm{EF}\ a_i \wedge (\mathrm{EF}\ b_j \wedge \dots)$

- Limited complexity but may be inconclusive (Yes/No/Inconc).
- Abstract interpretation techniques.
- Extraction of key processes (towards control).

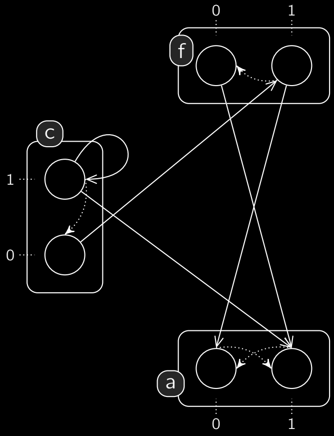
Outline


Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision

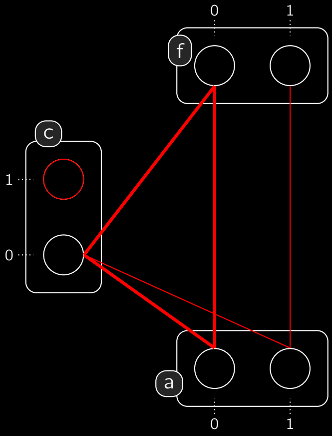
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



Hitless graph

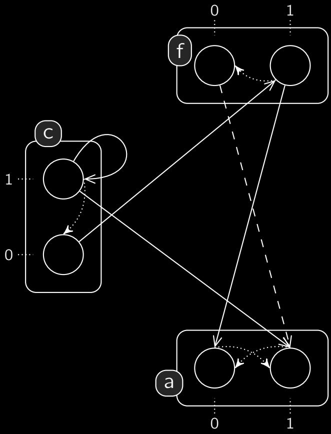


n-cliques <u>are</u> fixed points

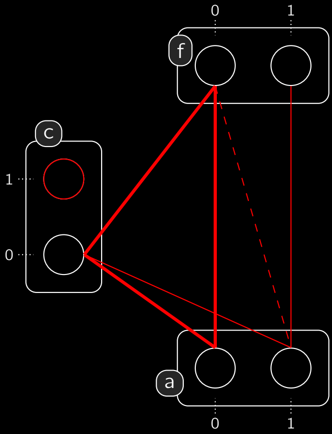
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



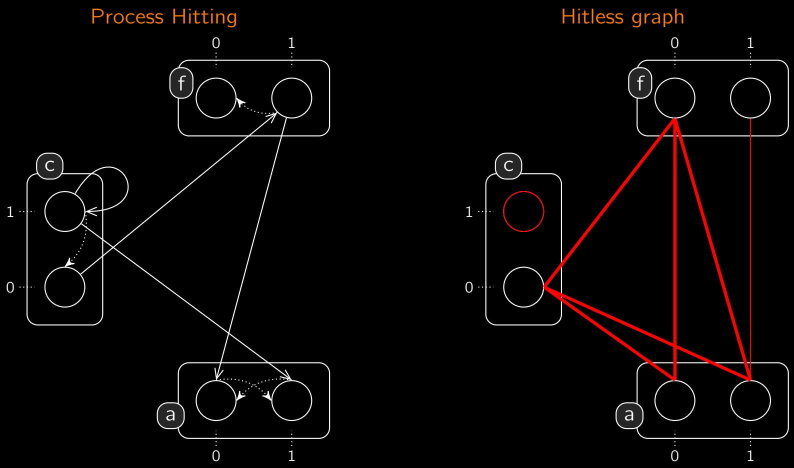
Hitless graph



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

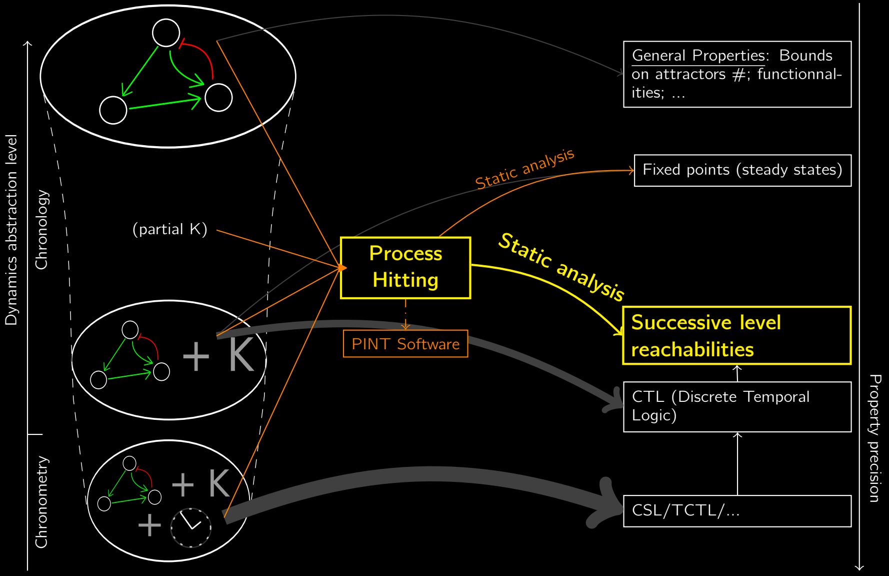
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]



n-cliques are fixed points

Outline


General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Static analysis
(partial K)
Process Hitting
Static analysis
Successive level reachabilities
PINT Software
+ K
CTL (Discrete Temporal Logic)
+ K
+
CSL/TCTL/...
Dynamics abstraction level
Chronology
Chronometry
Property precision

Static Analysis of Successive Reachability Properties

[Paulevé, Magnin, Roux in MSCS 2012]

Successive Reachability $\mathcal{R}$

- Given a Process Hitting $\mathcal{PH}$ with an initial state,
- is it possible to reach the process a_i? ...
- then the process b_j? ... etc.

