Modelling, Simulation and Verification of Large Biological Regulatory Networks

Journée AFSEC - Méthodes formelles pour la bio-informatique - 20 octobre 2011

Loïc Paulevé[1,2], Morgan Magnin[2], Olivier Roux[2]

[1]: École Polytechnique / LIX (équipe AMIB)
pauleve@lix.polytechnique.fr
http://loicpauleve.name

[2]: École Centrale de Nantes / IRCCyN (équipe MeForBio)
morgan.magnin@irccyn.ec-nantes.fr
olivier.roux@irccyn.ec-nantes.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.

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]

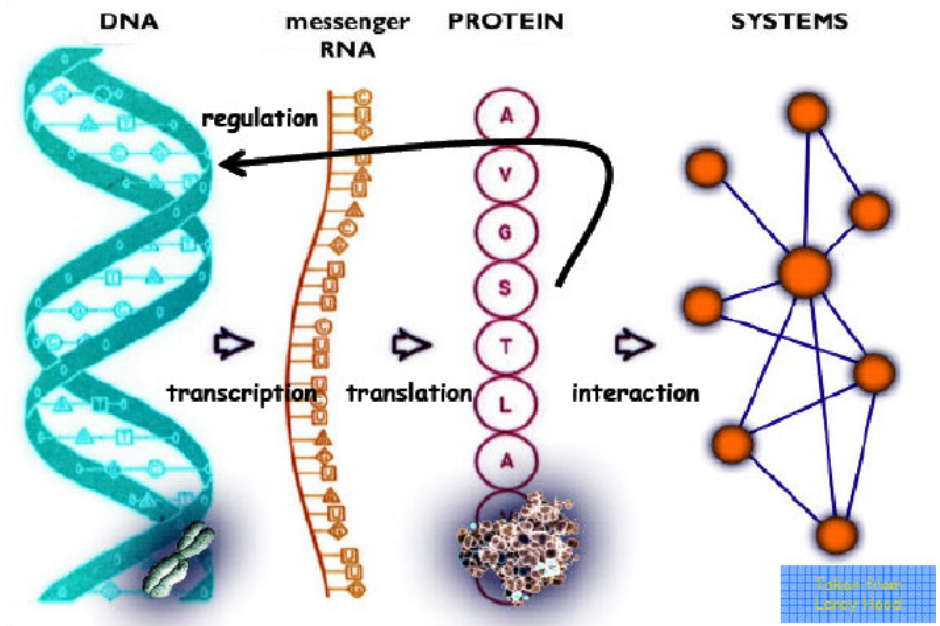
- *Elementary* framework for dynamical complex systems;
- Applied to BRNs; not limited to.
- Stochastic and Time dimensions (simulation + standard model checking).
- Software available (PINT - `http://process.hitting.free.fr`).

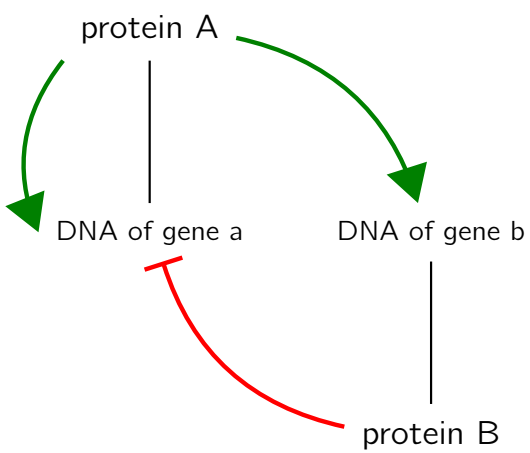
Large-scale model checking (discrete dynamical properties)

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

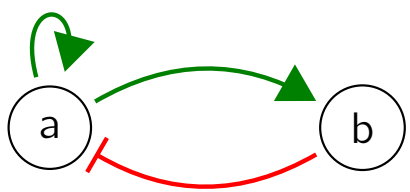
Biological Regulatory Networks (BRNs)

The interaction graph





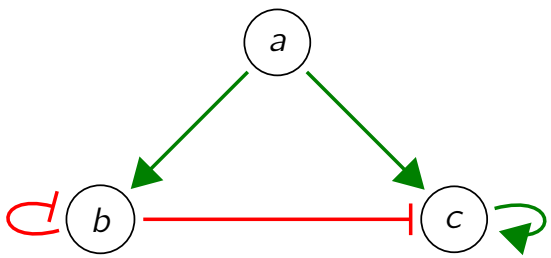
Interaction graph



Discrete Networks

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

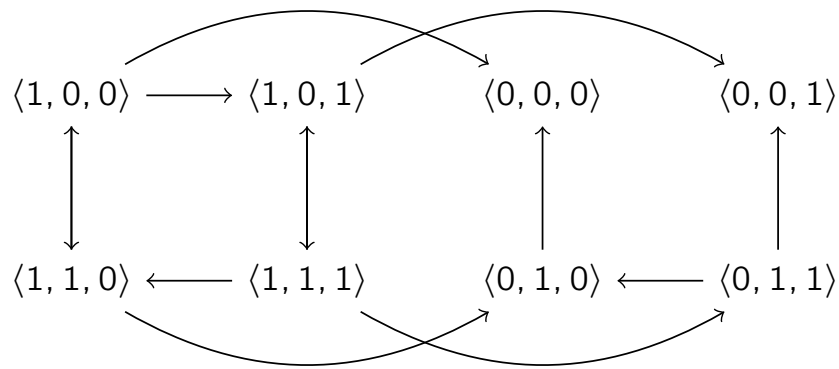
Boolean example:



$$f^a(x) = 0$$

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

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



[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

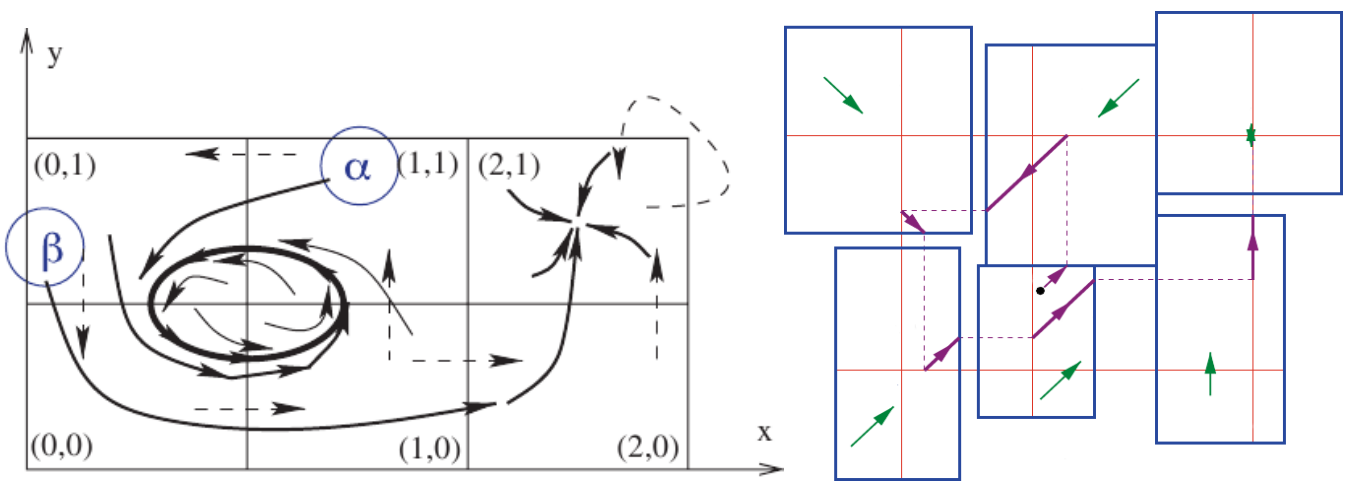
Introduce delays to actions

Stochastic Models

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

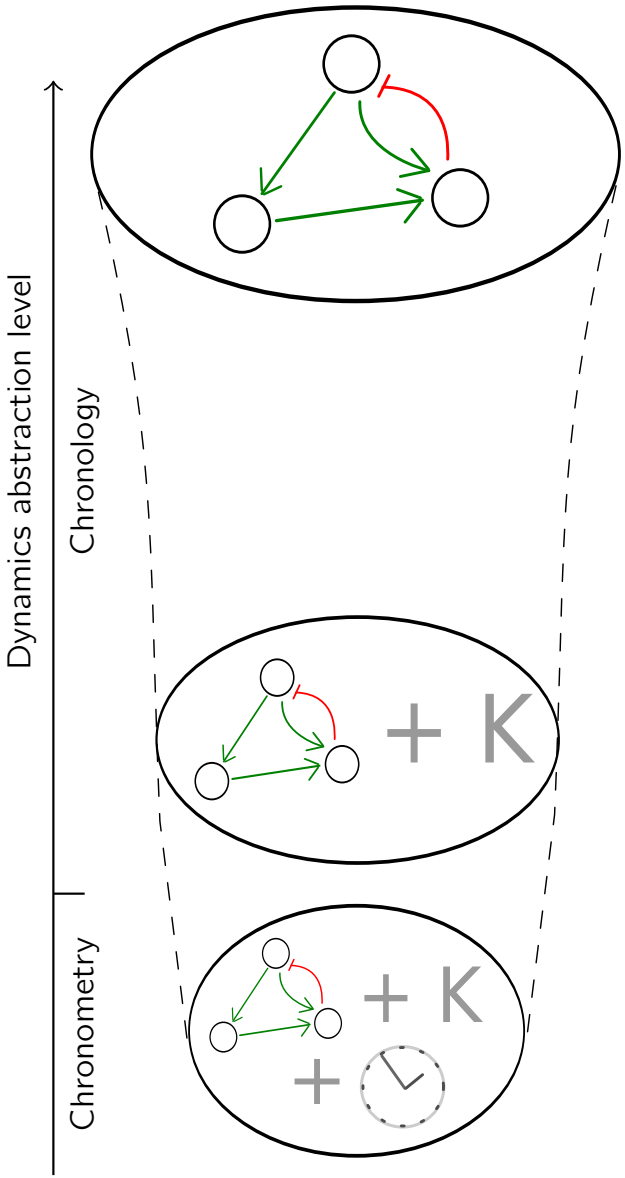
Stochastic Petri Nets / π-calculus, etc. [Heiner, Regev, Priami, Phillips, etc.]

Timed Models

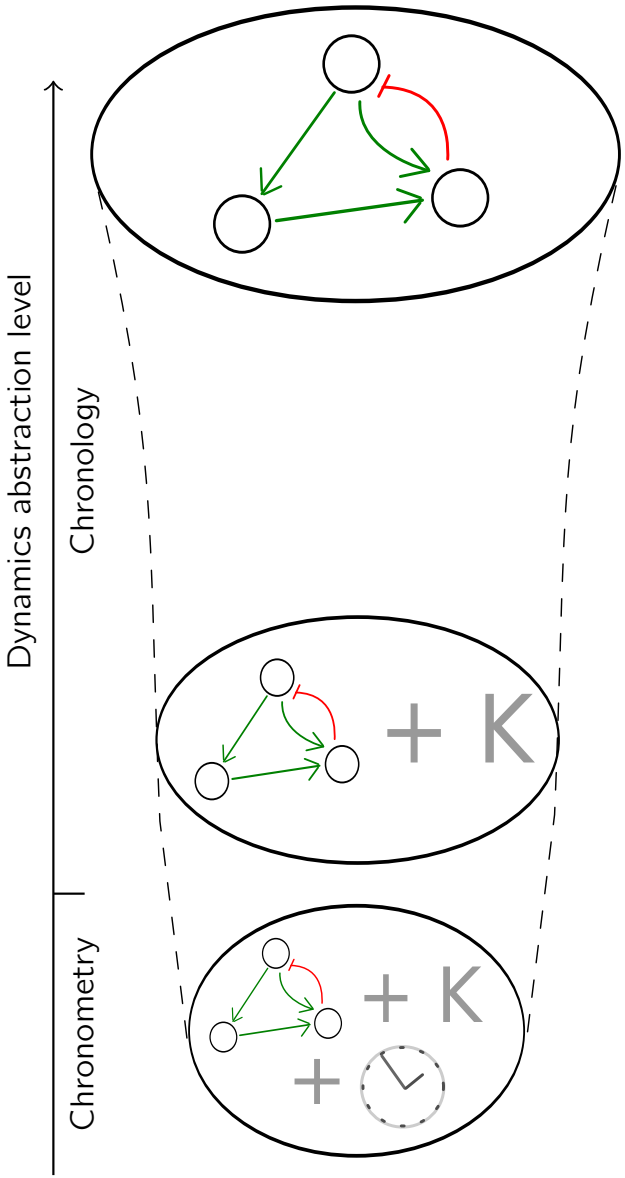


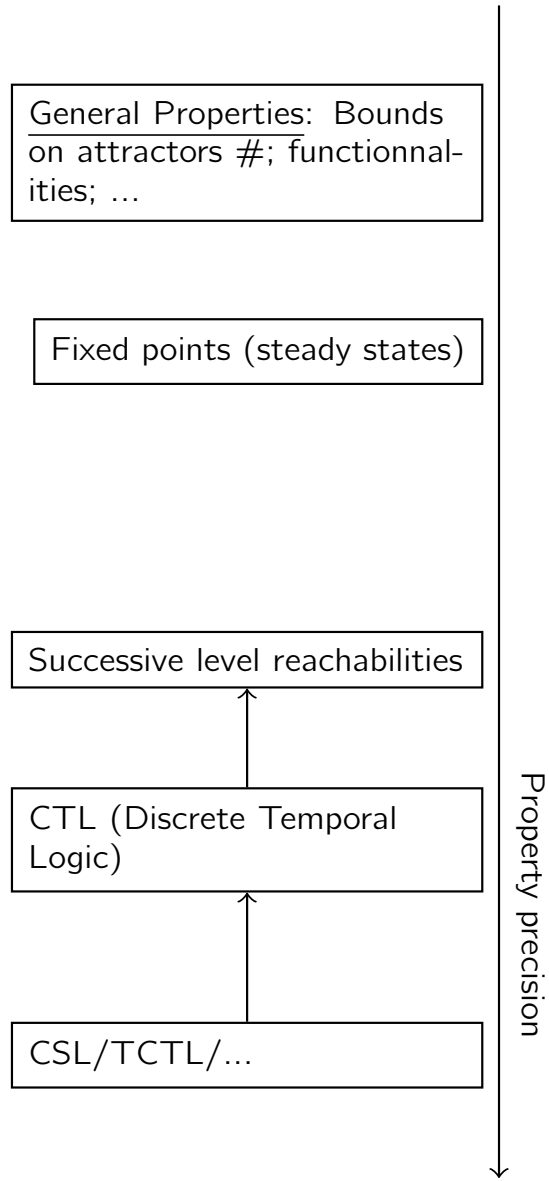
Timed / Hybrid Automata [Ahmad, Roux, Batt, Bockmayr, etc.]