Static Analysis of Successive Reachability Properties

[Paulevé, Magnin, Roux in MSCS 2012]

Successive Reachability $\mathcal{R}$

- Given a Process Hitting $\mathcal{PH}$ with an initial state,
- is it possible to reach the process a_i? ...
- then the process b_j? ... etc.

Difficulties: combinatorial explosion of dynamics to explore.

Static Analysis of Successive Reachability Properties

[Paulevé, Magnin, Roux in MSCS 2012]

Successive Reachability $\mathcal{R}$

- Given a Process Hitting $\mathcal{PH}$ with an initial state,
- is it possible to reach the process a_i? ...
- then the process b_j? ...etc.

Difficulties: combinatorial explosion of dynamics to explore.

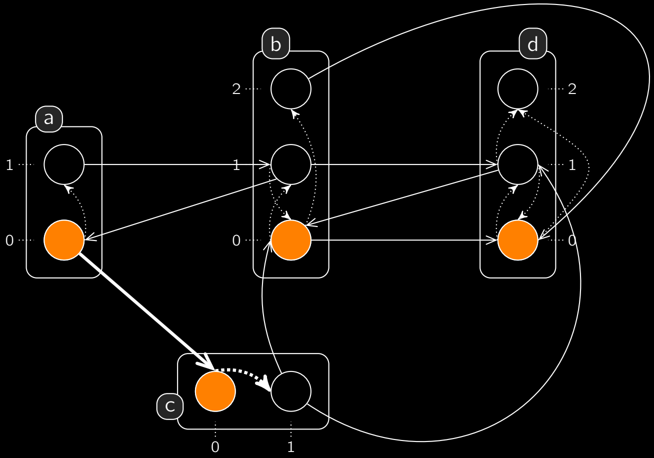
Chosen approach

Over-approximations
$\mathcal{PH}$ does not satisfy $\mathcal{P} \implies \mathcal{R}$ is impossible.

Under-approximations
$\mathcal{PH}$ satisfies $\mathcal{Q} \implies \mathcal{R}$ is possible.

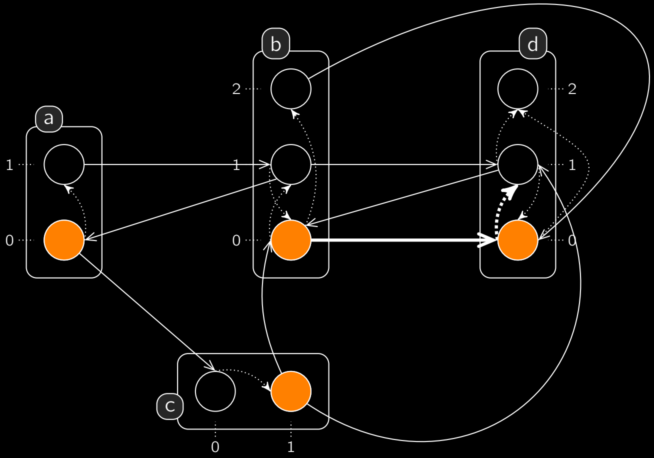
Requirement: checking $\mathcal{P}$ ($\mathcal{Q}$) is fast.

Scenarios



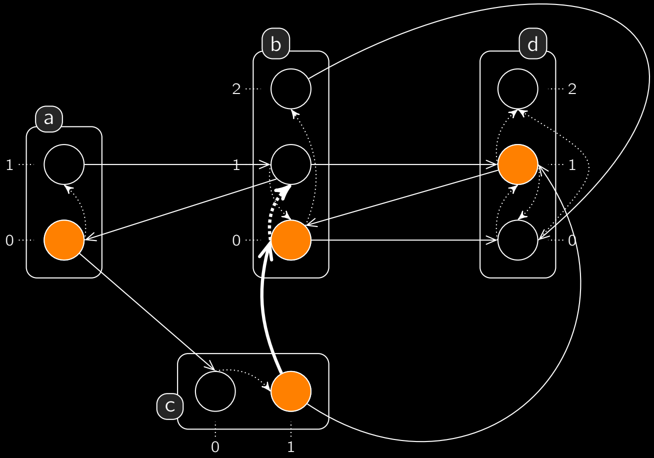
$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$

Scenarios



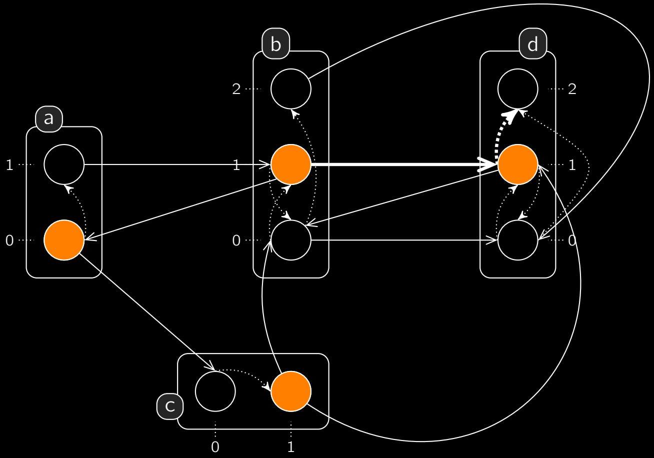
$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$

Scenarios



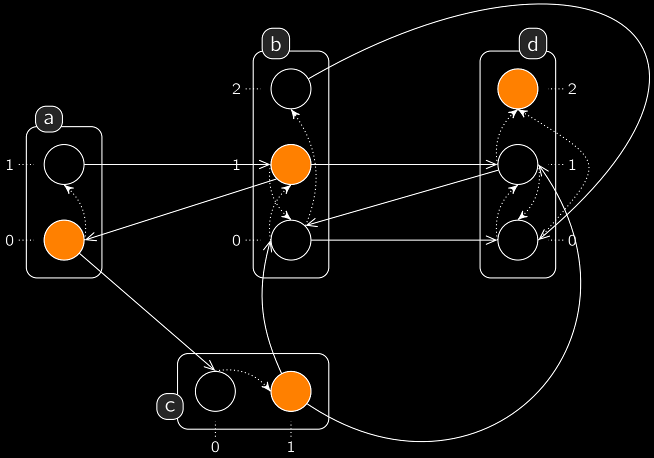
$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$

Scenarios



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

Scenarios



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

Two Complementary Abstractions of Scenarios

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

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 of Scenarios

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

Abstraction by Objective Sequences

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

Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

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

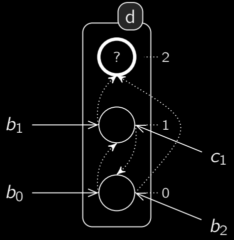
Two Complementary Abstractions of Scenarios

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

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.: $b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2$ $(d_0 \Rsh^* d_2)$

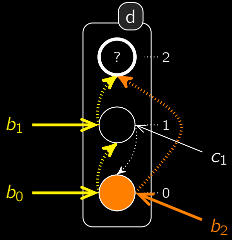
Two Complementary Abstractions of Scenarios

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

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:

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

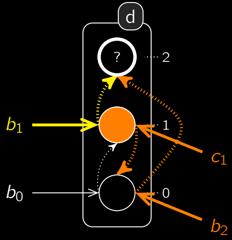
Two Complementary Abstractions of Scenarios

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

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:

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

Abstract Interpretation of Scenarios

Scenarios – Successively playable actions.

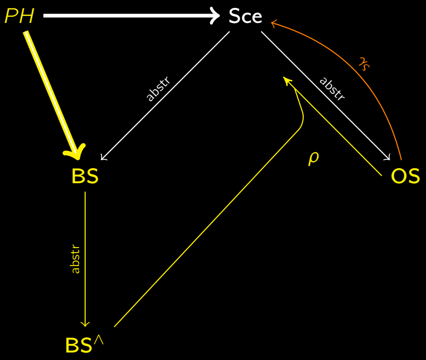
- E.g. $\delta = a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$.

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: $\rho : \mathbf{OS} \mapsto \wp(\mathbf{OS})$;
- $\gamma_\varsigma(\omega) = \gamma_\varsigma(\rho(\omega))$.



Objective Sequence Refinements

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

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

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

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

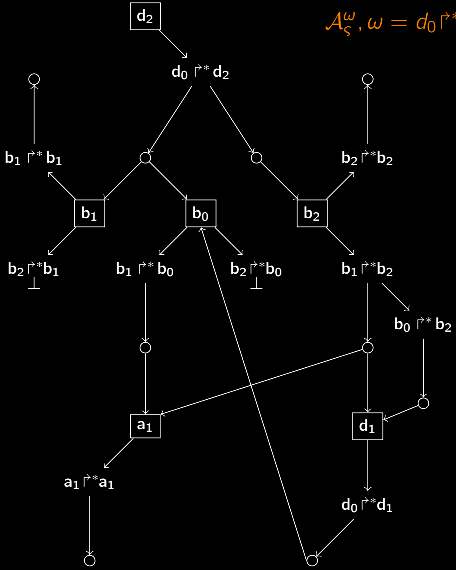
$\textbf{Obj} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$d_0 \Rsh^* d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \Rsh^* b_0 :: b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_1 :: b_1 \Rsh^* b_0 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_2 :: d_0 \Rsh^* d_2$
$\gamma_\varsigma(d_0 \Rsh^* d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0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

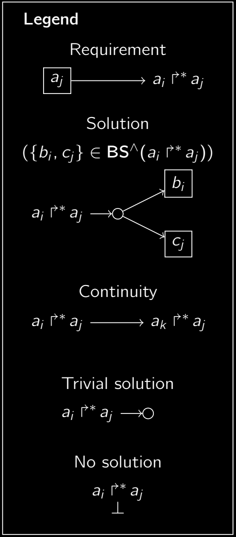
Generalization to **OS** refinements: $\widetilde{\rho}$

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

Abstract Structure of Process Hitting

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





Approximations of Successive Reachability

Over-approximations	• Un-ordered approximation. • Ordered approximation. • Ordered Approximation with occurences order constraints.	No / Inconc
	⇑ Successive Reachability ⇑	
Under-approximations	• Un-ordered approximation. • Ordered approximation.	Yes / Inconc

Approximations of Successive Reachability

Over-approximations

- Un-ordered approximation.
- Ordered approximation.
- Ordered Approximation with occurences order constraints.