State of the Art

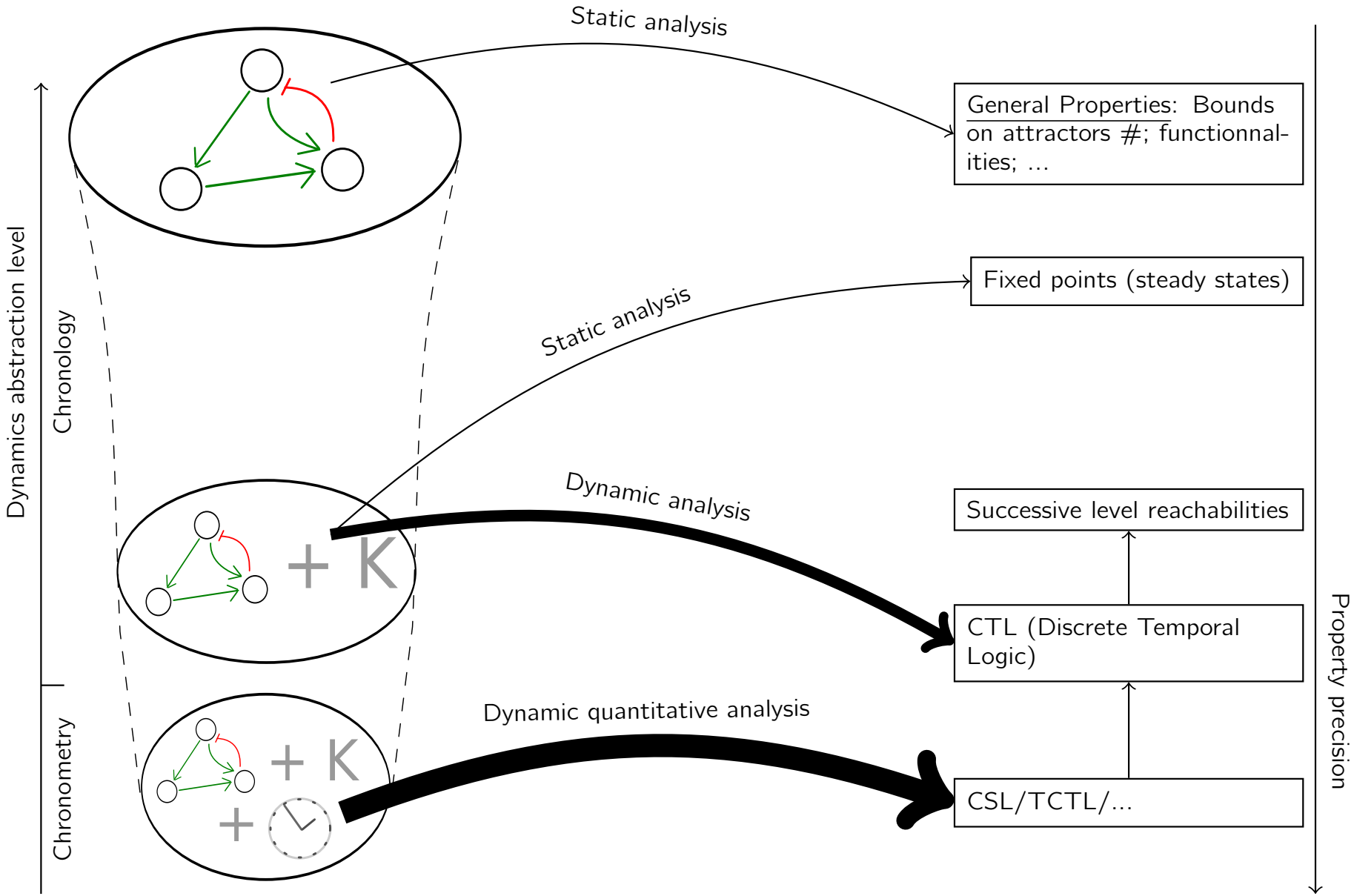


State of the Art

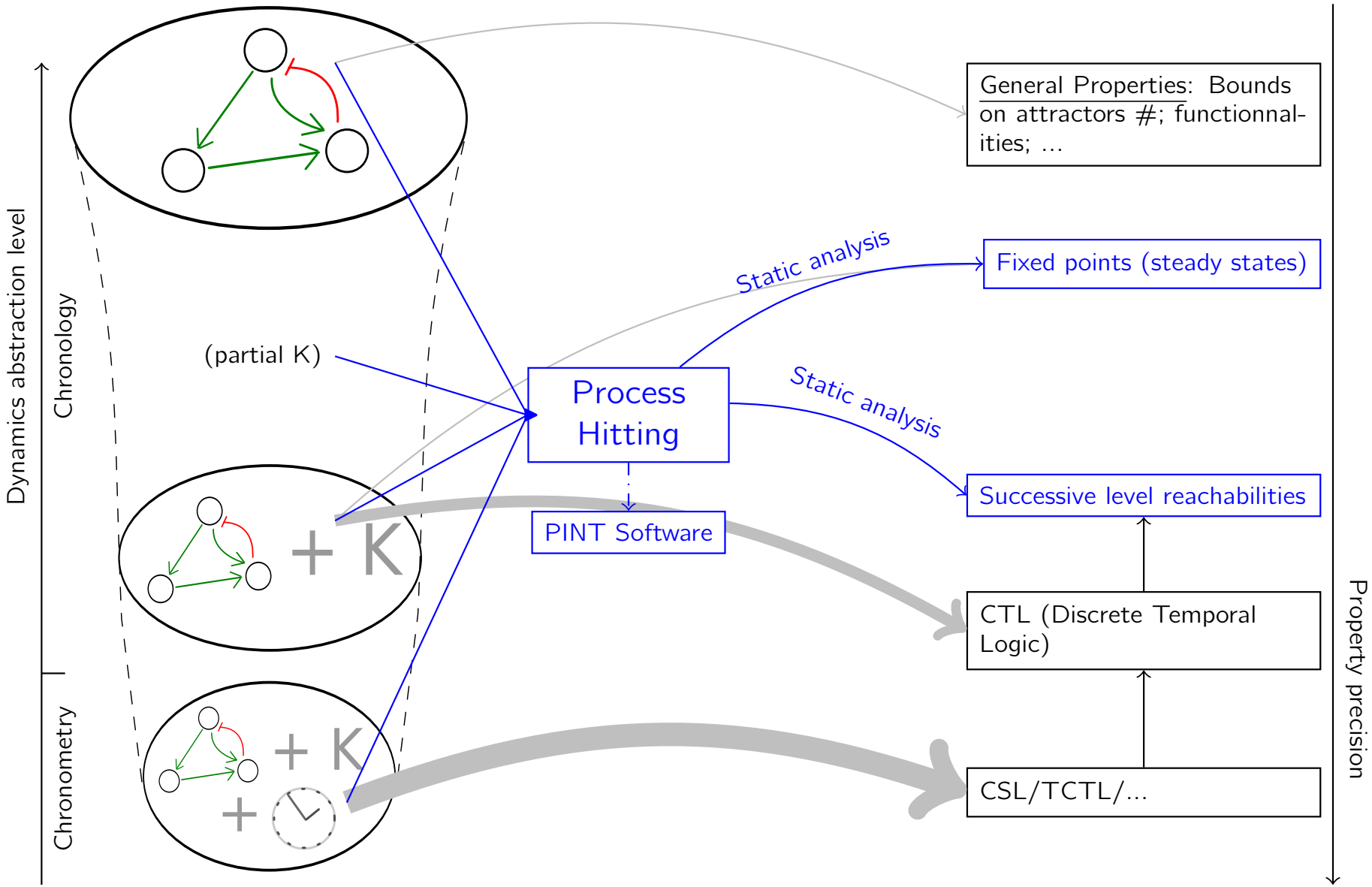




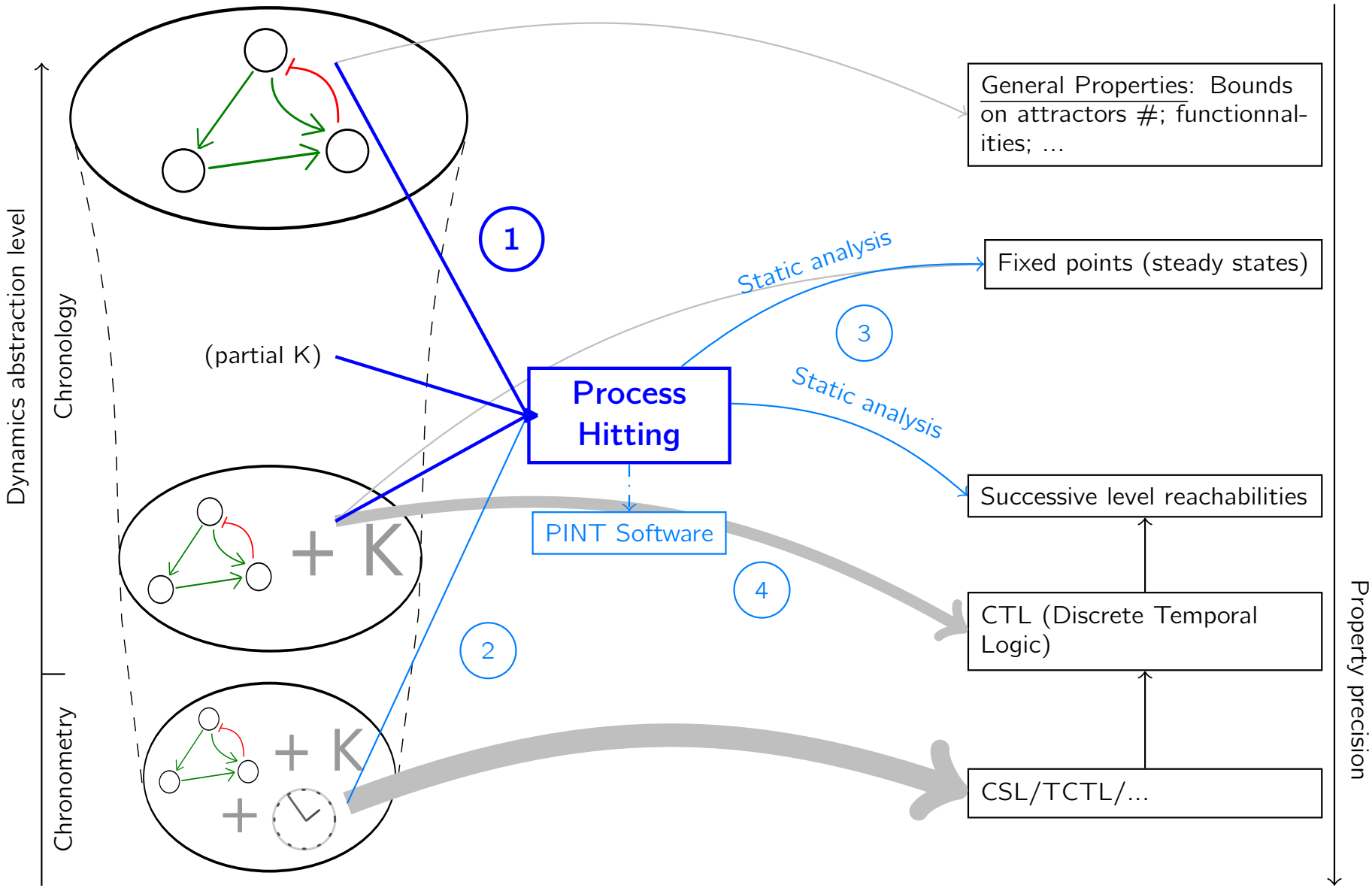
State of the Art



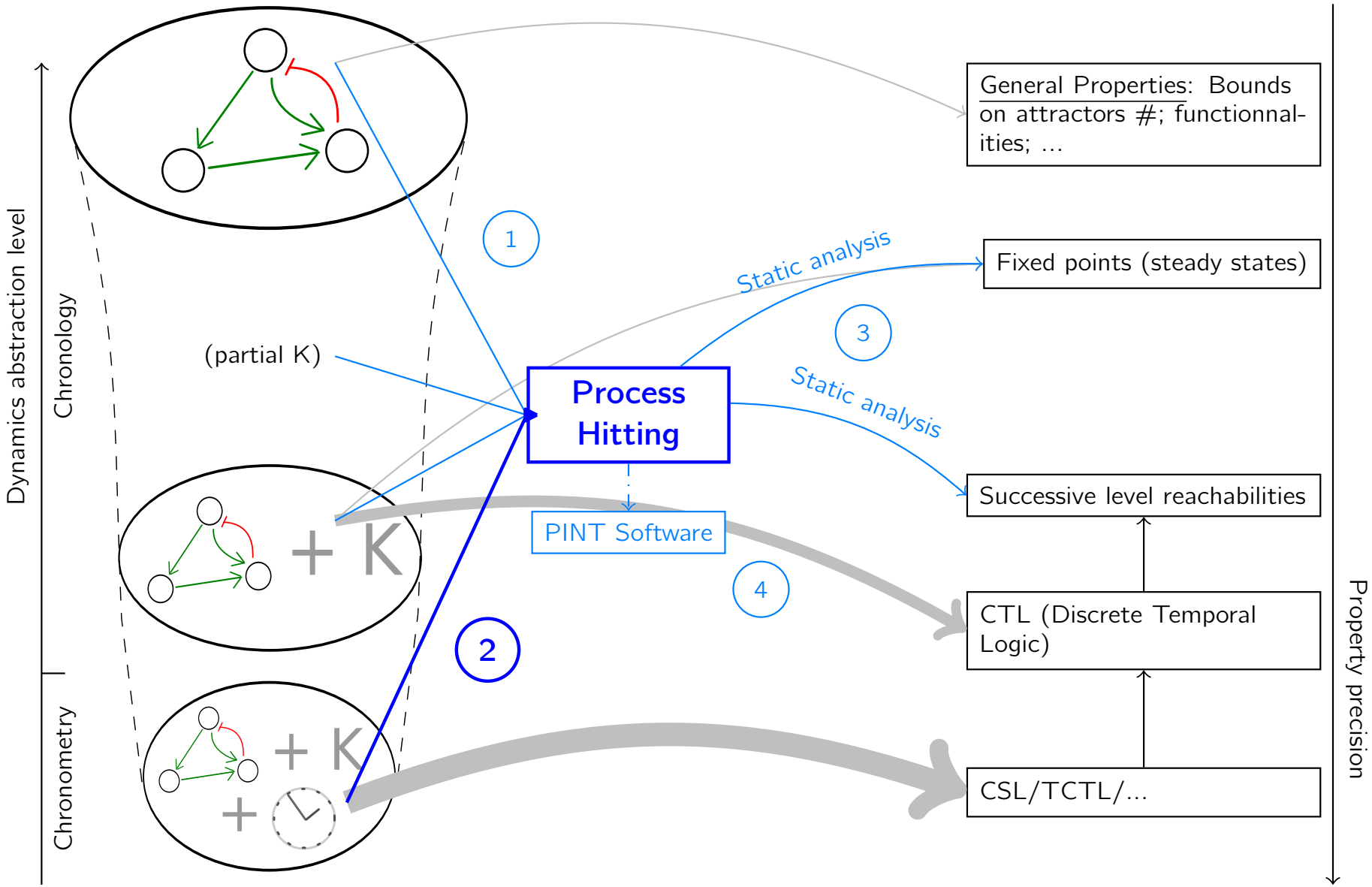
Contributions



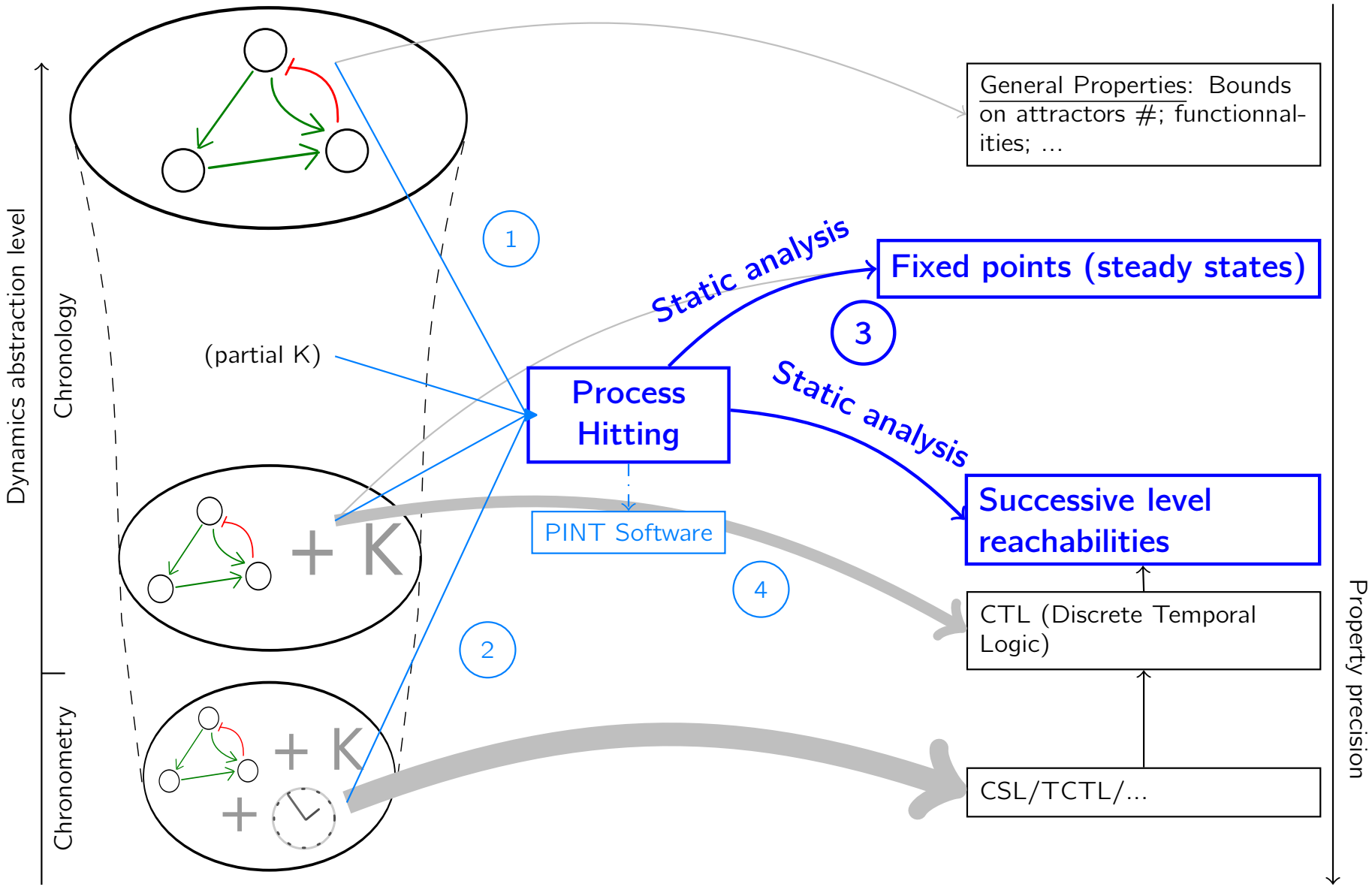
Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
(partial K)
+ K
+ K
+
1
2
3
4
Static analysis
Static analysis
Process Hitting
PINT Software
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...

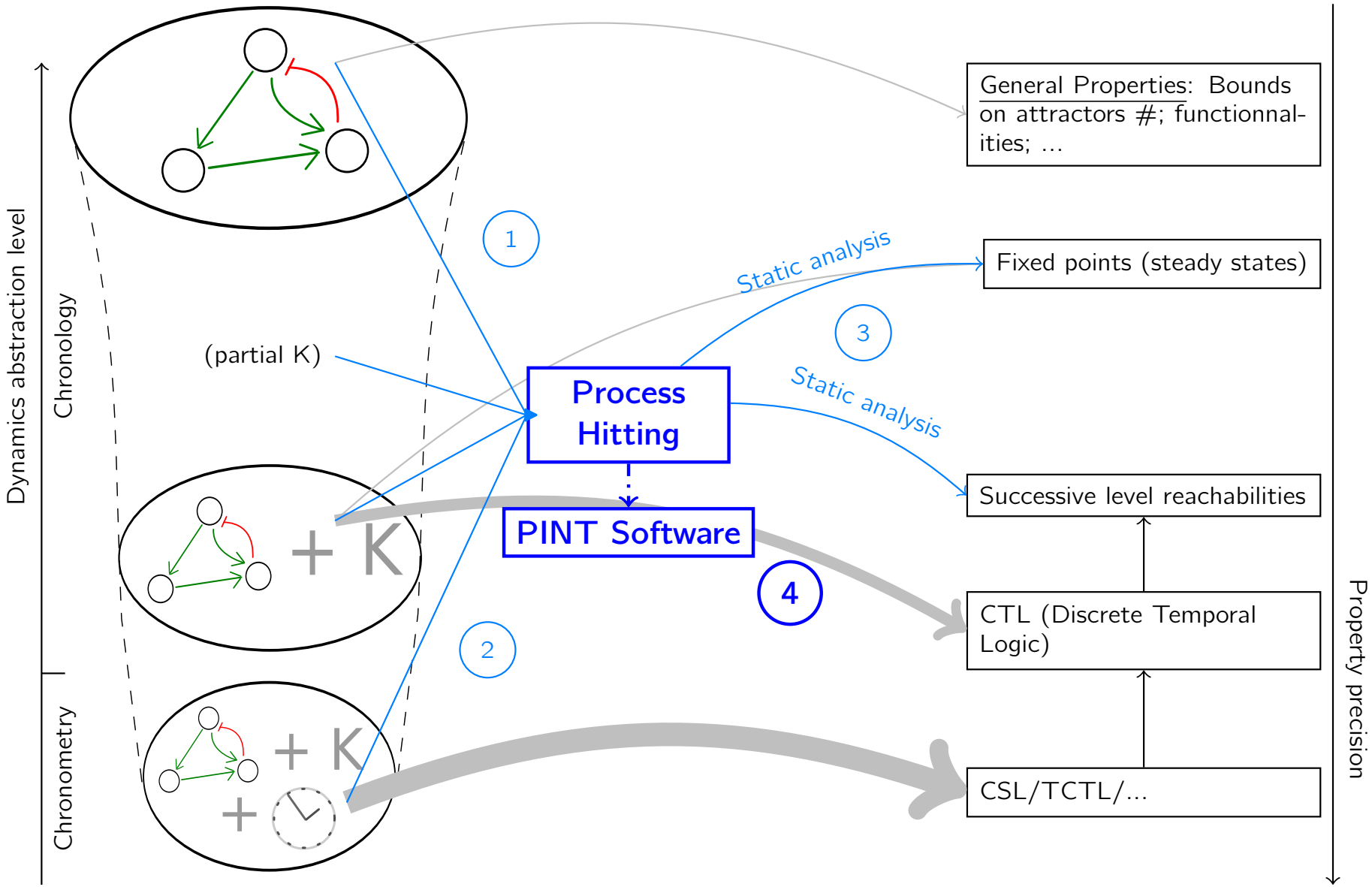
Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
(partial K)
+ K
+ K
+
1
2
3
4
Static analysis
Static analysis
Process Hitting
PINT Software
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...