No / Inconc

⇑

Successive Reachability

⇑

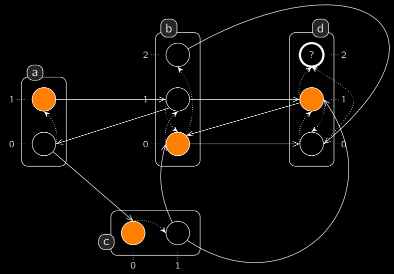
Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

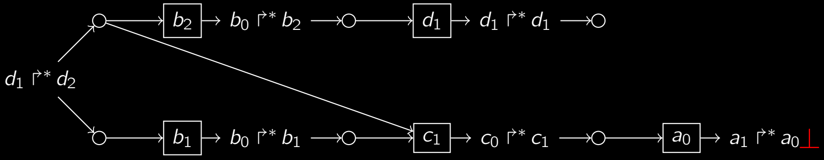
Un-ordered Over-approximation

Example



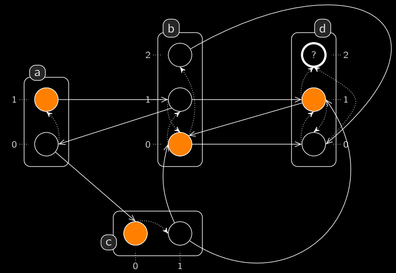
Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}^\omega_\varsigma$ such that:

- objective $\rightarrow$ follow at least one solution;
- process $\rightarrow$ follow all objectives;
- no cycle.



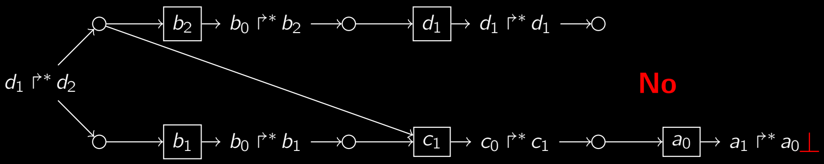
Un-ordered Over-approximation

Example



Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}^\omega_\varsigma$ such that:

- objective $\rightarrow$ follow at least one solution;
- process $\rightarrow$ follow all objectives;
- no cycle.

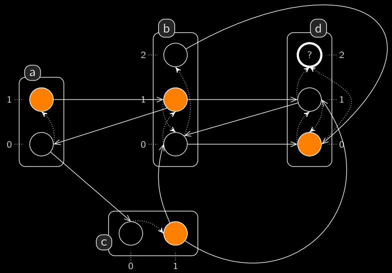


No

Loïc Paulevé

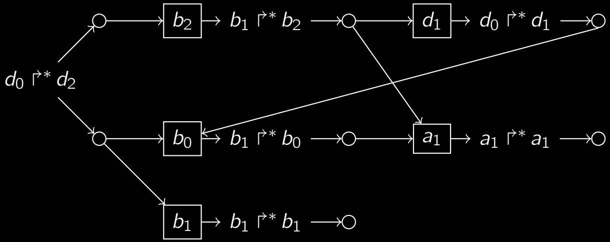
Un-ordered Over-approximation

Example



Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}_\varsigma^\omega$ such that:

- objective $\rightarrow$ follow at least one solution;
- process $\rightarrow$ follow all objectives;
- no cycle.



Inconc

Approximations of Successive Reachability

Over-approximations	• Un-ordered approximation. • Ordered approximation. • Ordered Approximation with occurences order constraints.	No / Inconc

⇑

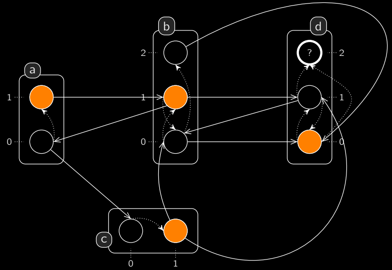
Successive Reachability

⇑

Under-approximations	• Un-ordered approximation. • Ordered approximation.	Yes / Inconc

Un-ordered Under-approximation

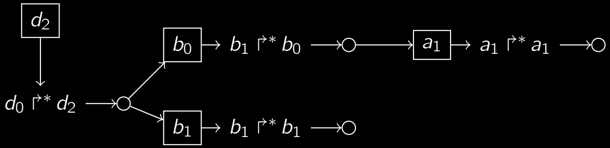
Example



Sufficient condition for $\gamma_\varsigma(\omega) \neq \emptyset$:

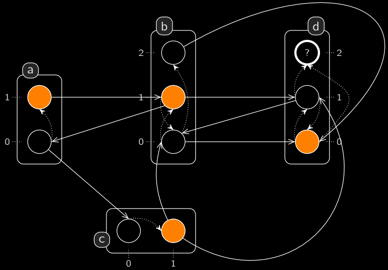
- $\lceil \mathcal{B}_\varsigma^\omega \rceil$ has no cycle;
- each objective has at least one solution.

$\lceil \mathcal{B}_\varsigma^\omega \rceil$: saturated $\mathcal{A}_\varsigma^\omega$.