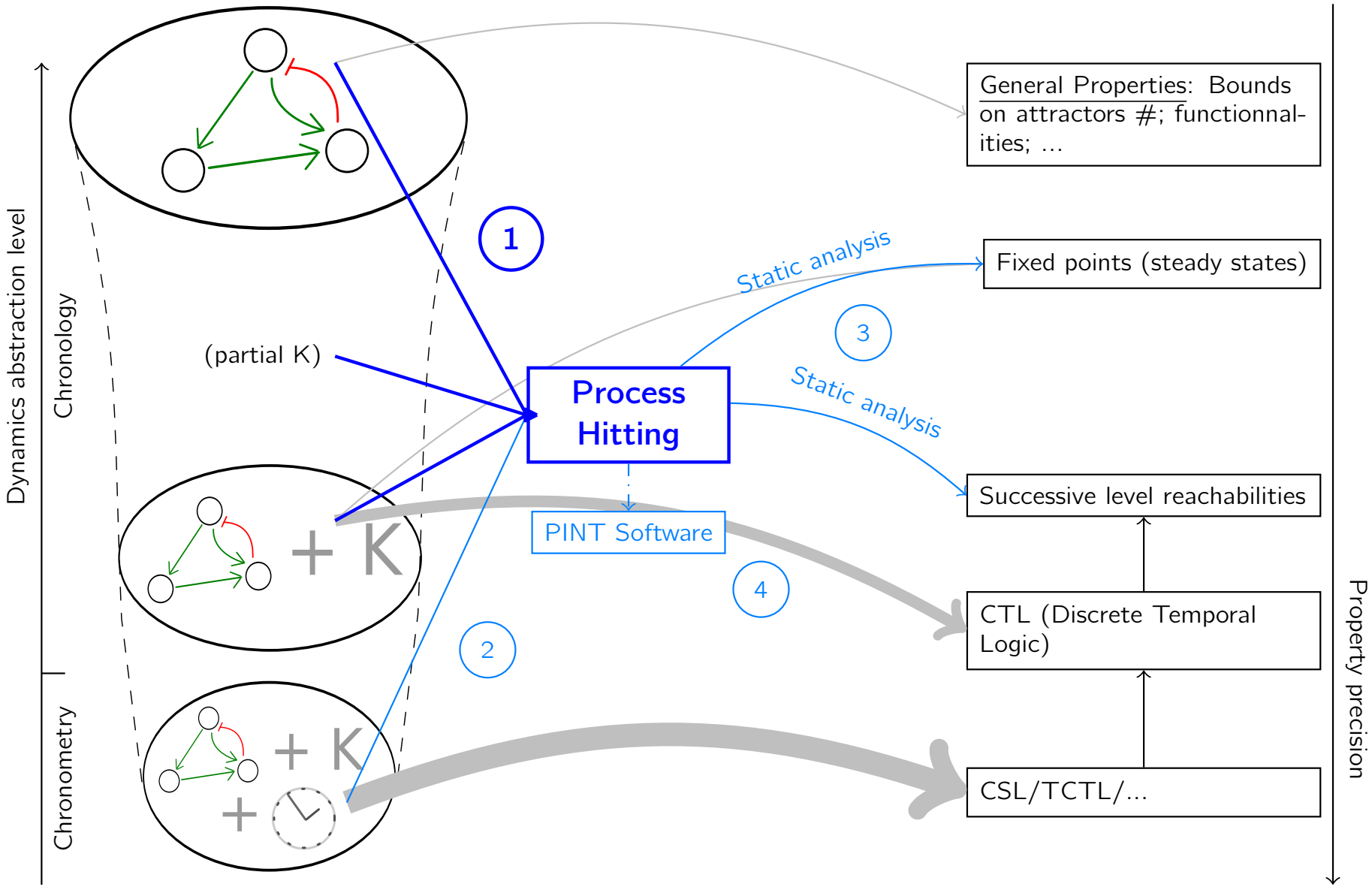
Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
General Properties: Bounds on attractors #; functionnalities; ...
1
Static analysis
Fixed points (steady states)
3
(partial K)
Process Hitting
Static analysis
Successive level reachabilities
PINT Software
4
CTL (Discrete Temporal Logic)
2
+ K
+ K
+
CSL/TCTL/...

Outline

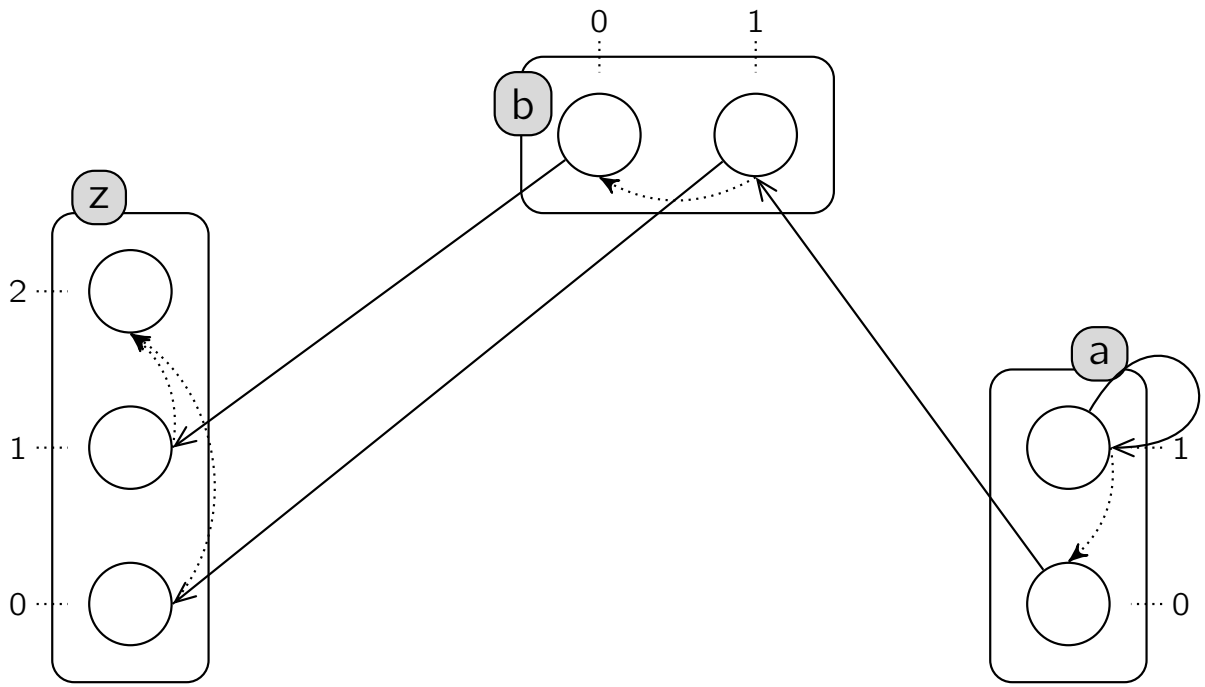

Dynamics abstraction level
Chronology
Chronometry
Property precision
(partial K)
+ K
+ K
+
1
2
3
4
Static analysis
Static analysis
Process Hitting
PINT Software
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...

Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
(partial K)
+ K
+ K
+
1
2
3
4
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/...

The Process Hitting Framework

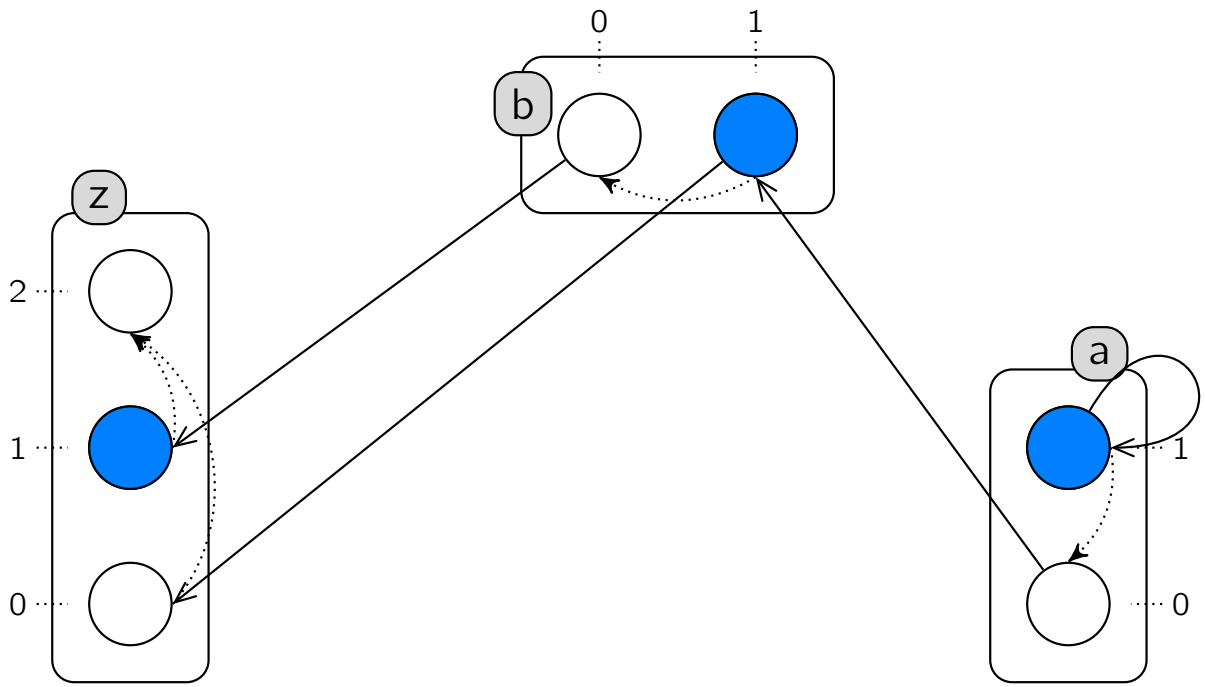
[Paulevé, Magnin, Roux in TCSB 2011]



- 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

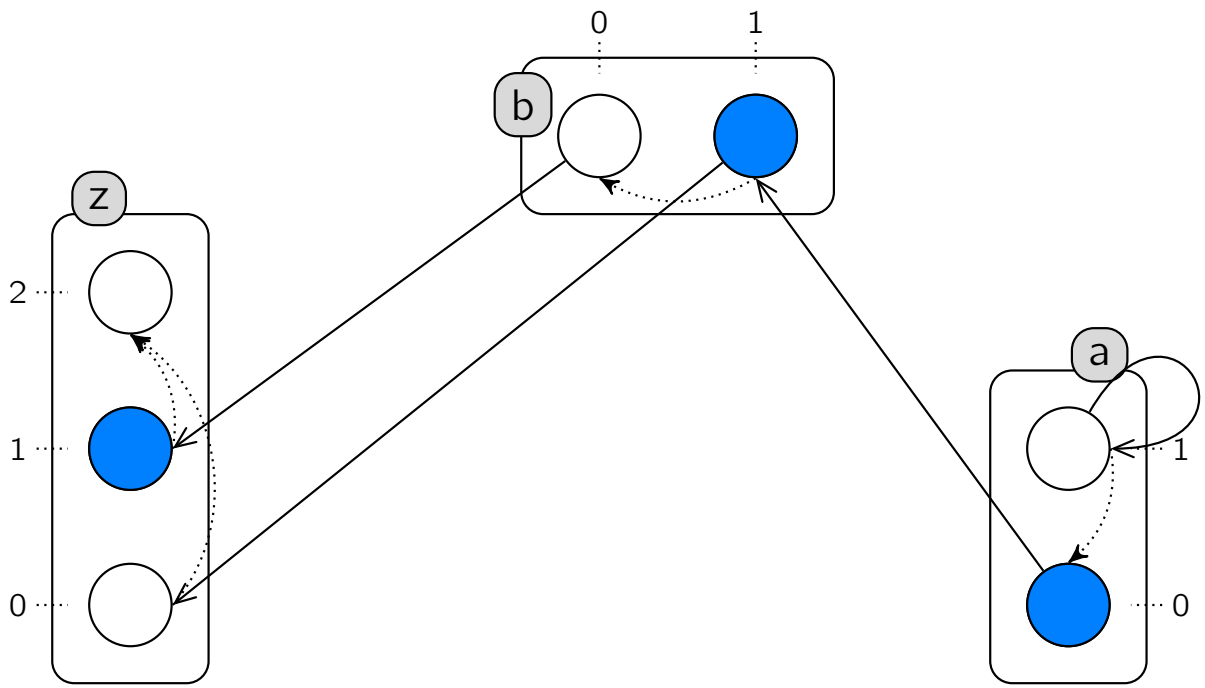
[Paulevé, Magnin, Roux in TCSB 2011]



- 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

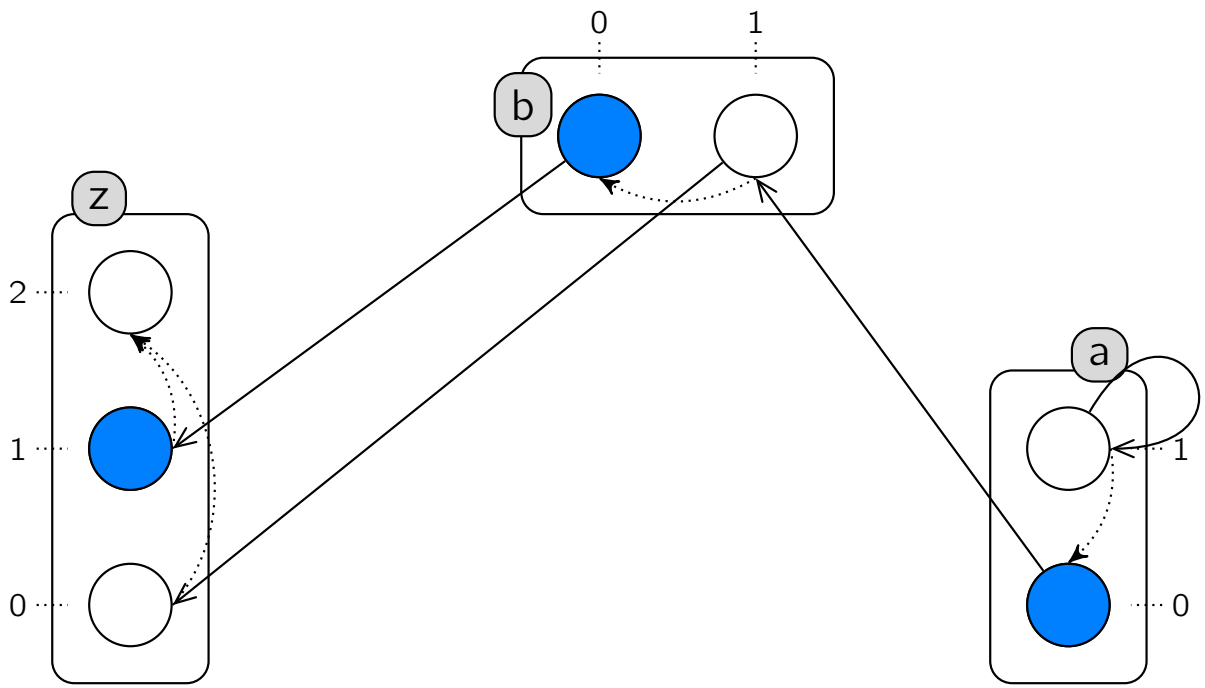
[Paulevé, Magnin, Roux in TCSB 2011]



- 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

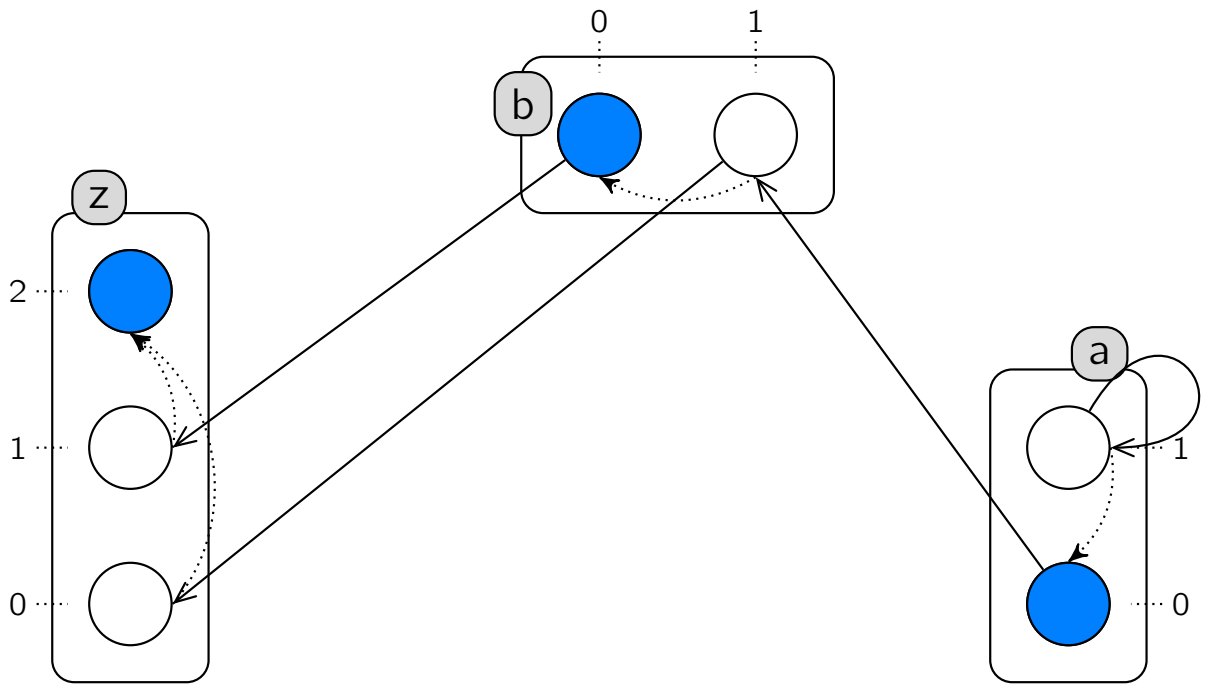
[Paulevé, Magnin, Roux in TCSB 2011]



- 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]

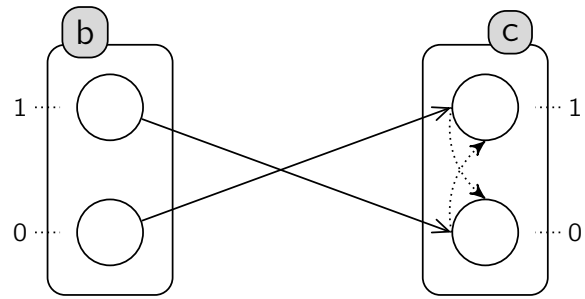


- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines (CFSM).

Generalized Dynamics of BRNs

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

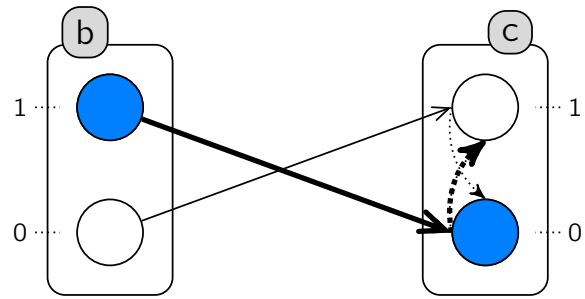
Boolean case:



Generalized Dynamics of BRNs

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

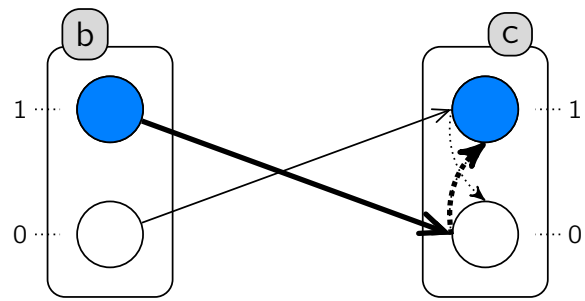
Boolean case:



Generalized Dynamics of BRNs

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

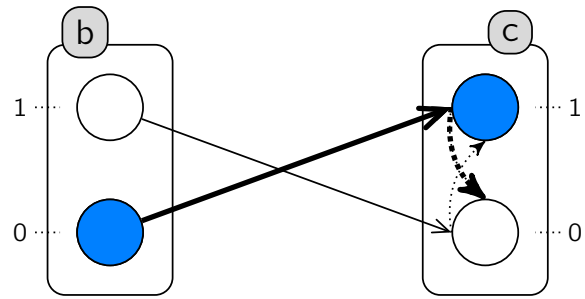
Boolean case:



Generalized Dynamics of BRNs

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

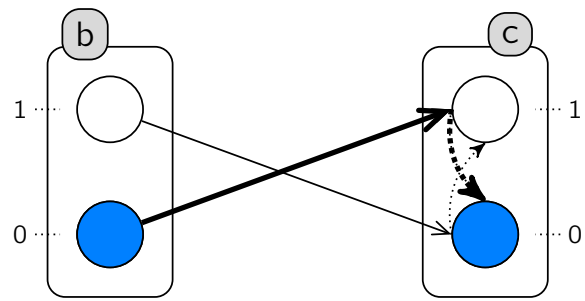
Boolean case:



Generalized Dynamics of BRNs

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

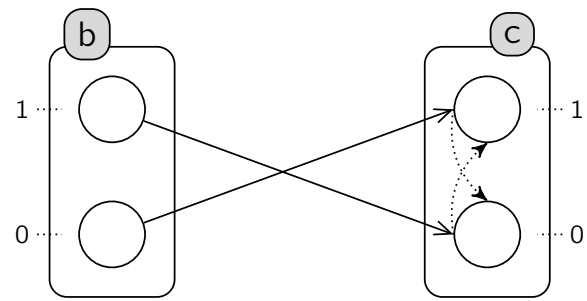
Boolean case:

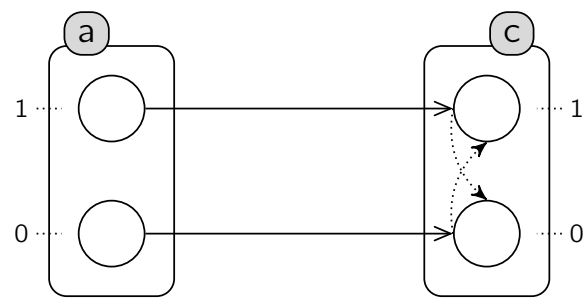


Generalized Dynamics of BRNs

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

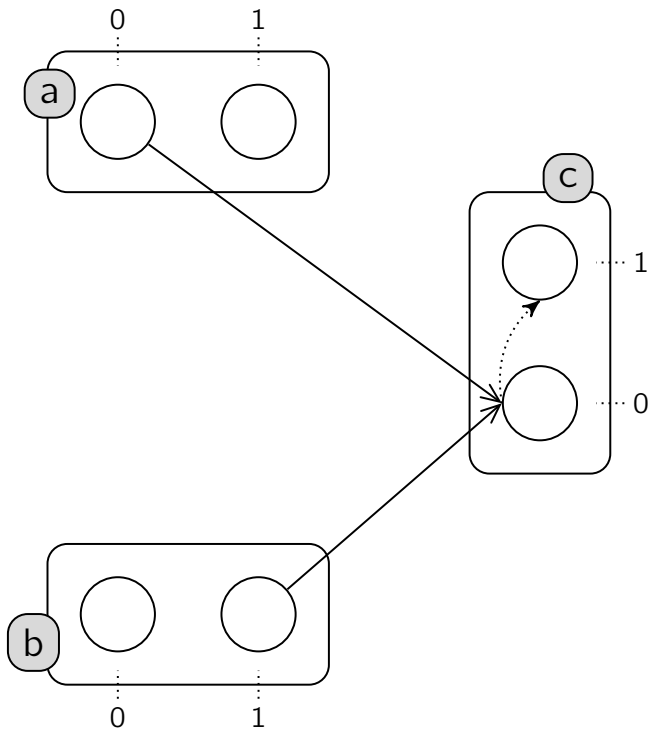
Boolean case:





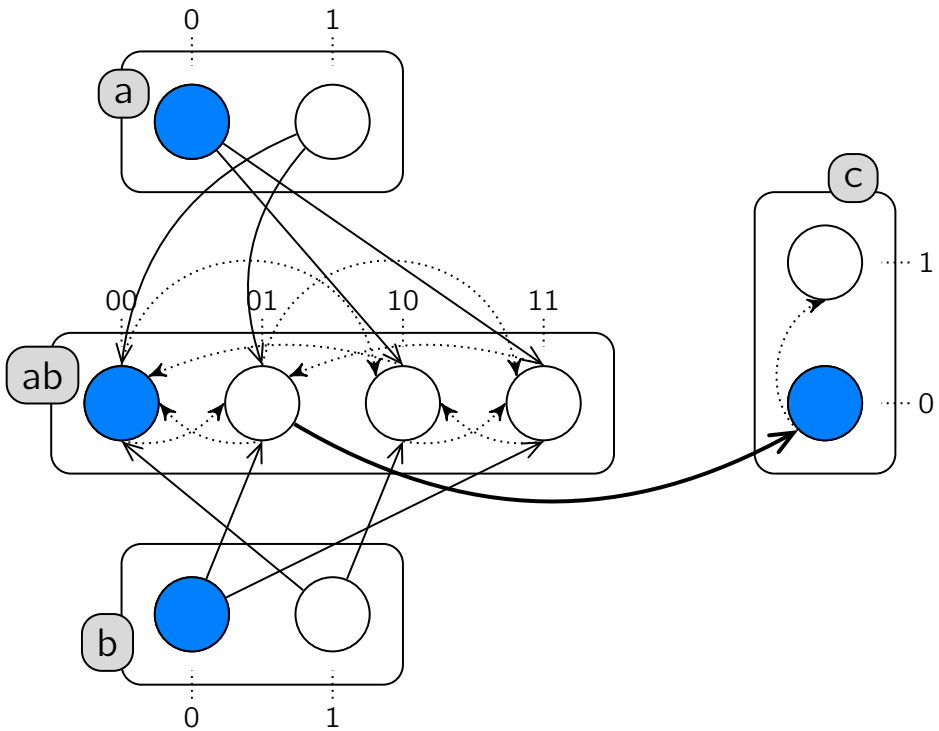
Refining with Cooperation

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



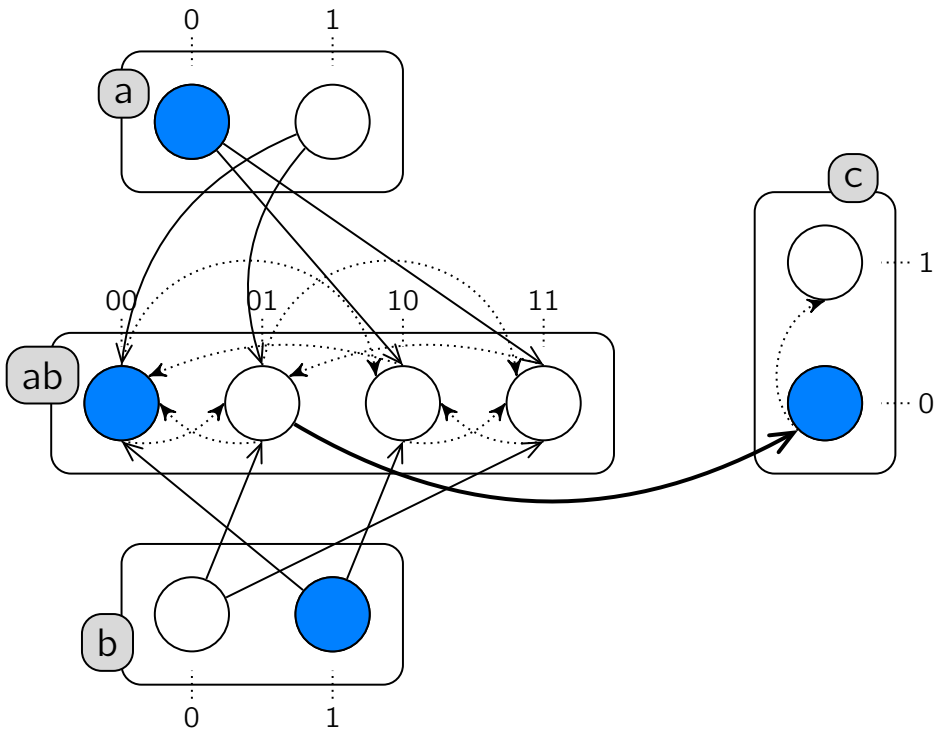
Refining with Cooperation

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



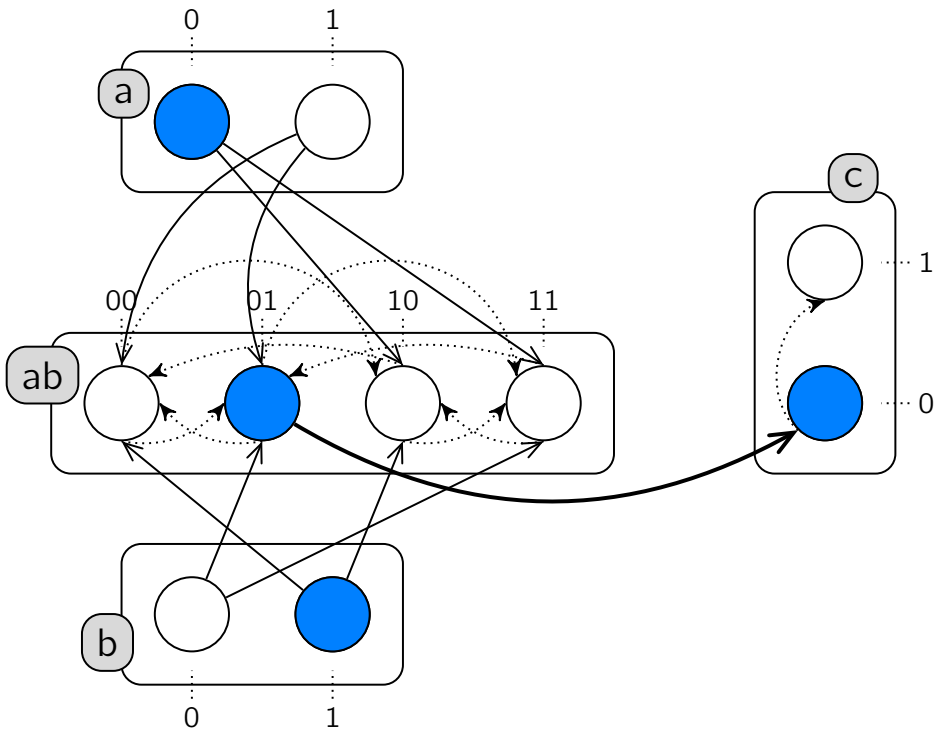
Refining with Cooperation

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



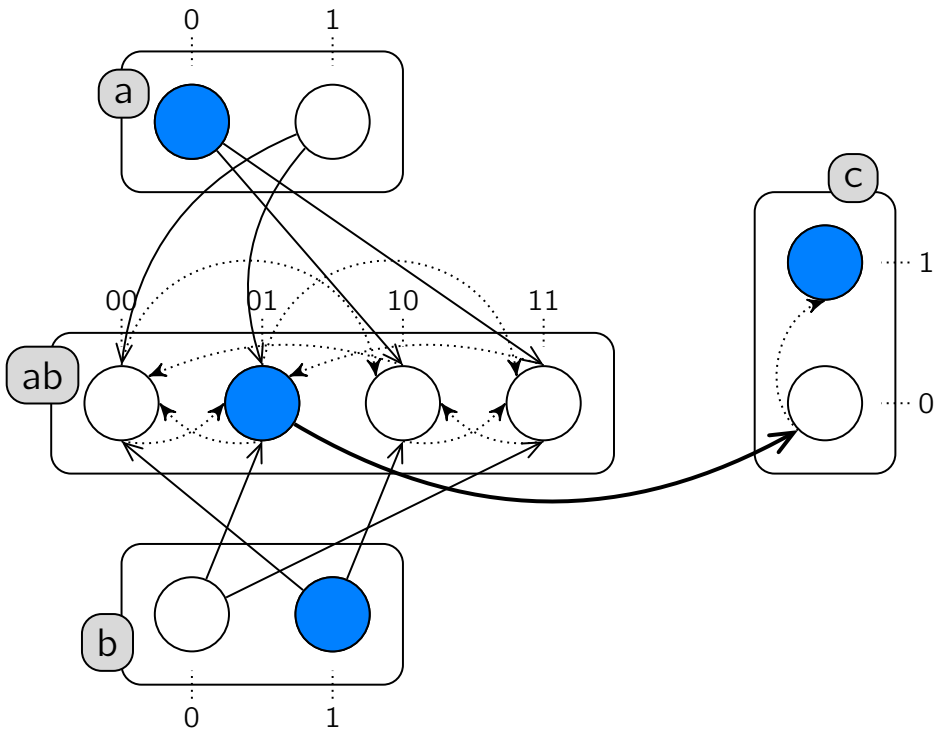
Refining with Cooperation

- Idea: $c_0 \Rsh c_1$ 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

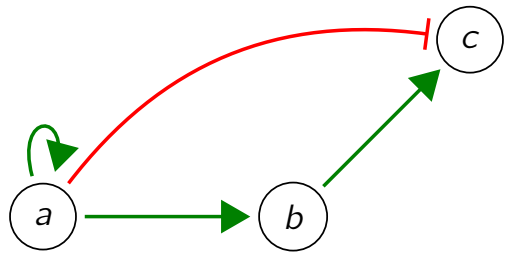
- Idea: $c_0 \overset{\uparrow}{\cdot} c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative sort reflecting the state of the sorts a and b.

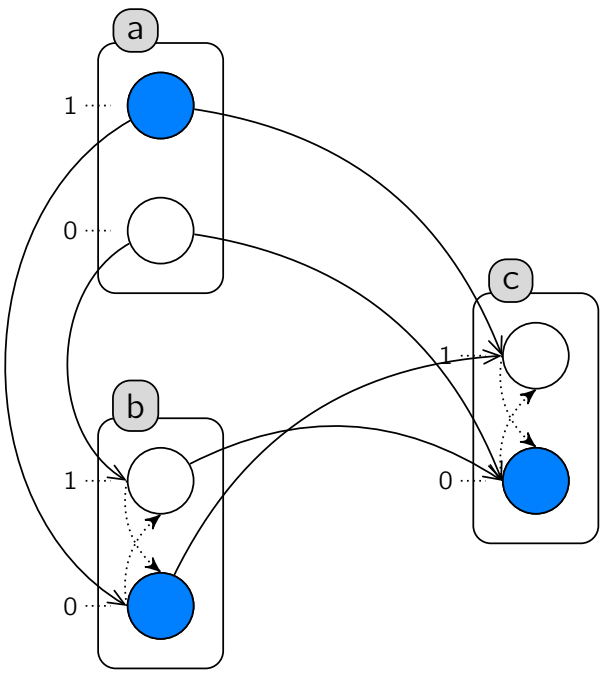


$\Rightarrow$ introduce a temporal shift; similar to complexes.