Un-ordered Under-approximation

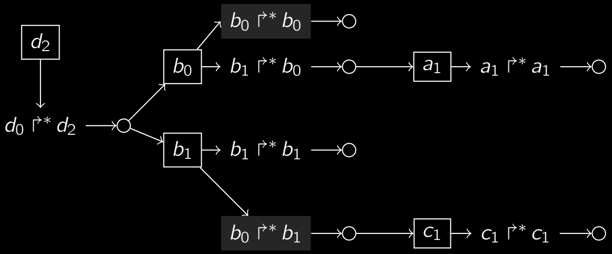
Example



Sufficient condition for $\gamma_\varsigma(\omega) \neq \emptyset$:

- $\lceil \mathcal{B}_\varsigma^\omega \rceil$ has no cycle;
- each objective has at least one solution.

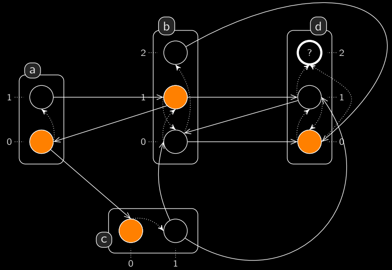
$\lceil \mathcal{B}_\varsigma^\omega \rceil$: saturated $\mathcal{A}_\varsigma^\omega$.



Yes

Un-ordered Under-approximation

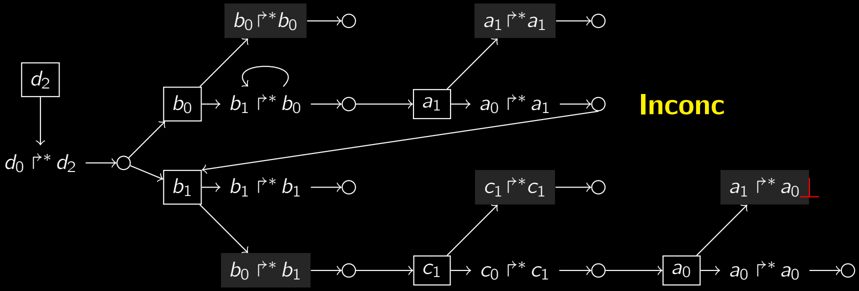
Example



Sufficient condition for $\gamma_\varsigma(\omega) \neq \emptyset$:

- $\lceil \mathcal{B}^\omega_\varsigma \rceil$ has no cycle;
- each objective has at least one solution.

$\lceil \mathcal{B}^\omega_\varsigma \rceil$: saturated $\mathcal{A}^\omega_\varsigma$.



Static Analysis of Successive Reachability

Over-approximations

- Un-ordered approximation.
- Ordered approximation.
- Ordered Approximation with occurences order constraints.

No / Inconc

⇑

Successive Reachability

⇑

Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

Loïc Paulevé

Static Analysis of Successive Reachability

Over-approximations	• Un-ordered approximation. • Ordered approximation. • Ordered Approximation with occurences order constraints.	No / Inconc
	⇑ Successive Reachability ⇑	
Under-approximations	• Un-ordered approximation. • Ordered approximation.	Yes / Inconc

Still inconclusive?

- Require new analyses of the abstract structure
- ⇒ drive refinements of ω.

Complexity

Abstract Strutures $\mathcal{A}_{\varsigma}^{\omega}$, $\lceil \mathcal{B}_{\varsigma}^{\omega} \rceil$

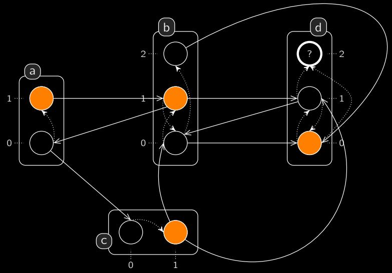
- $\mathbf{BS}^{\wedge}$ computation: exponential in the number of processes within a single sort.
- Size of $\mathbf{BS}^{\wedge}$: combinaisons of $|\mathbf{Proc}_a|$ processes $\binom{|\mathbf{Proc}|}{|\mathbf{Proc}_a|}$.
- Size of $\mathcal{A}_{\varsigma}^{\omega}$ (and $\lceil \mathcal{B}_{\varsigma}^{\omega} \rceil$): polynomial in processes number $\times$ size of $\mathbf{BS}^{\wedge}$.

Analyses

- Over-approximations: polynomial in the size of $\mathcal{A}_{\varsigma}^{\omega}$.
- Different strategies of under-approximation:
  - global: polynomial in the size of $\lceil \mathcal{B}_{\varsigma}^{\omega} \rceil$;
  - per solution: $\times$ exponential in the size of $\mathbf{BS}^{\wedge}$.

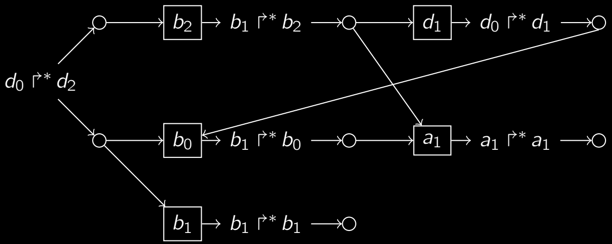
$\Longrightarrow$ efficient with a small number of processes per sort, while a very large number of sorts can be handled.

Extraction of Key Processes



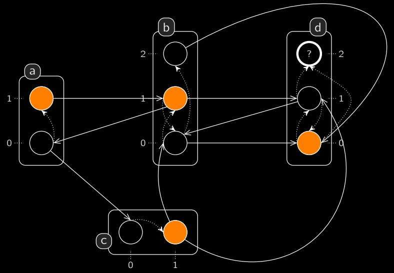
Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}^\omega_\varsigma$ such that:

- objective $\rightarrow$ follow at least one solution;
- process $\rightarrow$ follow all objectives;
- no cycle.



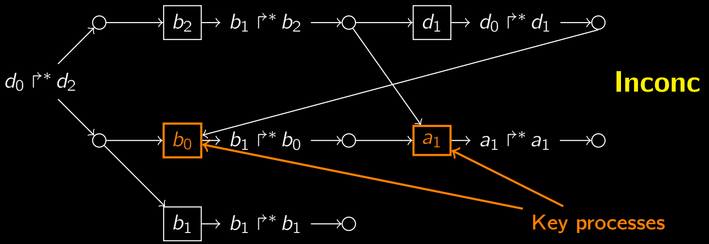
Inconc

Extraction of Key Processes

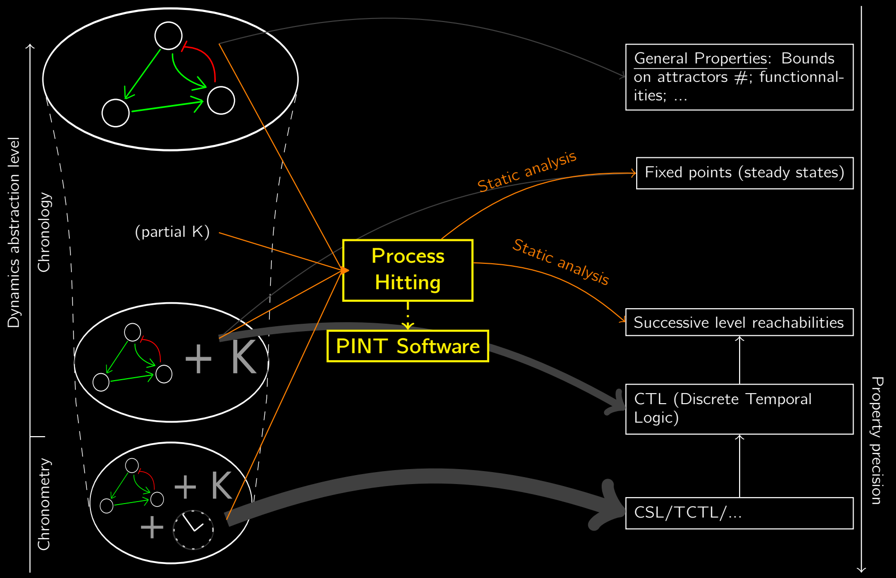


Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}_\varsigma^\omega$ such that:

- objective $\rightarrow$ follow at least one solution;
- process $\rightarrow$ follow all objectives;
- no cycle.



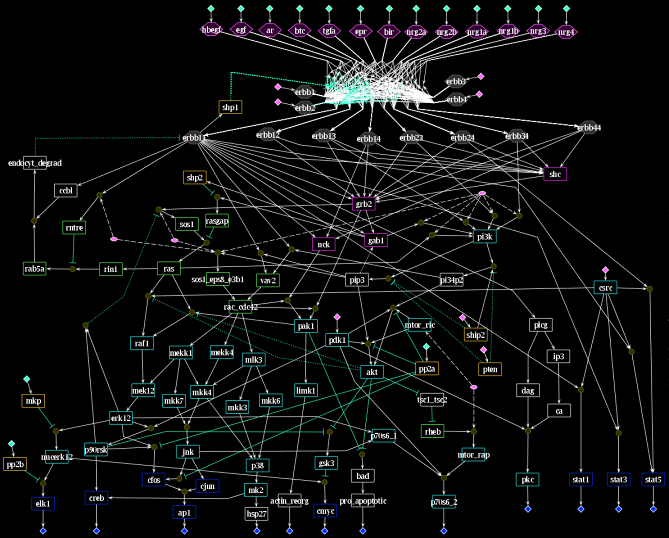
Outline



General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Property precision
Dynamics abstraction level
Chronology
Chronometry
(partial K)
+ K
+ K
+
Process Hitting
PINT Software
Static analysis
Static analysis

EGFR/ErbB Signalling Network

(104 components)



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

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

Execution times

- Real biological models.
- Wide-range of biological/arbitrary reachability analysis.
- Always conclusive.

Model	sorts	procs	actions	states	Biocham[1]	libDDD[2]	PINT[3]
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] `http://contraintes.inria.fr/biocham` (using NuSMV2)
[2] `http://move.lip6.fr/software/DDD`
[3] `http://process.hitting.free.fr`

Current work: signalling networks (TGF-β) with more than 8000 components.

Conclusion

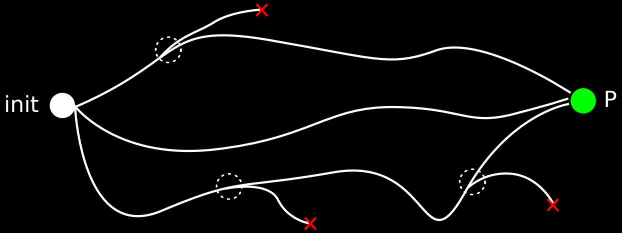
The Process Hitting

- Simple framework for dynamical complex systems;
- Abstract modelling of Biological Regulatory Networks;
- Future work: abstract modelling of biochemical networks.