Toy example

Incoherent feed-forward loop





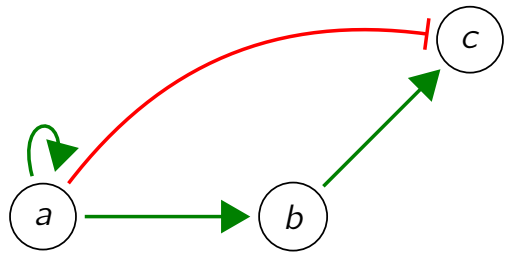
$\langle a_1, b_0, c_0 \rangle$

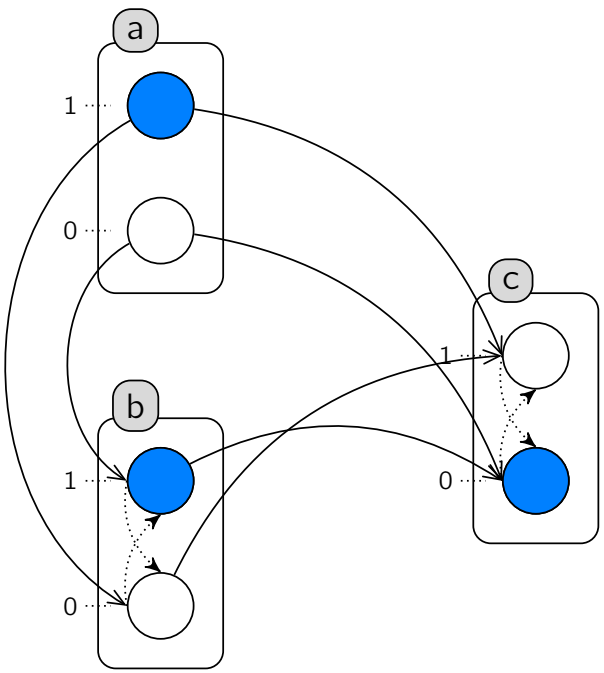
$\downarrow$

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

Toy example

Incoherent feed-forward loop





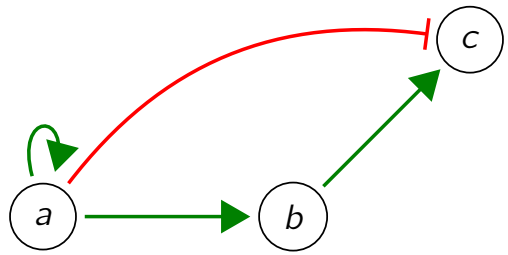
$\langle a_1, b_0, c_0 \rangle$

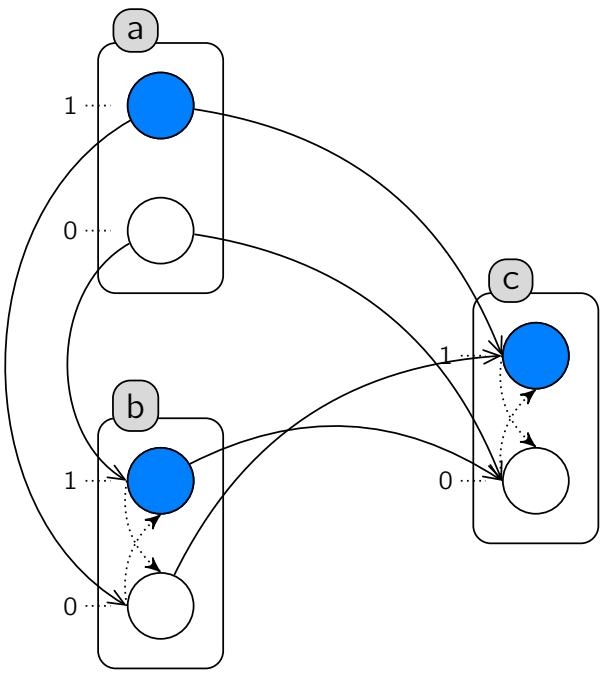
$\downarrow$

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

Toy example

Incoherent feed-forward loop





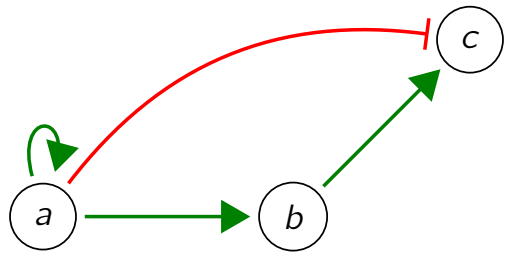
$\langle a_1, b_0, c_0 \rangle$

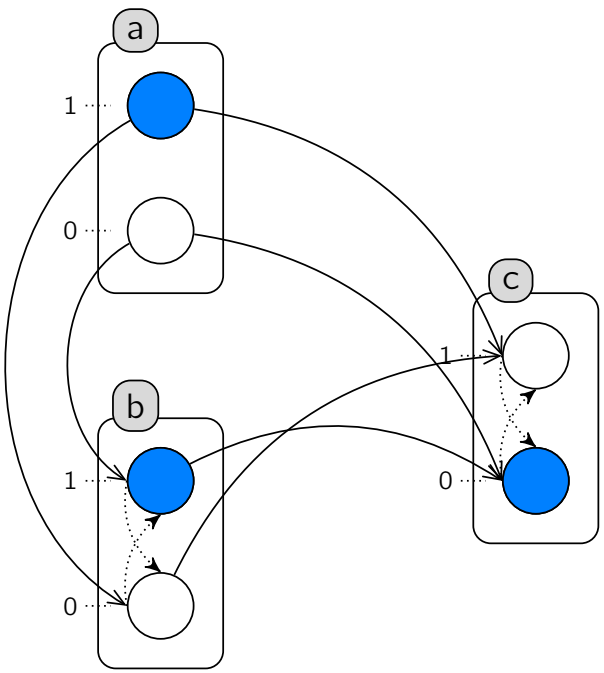
$\downarrow$

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

Toy example

Incoherent feed-forward loop





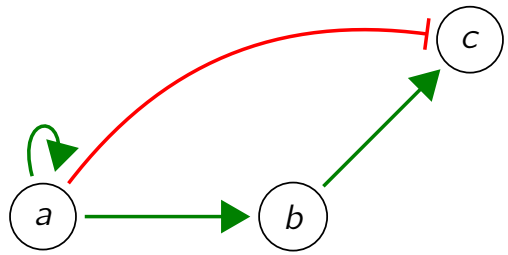
$\langle a_1, b_0, c_0 \rangle$

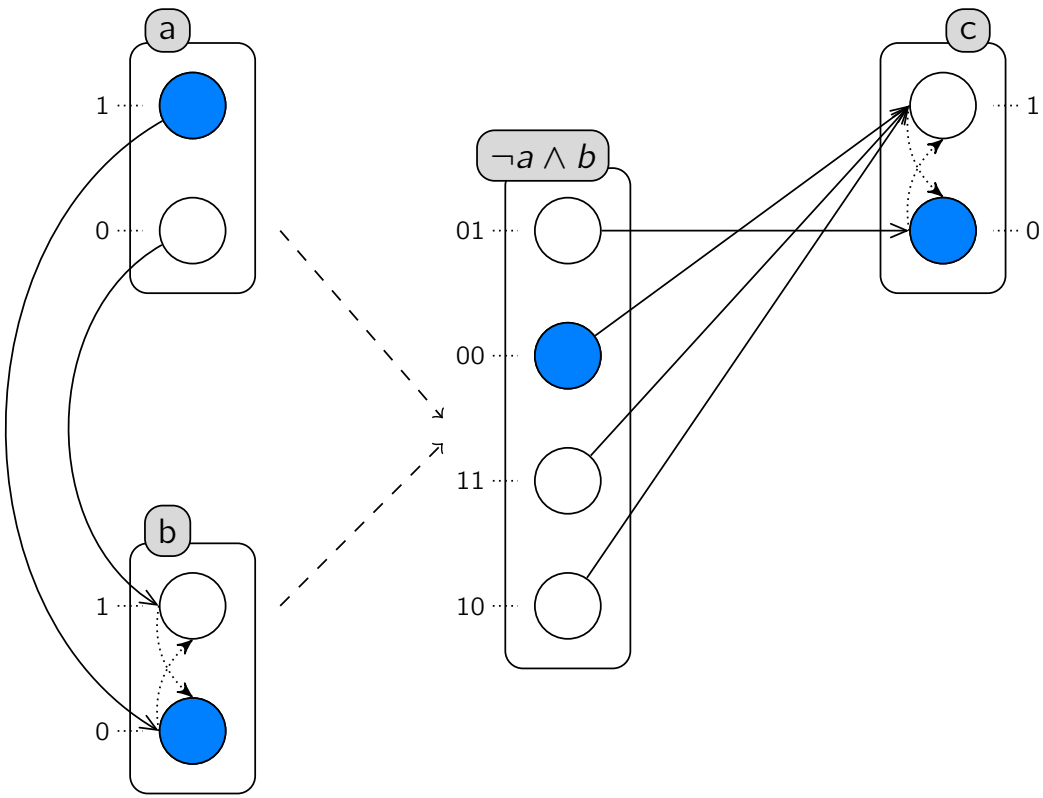
$\downarrow$

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

Toy example

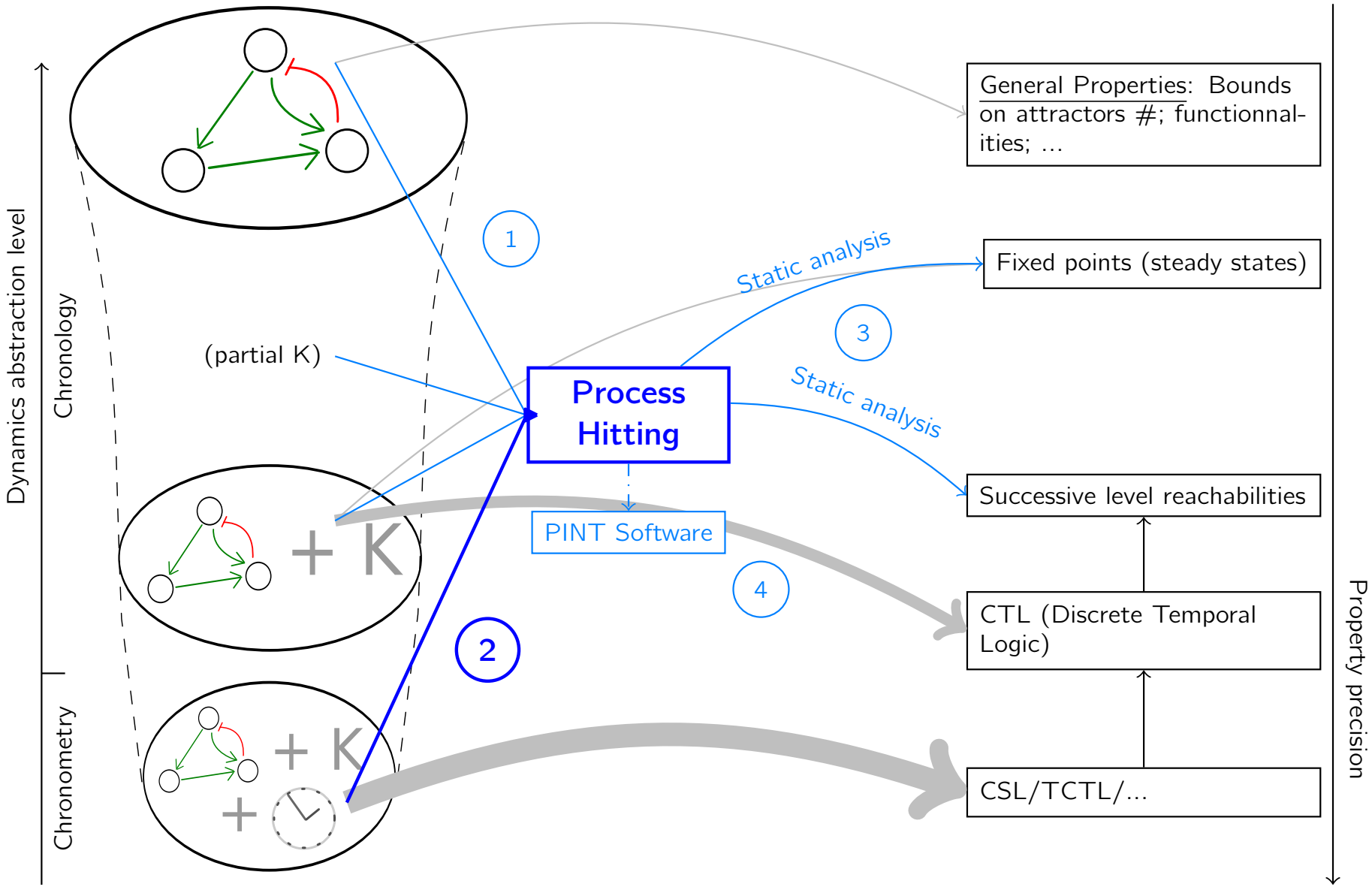
Incoherent feed-forward loop





$$\langle a_1, b_0, ab_{00}, c_0\rangle \longrightarrow \langle a_1, b_0, ab_{10}, c_0\rangle$$
$$\downarrow \qquad\qquad \downarrow$$
$$\langle a_1, b_1, ab_{00}, c_0\rangle \longrightarrow \langle a_1, b_1, ab_{10}, c_0\rangle$$
$$\downarrow \qquad\qquad \downarrow$$
$$\langle a_1, b_1, ab_{01}, c_0\rangle \longrightarrow \langle a_1, b_1, ab_{11}, c_0\rangle$$
$$\downarrow \qquad\qquad \uparrow$$
$$\langle a_1, b_1, ab_{01}, \mathbf{c_1}\rangle \qquad\qquad \uparrow$$
$$\downarrow \qquad\qquad \uparrow$$
$$\langle a_1, b_1, ab_{11}, \mathbf{c_1}\rangle$$

Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...
Process Hitting
PINT Software
(partial K)
+ K
+ K
+
Static analysis
Static analysis
1
2
3
4

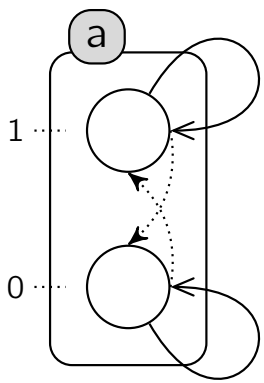
Introduction of Time and Stochastic Features

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

Goal: tune time features within a stochastic framework.

Action parameters

- Rate r + stochasticity absorption factor sa.
- Action duration $\approx$ sum of sa exponentials at rate $r.sa$.
- Mean duration: r^{-1}; variance: $r^{-2}sa^{-1}$ (Erlang distribution).



(done in the stochastic π-calculus framework)

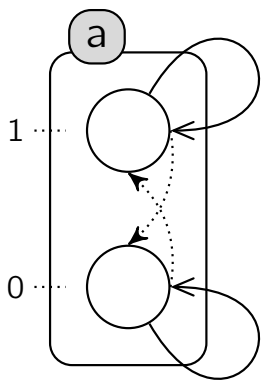
Introduction of Time and Stochastic Features

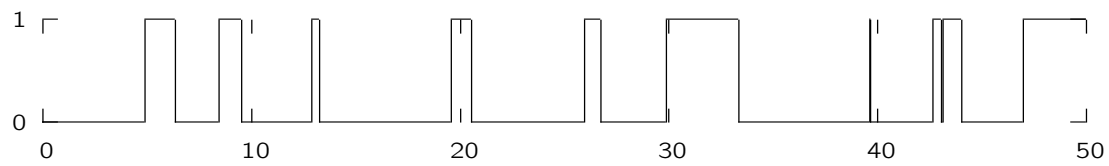
[Paulevé, Magnin, Roux in IEEE TSE, 2010]

Goal: tune time features within a stochastic framework.

Action parameters

- Rate r + stochasticity absorption factor sa.
- Action duration $\approx$ sum of sa exponentials at rate $r.sa$.
- Mean duration: r^{-1}; variance: $r^{-2}sa^{-1}$ (Erlang distribution).





sa=1

(done in the stochastic π-calculus framework)

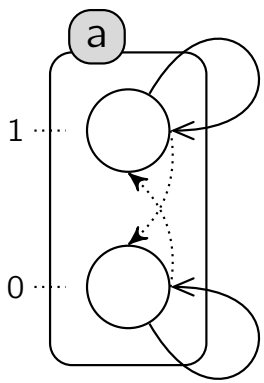
Introduction of Time and Stochastic Features

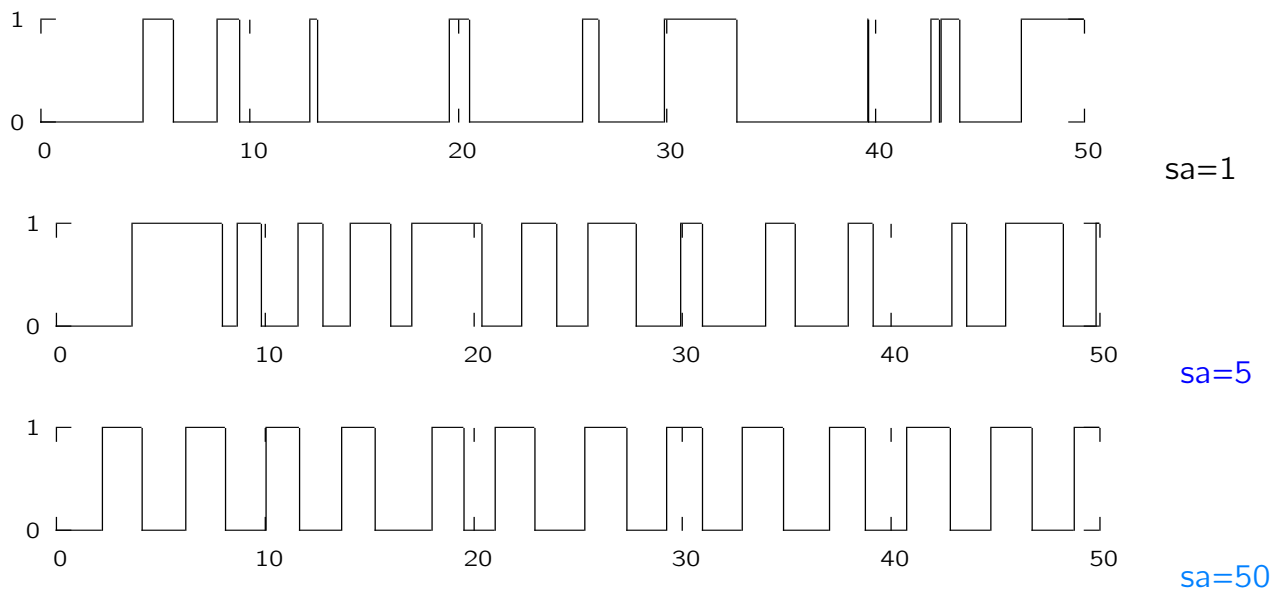
[Paulevé, Magnin, Roux in IEEE TSE, 2010]

Goal: tune time features within a stochastic framework.

Action parameters

- Rate r + stochasticity absorption factor sa.
- Action duration $\approx$ sum of sa exponentials at rate $r.sa$.
- Mean duration: r^{-1}; variance: $r^{-2}sa^{-1}$ (Erlang distribution).



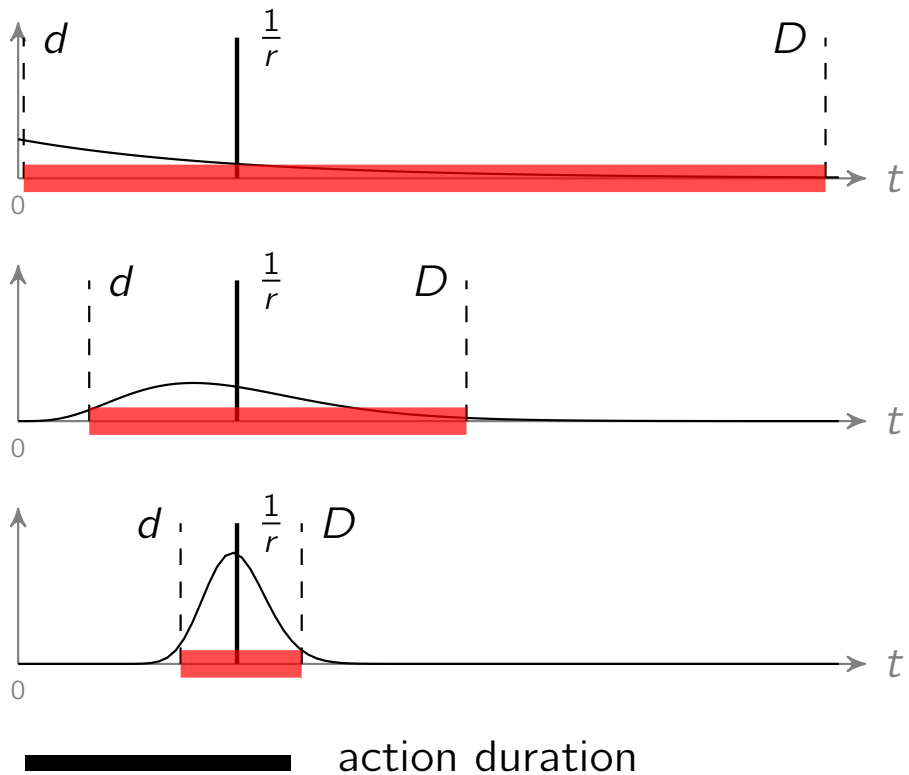


(done in the stochastic π-calculus framework)

Time and Stochastic Parameters

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

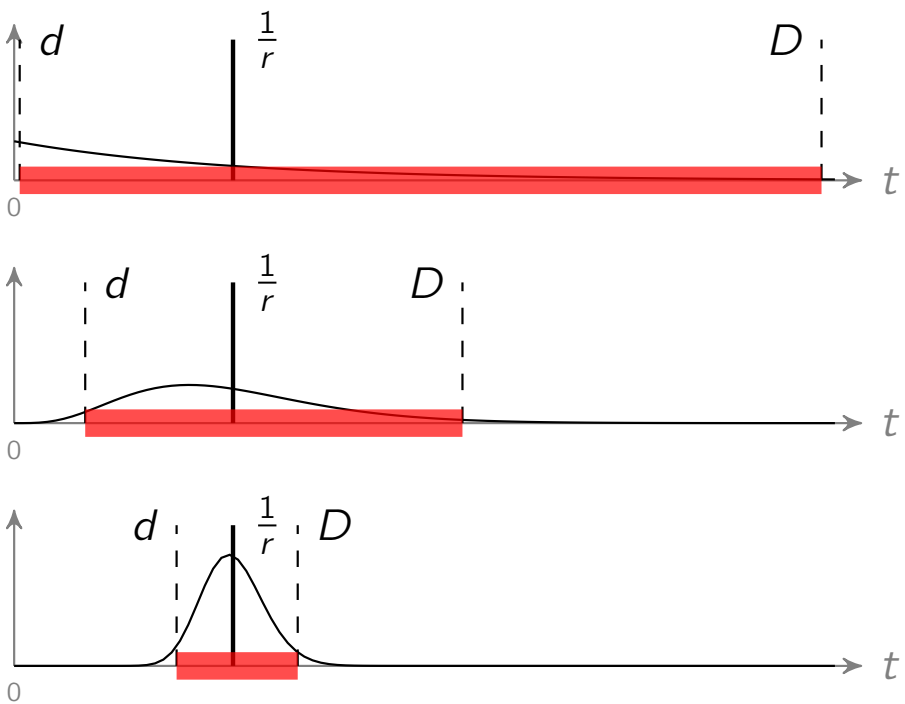
- Parameters: either (r, sa), or the firing interval $[d; D]$.
- Confidence interval of the duration (given a confidence coefficient).
- We have estimators translating $[d; D]$ to (r, sa).