Static Analysis by Abstract Interpretation of Process Hitting

- Very efficient over- and under-approximations of process reachability;
- Extract necessary processes for achieving reachabilities: towards control.
- Future work may establish other dynamical properties: attractors.

Outlook



Towards Quantitative analysis

- Static bifurcation analysis.
- Process Hitting with Priorities; Stochastic and Time Process Hitting;
- Identify key processes/actions/parameters (controlling bifurcations).

Thank you for your attention.

Approximations of Successive Reachability

Over-approximations

- Un-ordered approximation.
- Ordered approximation.
- Ordered Approximation with occurences order constraints.

No / Inconc

$\Uparrow$

Successive Reachability

$\Uparrow$

Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

Ordered Over-approximation

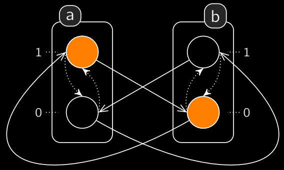
Goal: $\gamma_\varsigma(a_1 \upharpoonright^* a_0 :: \omega) \neq \emptyset \implies \gamma_{max\varsigma}(\omega) \neq \emptyset$

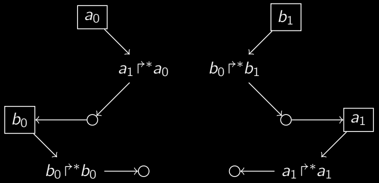
- By default, use the saturated context of $\lceil \mathcal{A}^\omega_\varsigma \rceil$:
$$\lceil \mathcal{A}^\omega_\varsigma \rceil = \mathrm{lfp} \left(\mathcal{A}^\omega_\varsigma \mapsto \mathcal{A}^\omega_{\varsigma \Cap \mathrm{procs}(\mathcal{A}^\omega_\varsigma)} \right) .$$
- $a_0 \notin \varsigma$, $\delta \in \gamma_\varsigma(a_1 \upharpoonright^* a_0 :: \omega)$
$\implies \delta = \delta_{1..n} :: c_i \rightarrow a_? \upharpoonright a_0 :: \delta_{m..|\delta|}$ with $\delta_{m..|\delta|} \in \gamma_{\varsigma'}(\omega)$
$\implies max\varsigma[a] = \{a_0\}$ and $max\varsigma[c] = \{c_i\}$.

Ordered Over-approximation

Goal: $\gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega) \neq \emptyset \Longrightarrow \gamma_{max\varsigma}(\omega) \neq \emptyset$

- By default, use the saturated context of $\lceil \mathcal{A}^\omega_\varsigma \rceil$:
$$\lceil \mathcal{A}^\omega_\varsigma \rceil = \mathrm{lfp} \left(\mathcal{A}^\omega_\varsigma \mapsto \mathcal{A}^\omega_{\varsigma \Cap \mathtt{procs}(\mathcal{A}^\omega_\varsigma)} \right) .$$
- $a_0 \notin \varsigma$, $\delta \in \gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega)$
$$\Longrightarrow \delta = \delta_{1..n} :: c_i \rightarrow a_? \Rsh a_0 :: \delta_{m..|\delta|} \text{ with } \delta_{m..|\delta|} \in \gamma_{\varsigma'}(\omega)$$
$$\Longrightarrow max\varsigma[a] = \{a_0\} \text{ and } max\varsigma[c] = \{c_i\}.$$



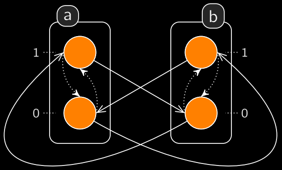


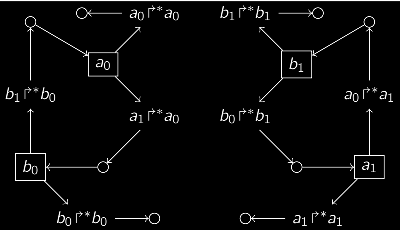
$\gamma_{\langle a_1, b_0 \rangle}(a_1 \Rsh^* a_0 :: b_0 \Rsh^* b_1) \neq \emptyset$

Ordered Over-approximation

Goal: $\gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega) \neq \emptyset \Longrightarrow \gamma_{max\varsigma}(\omega) \neq \emptyset$

- By default, use the saturated context of $\lceil \mathcal{A}^\omega_\varsigma \rceil$:
$$\lceil \mathcal{A}^\omega_\varsigma \rceil = \mathrm{lfp}\ \left(\mathcal{A}^\omega_\varsigma \mapsto \mathcal{A}^\omega_{\varsigma \Cap \mathrm{procs}(\mathcal{A}^\omega_\varsigma)} \right) \quad .$$
- $a_0 \notin \varsigma$, $\delta \in \gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega)$
$$\Longrightarrow \delta = \delta_{1..n} :: c_i \!\rightarrow a_? \Rsh a_0 :: \delta_{m..|\delta|} \text{ with } \delta_{m..|\delta|} \in \gamma_{\varsigma'}(\omega)$$
$$\Longrightarrow max\varsigma[a] = \{a_0\} \text{ and } max\varsigma[c] = \{c_i\}.$$