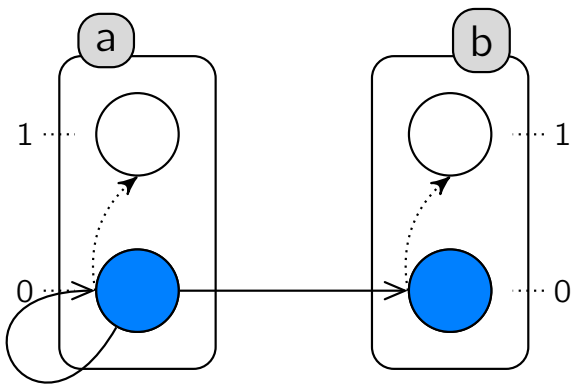


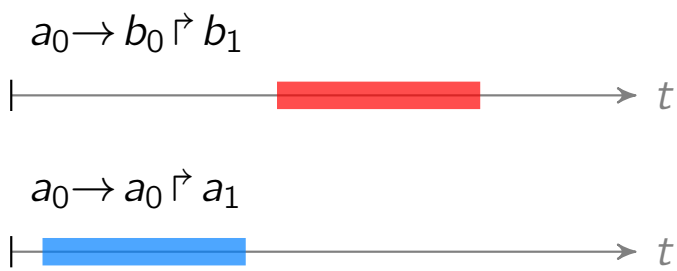
Time and Stochastic Parameters

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

- Parameters: either (r, sa), or the firing interval $[d; D]$.
- Confidence interval of the duration (given a confidence coefficient).
- We have estimators translating $[d; D]$ to (r, sa).







$\Rightarrow b_1$ reached with a very low probability.

Simulation and formal verification

Simulation

- Non-Markovian simulation.
- Generic abstract machine for stochastic process calculi [Paulevé, Youssef, Lakin, Phillips in CMSB 2010]:
- straightforward instantiation with Process Hitting.

Simulation and formal verification

Simulation

- Non-Markovian simulation.
- Generic abstract machine for stochastic process calculi [Paulevé, Youssef, Lakin, Phillips in CMSB 2010]:
- straightforward instantiation with Process Hitting.

Quantitative model checking

- Traduction from the Erlang-distributed π-calcul to PRISM [Paulevé, Magnin, Roux in IEEE TSE, 2010].
- Applies to Process Hitting.
- Combinatorial explosion with large stochasticity absorption factors;
- but there is hope in various methods using reduction/abstractions, etc.

Simulation and formal verification

Simulation

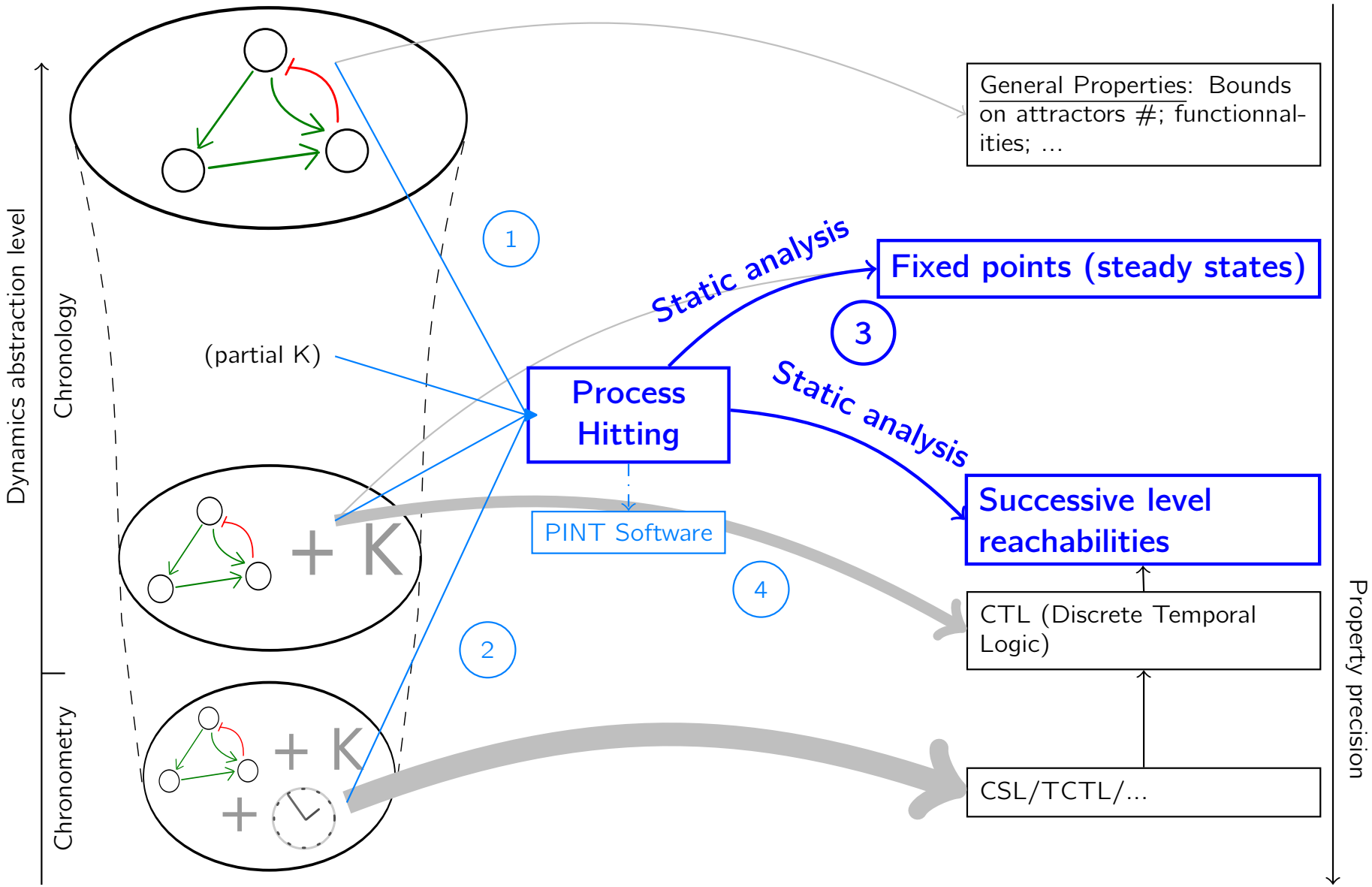
- Non-Markovian simulation.
- Generic abstract machine for stochastic process calculi [Paulevé, Youssef, Lakin, Phillips in CMSB 2010]:
- straightforward instantiation with Process Hitting.

Quantitative model checking

- Traduction from the Erlang-distributed π-calcul to PRISM [Paulevé, Magnin, Roux in IEEE TSE, 2010].
- Applies to Process Hitting.
- Combinatorial explosion with large stochasticity absorption factors;
- but there is hope in various methods using reduction/abstractions, etc.

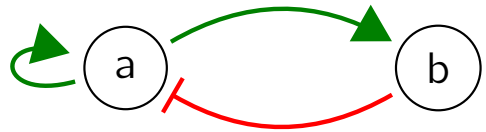
Can we efficiently analyse Process Hittings?

Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
General Properties: Bounds on attractors #; functionnalities; ...
1
Static analysis
Fixed points (steady states)
3
(partial K)
Process Hitting
Static analysis
Successive level reachabilities
PINT Software
4
CTL (Discrete Temporal Logic)
2
+ K
+ K
+
CSL/TCTL/...

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 ⇒ models with simple structures.
- Efficient static derivation of dynamical properties.

Static Analysis of Process Hittings

Intuition

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

Fixed Points

- Reduction to the n-cliques of an n-partite graph.
- Connected result: Topological fixed points within Boolean Networks [Paulevé, Richard in CRAS 2010].

Static Analysis of Process Hittings

Intuition

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

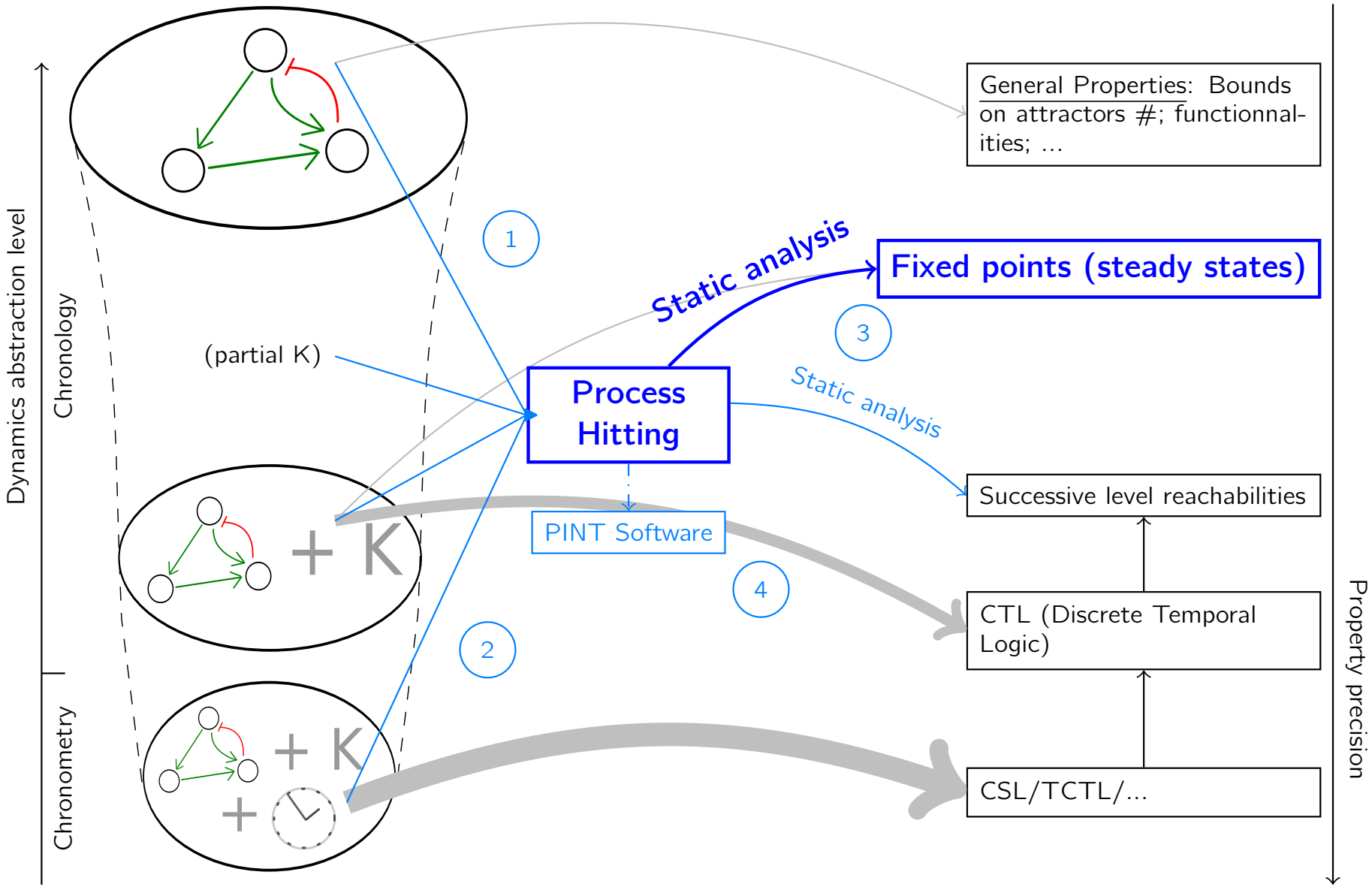
Fixed Points

- Reduction to the n-cliques of an n-partite graph.
- Connected result: Topological fixed points within Boolean Networks [Paulevé, Richard in CRAS 2010].

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).

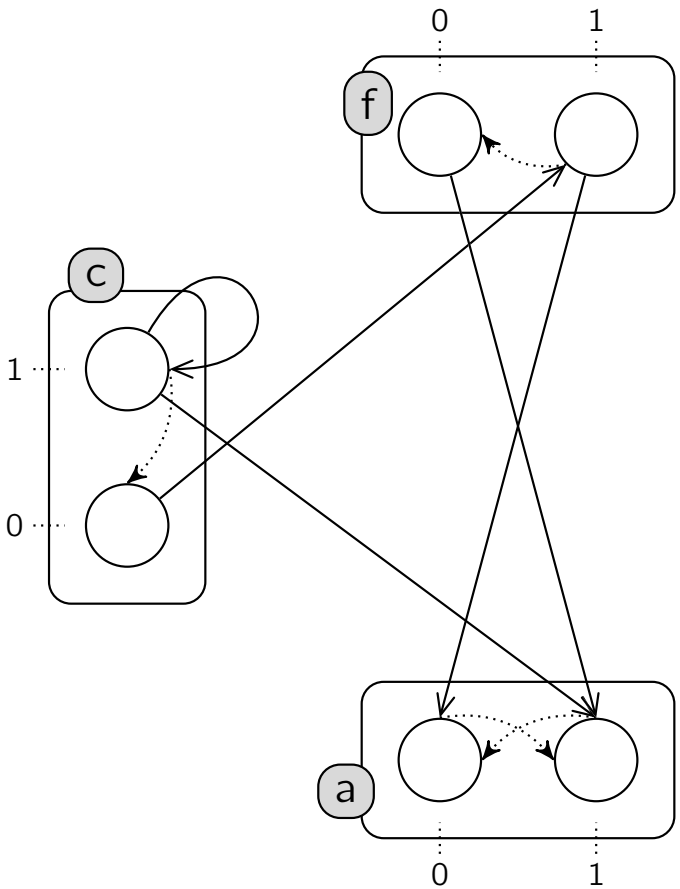
Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
General Properties: Bounds on attractors #; functionnalities; ...
1
Static analysis
Fixed points (steady states)
3
(partial K)
Process Hitting
Static analysis
Successive level reachabilities
PINT Software
4
CTL (Discrete Temporal Logic)
2
+ K
+ K
+
CSL/TCTL/...

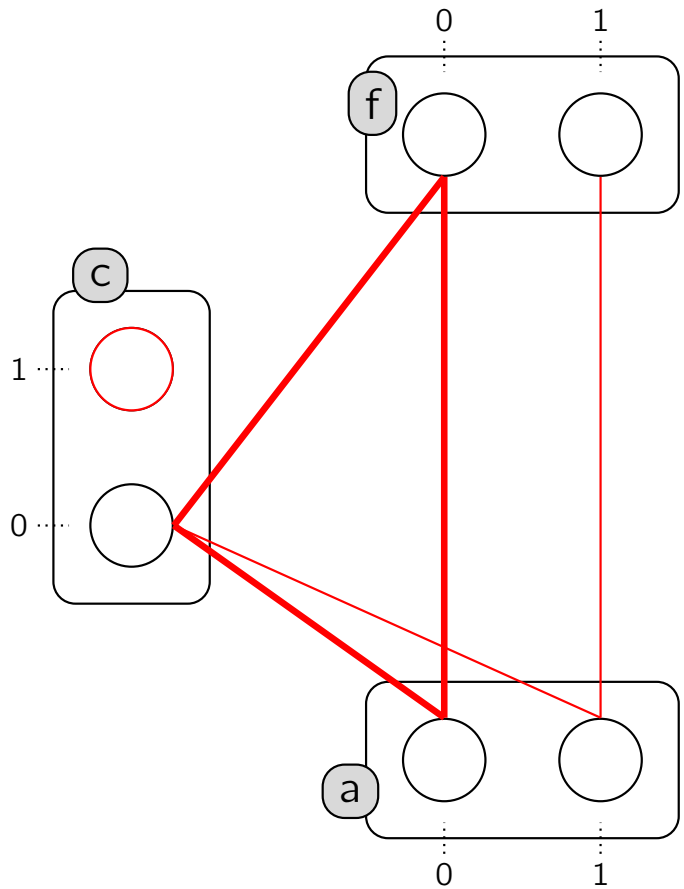
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



Hitless graph

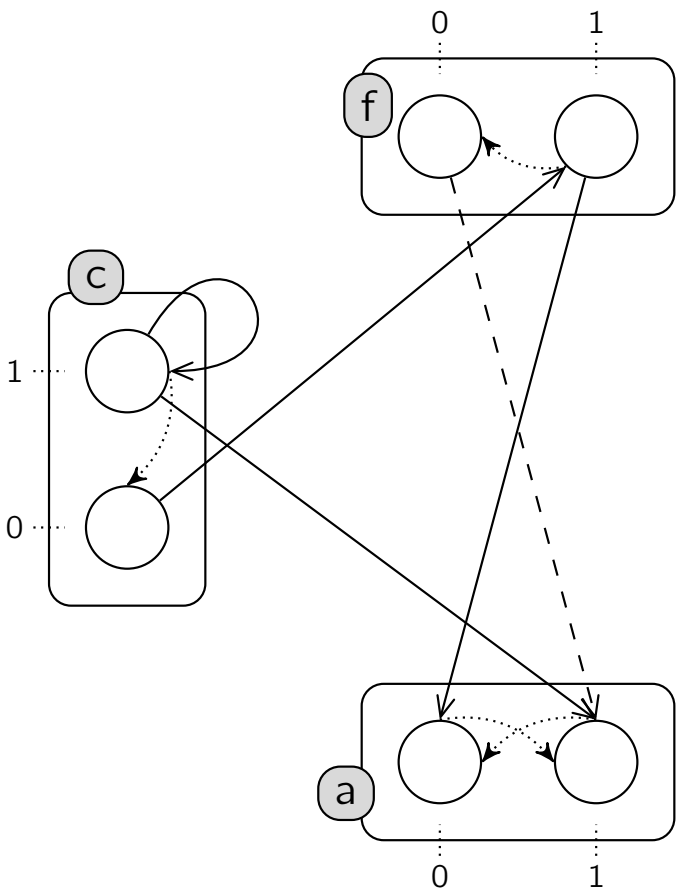


n-cliques are fixed points

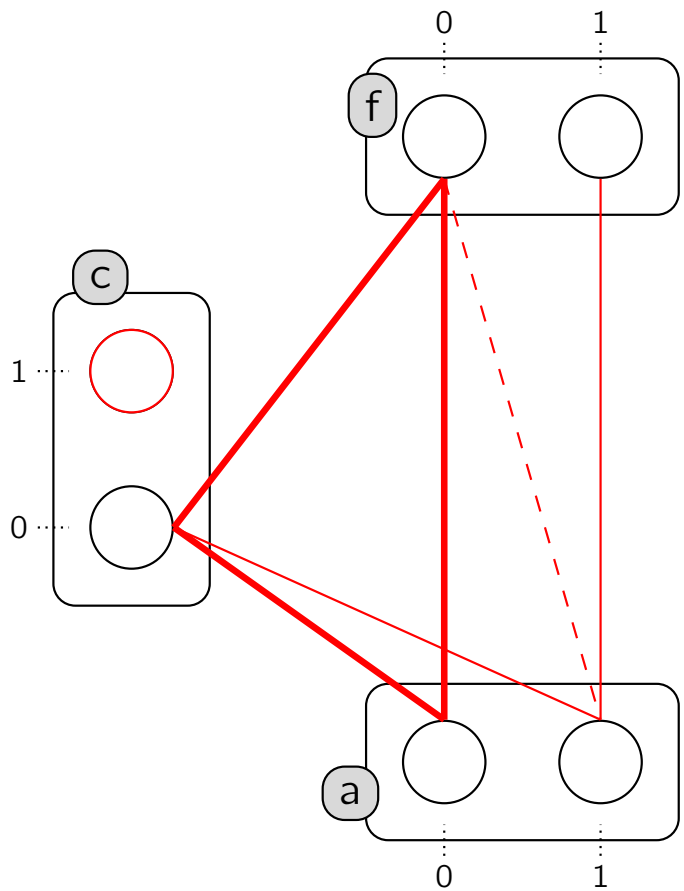
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



Hitless graph

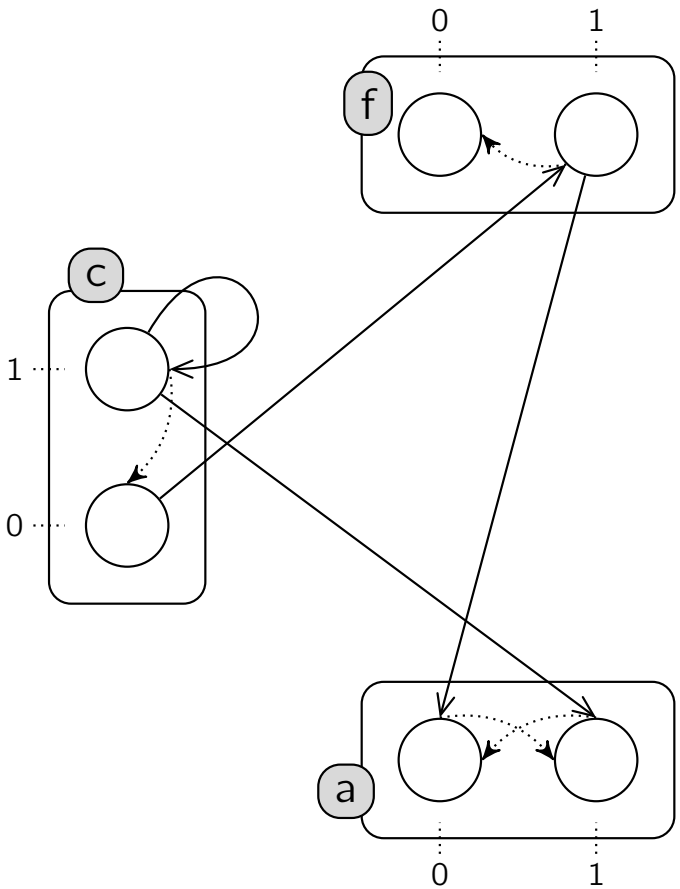


n-cliques are fixed points

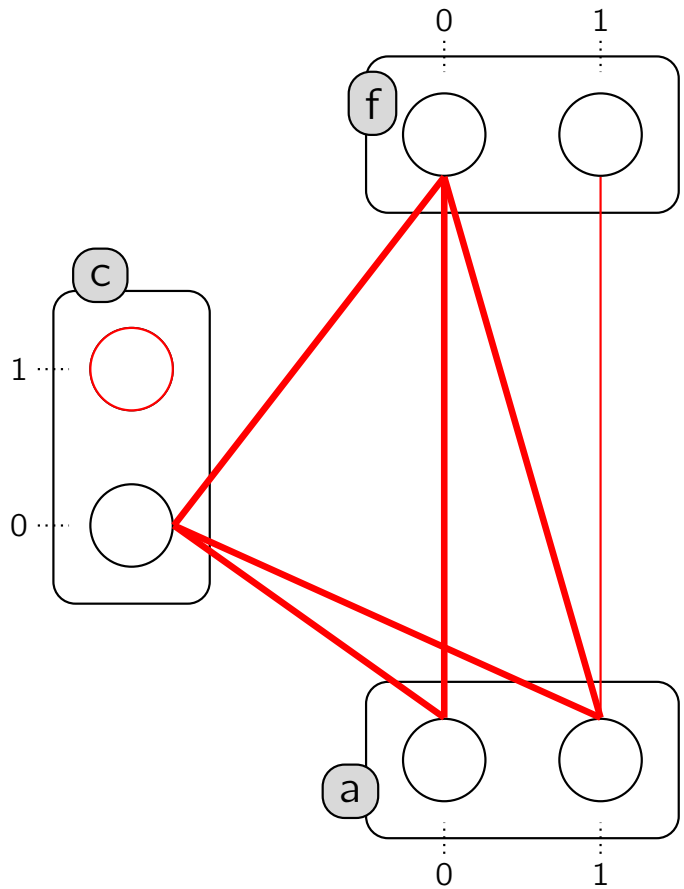
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting

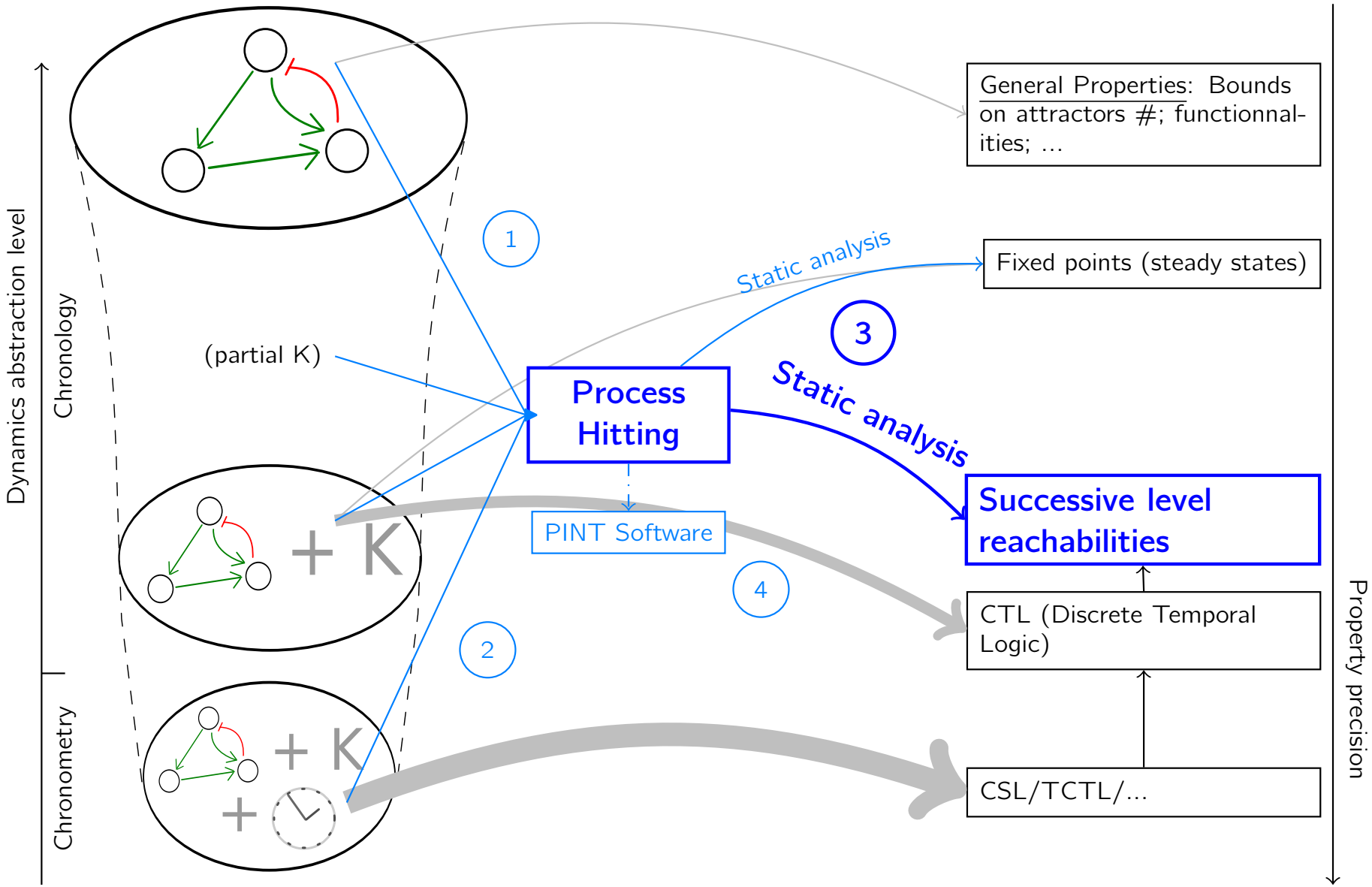


Hitless graph

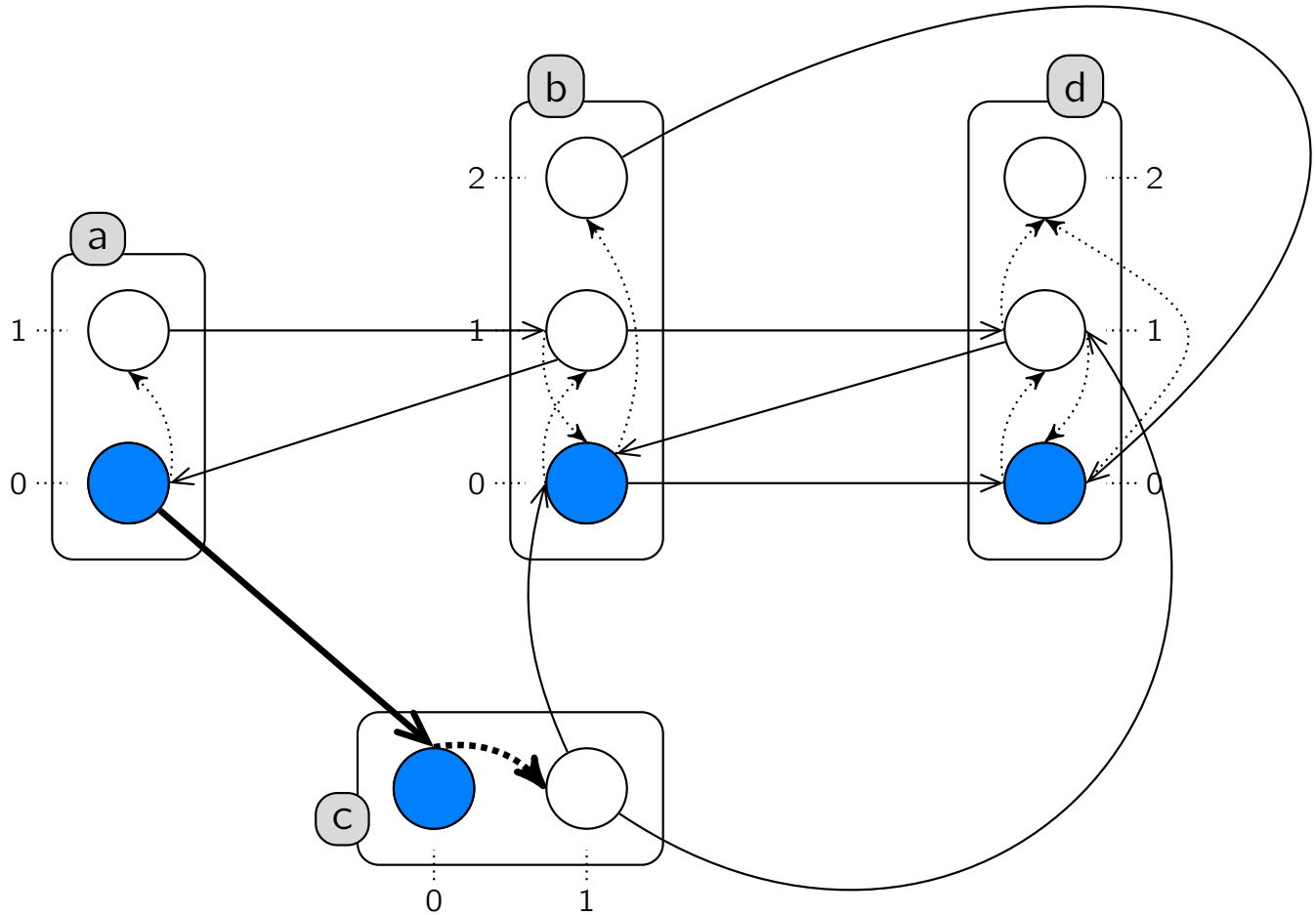


n-cliques are fixed points

Outline

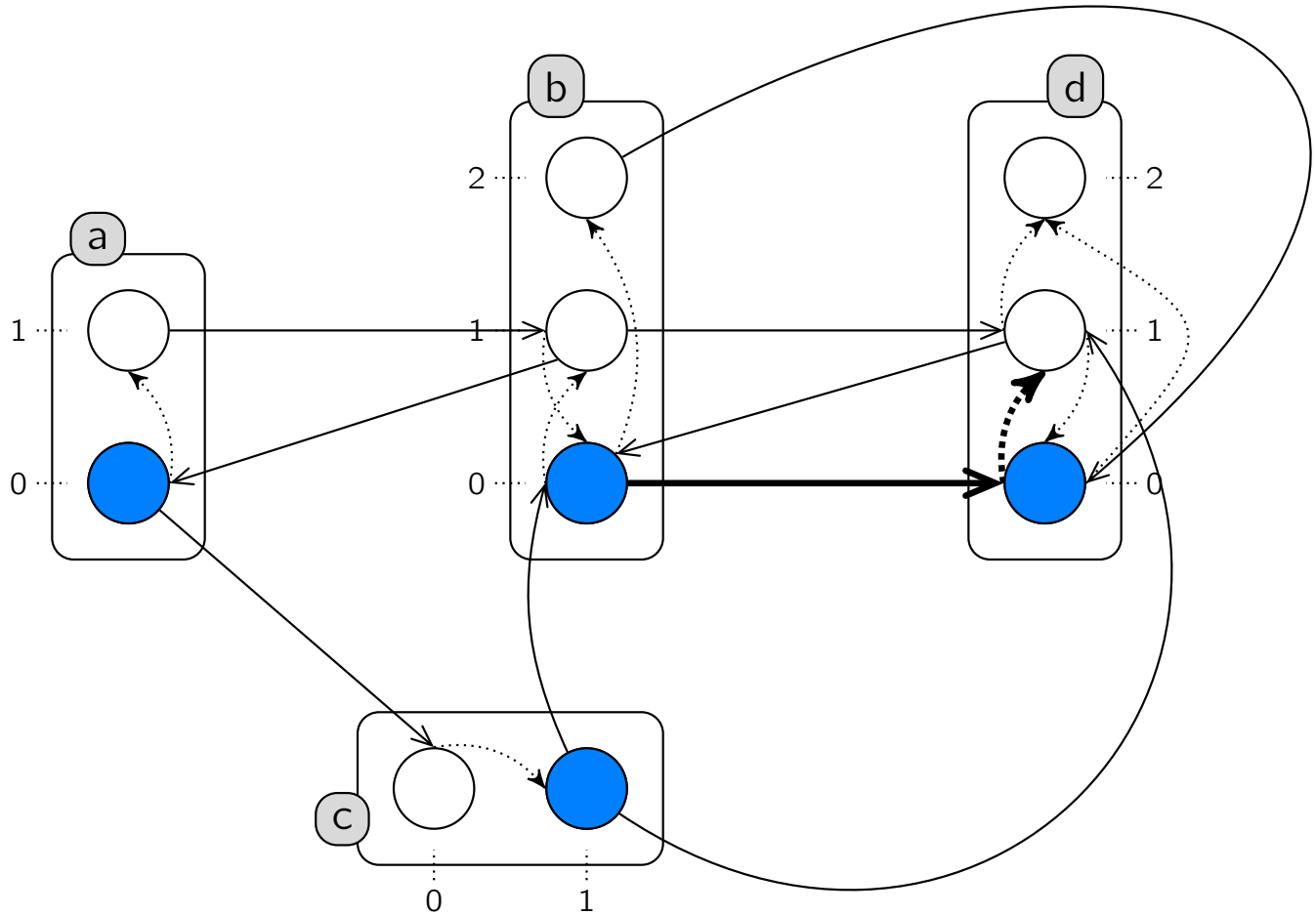

Dynamics abstraction level
Chronology
Chronometry
Property precision
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Static analysis
1
(partial K)
3
Process Hitting
Static analysis
Successive level reachabilities
PINT Software
4
CTL (Discrete Temporal Logic)
2
+ K
+ K
+
CSL/TCTL/...