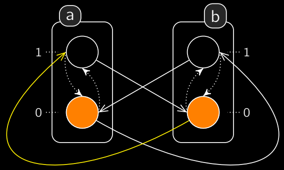


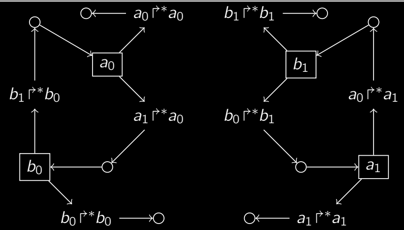
$\gamma_{\langle a_1, b_0 \rangle}(a_1 \Rsh^* a_0 :: b_0 \Rsh^* b_1) \neq \emptyset$

Ordered Over-approximation

Goal: $\gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega) \neq \emptyset \implies \gamma_{max\varsigma}(\omega) \neq \emptyset$

- By default, use the saturated context of $\lceil \mathcal{A}^\omega_\varsigma \rceil$:
$$\lceil \mathcal{A}^\omega_\varsigma \rceil = \mathrm{lfp}\left(\mathcal{A}^\omega_\varsigma \mapsto \mathcal{A}^\omega_{\varsigma \Cap \mathtt{procs}(\mathcal{A}^\omega_\varsigma)}\right) .$$
- $a_0 \notin \varsigma$, $\delta \in \gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega)$
$\implies \delta = \delta_{1..n} :: c_i \rightarrow a_? \Rsh a_0 :: \delta_{m..|\delta|}$ with $\delta_{m..|\delta|} \in \gamma_{\varsigma'}(\omega)$
$\implies max\varsigma[a] = \{a_0\}$ and $max\varsigma[c] = \{c_i\}$.



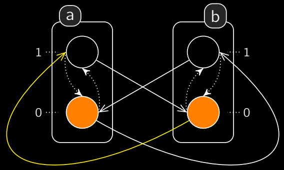


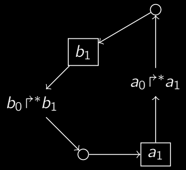
$$\gamma_{\langle a_1, b_0\rangle}(a_1 \Rsh^* a_0 :: b_0 \Rsh^* b_1) \neq \emptyset \implies \gamma_{\langle a_0, b_0\rangle}(b_0 \Rsh^* b_1) \neq \emptyset$$

Ordered Over-approximation

Goal: $\gamma_\varsigma(a_1 \upharpoonright^* a_0 :: \omega) \neq \emptyset \Longrightarrow \gamma_{max\varsigma}(\omega) \neq \emptyset$

- By default, use the saturated context of $\lceil \mathcal{A}^\omega_\varsigma \rceil$:
$$\lceil \mathcal{A}^\omega_\varsigma \rceil = \mathrm{lfp}\ \left(\mathcal{A}^\omega_\varsigma \mapsto \mathcal{A}^\omega_{\varsigma \Cap \mathrm{procs}(\mathcal{A}^\omega_\varsigma)} \right) \quad .$$
- $a_0 \notin \varsigma$, $\delta \in \gamma_\varsigma(a_1 \upharpoonright^* a_0 :: \omega)$
$$\Longrightarrow \delta = \delta_{1..n} :: c_i \rightarrow a_? \upharpoonright a_0 :: \delta_{m..|\delta|} \text{ with } \delta_{m..|\delta|} \in \gamma_{\varsigma'}(\omega)$$
$$\Longrightarrow max\varsigma[a] = \{a_0\} \text{ and } max\varsigma[c] = \{c_i\}.$$





$\gamma_{\langle a_1, b_0 \rangle}(a_1 \upharpoonright^* a_0 :: b_0 \upharpoonright^* b_1) \neq \emptyset \Longrightarrow \gamma_{\langle a_0, b_0 \rangle}(b_0 \upharpoonright^* b_1) \neq \emptyset$ **FAILURE**

Approximations of Successive Reachability

Over-approximations

- Un-ordered approximation.
- Ordered approximation.
- Ordered Approximation with occurences order constraints.

No / Inconc

⇑

Successive Reachability

⇑

Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

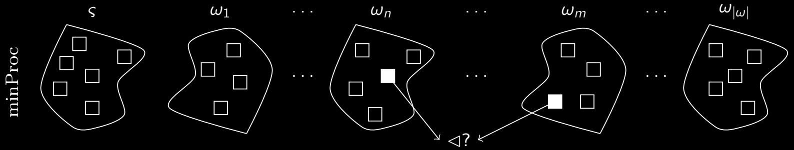
Process Occurrences Order Constraints

$a_j \lhd a_i \iff$ no scenario can be abstracted by $a_i \Rsh^* a_j$.

Uncovering Order Constraints

$\mathbf{BS}(a_i \Rsh^* a_j) = \emptyset \implies a_j \lhd a_i$

Idea of Over-Approximation



Based on the ordered over-approximation:

- $\mathrm{minProc}_\varsigma(\omega_n) = \{p \in \mathbf{Proc} \mid p$ occurs in all solutions of $\omega_n\}$;
- $$\begin{cases} \{a_i \in \varsigma\} & \text{if } n = 0 \\ \mathrm{minProc}_{max\varsigma}(\omega_n) & \text{otherwise,} \\ & \text{with } max\varsigma = \mathrm{maxCtx}(\varsigma, w, n-1) \end{cases}.$$