Scenarios



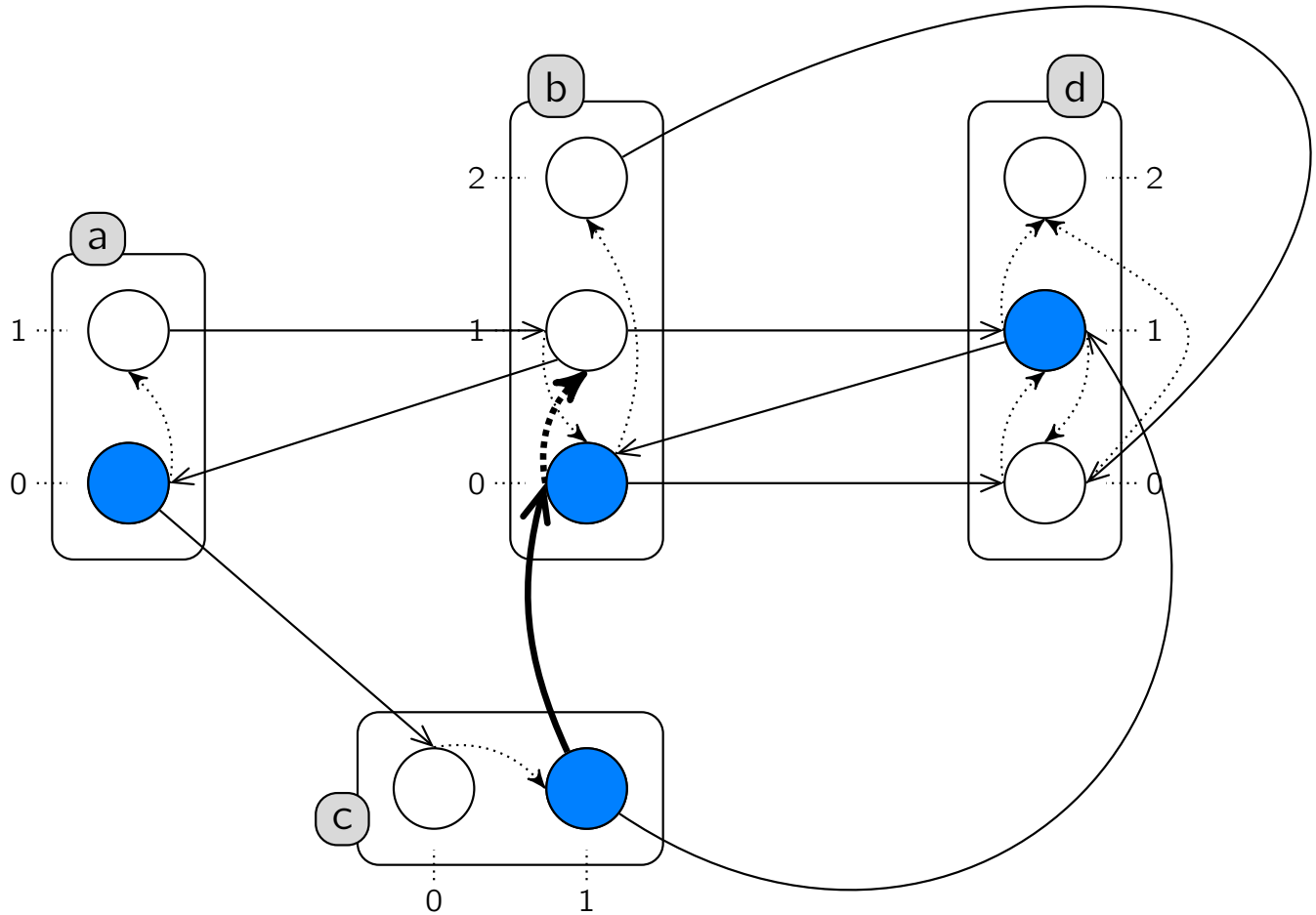
$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$

Scenarios



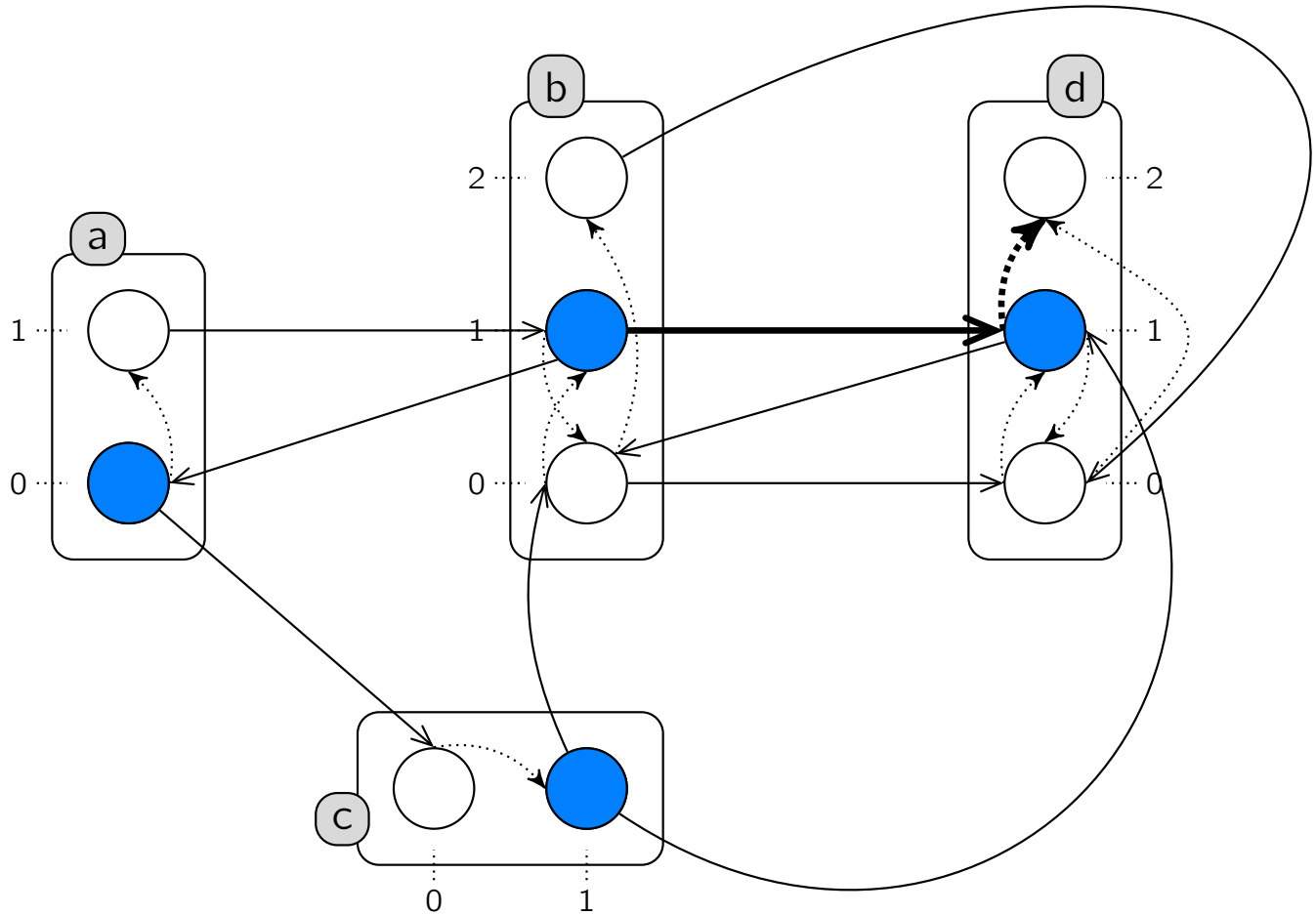
$$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$$

Scenarios



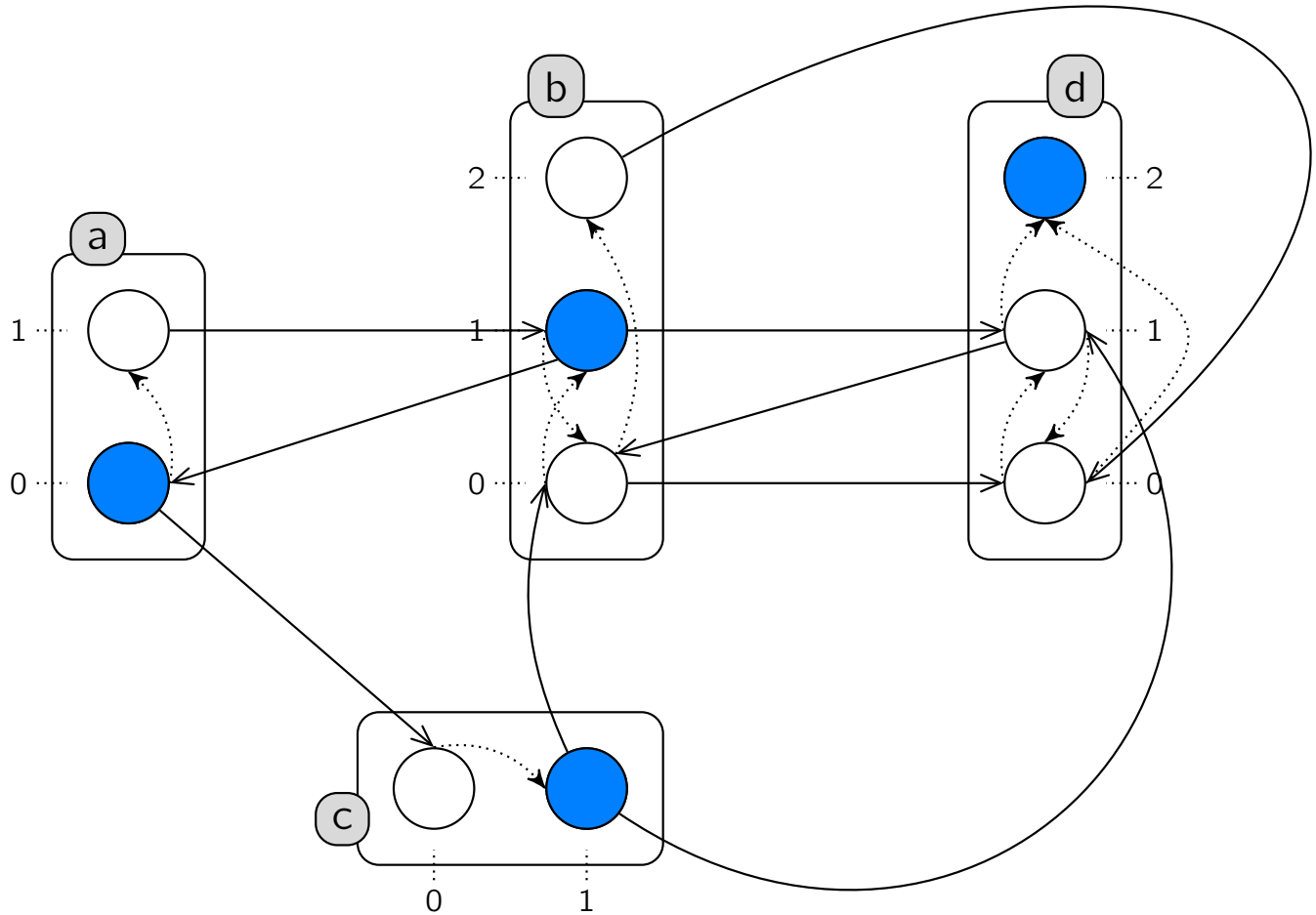
$$a_0 \longrightarrow c_0 \,\lceil\!\!\nearrow\, c_1 :: b_0 \longrightarrow d_0 \,\lceil\!\!\nearrow\, d_1 :: c_1 \longrightarrow b_0 \,\lceil\!\!\nearrow\, b_1 :: b_1 \longrightarrow d_1 \,\lceil\!\!\nearrow\, d_2$$

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



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

Static Analysis of Successive Reachability Properties

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

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

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 $\mathcal{P} \Longrightarrow \mathcal{R}$ is impossible.

Under-approximations

$\mathcal{PH}$ satisfies $\mathcal{Q} \Longrightarrow \mathcal{R}$ is possible.

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

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.

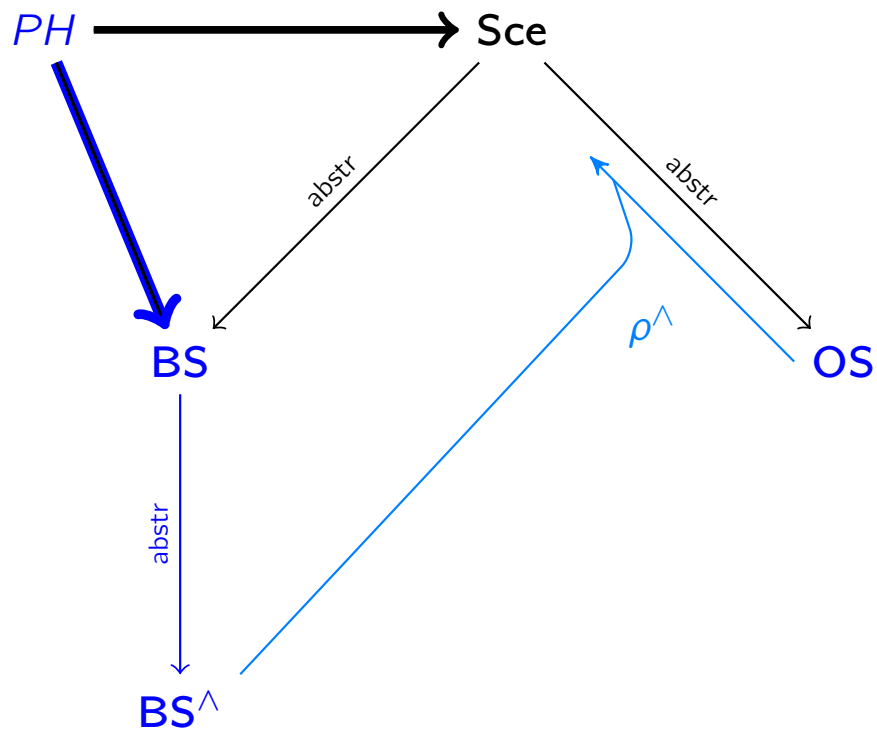
- E.g. $\delta = a_0 \rightarrow c_0 \nearrow c_1 :: b_0 \rightarrow d_0 \nearrow d_1 :: c_1 \rightarrow b_0 \nearrow b_1 :: b_1 \rightarrow d_1 \nearrow d_2$.

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 orthogonal 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 Orthogonal 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

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

Two Orthogonal Abstractions

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

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 Orthogonal Abstractions

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

Abstraction by Objective Sequences

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

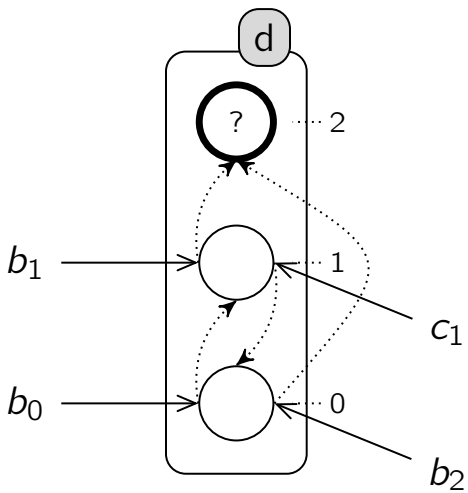
Two Orthogonal Abstractions

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

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

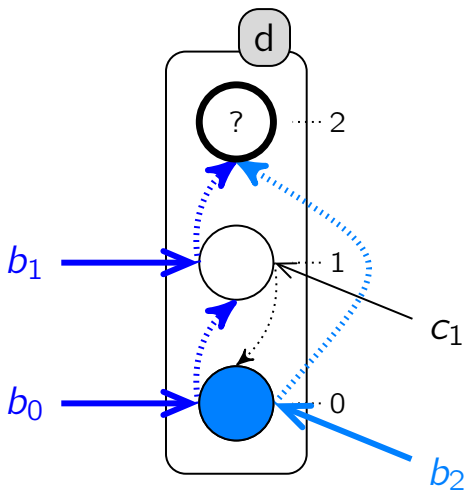
Two Orthogonal Abstractions

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

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:

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

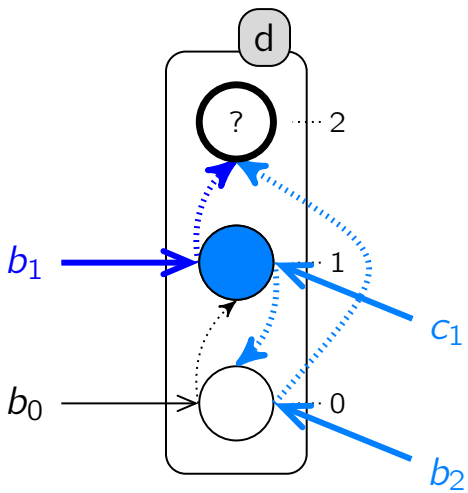
Two Orthogonal Abstractions

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

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:

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

Objective Sequence Refinements

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

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

Objective Sequence Refinements

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

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

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

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

Objective Sequence Refinements

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

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

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

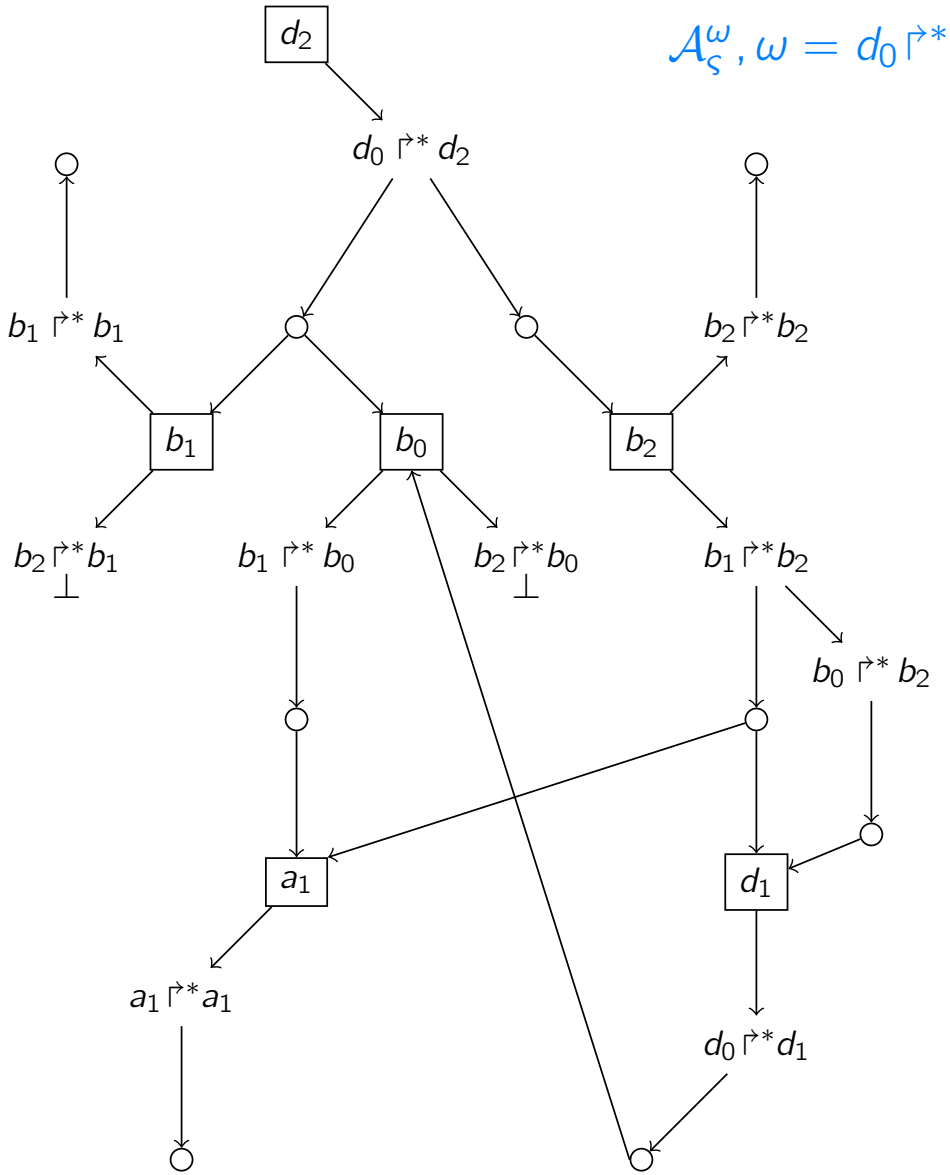
$\mathbf{Obj} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$d_0 \Rsh^* d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \Rsh^* b_0 :: b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_1 :: b_1 \Rsh^* b_0 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_2 :: d_0 \Rsh^* d_2$
$\gamma_\varsigma(d_0 \Rsh^* d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0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

Generalization to $\mathbf{OS}$ 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 $\omega' :: \omega_n \in \rho^\wedge(\omega_n, \mathbf{BS}^\wedge(\omega_n))$
$\gamma_\varsigma(\omega)$	$= \gamma_\varsigma(\widetilde{\rho}(\omega, \mathbf{BS}^\wedge))$

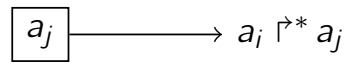
Abstract Structure of Process Hitting

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



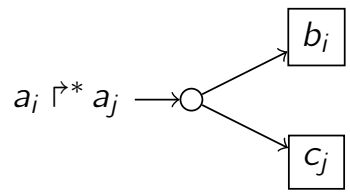
Legend

Requirement

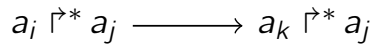


Solution

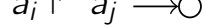
$(\{b_i, c_j\} \in \mathbf{BS}^{\wedge}(a_i \Rsh^* a_j))$



Continuity



Trivial solution



No solution

$a_i \Rsh^* a_j$
$\bot$

Approximations of Successive Reachability

Over-approximations

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

No / Inconc

Successive Reachability

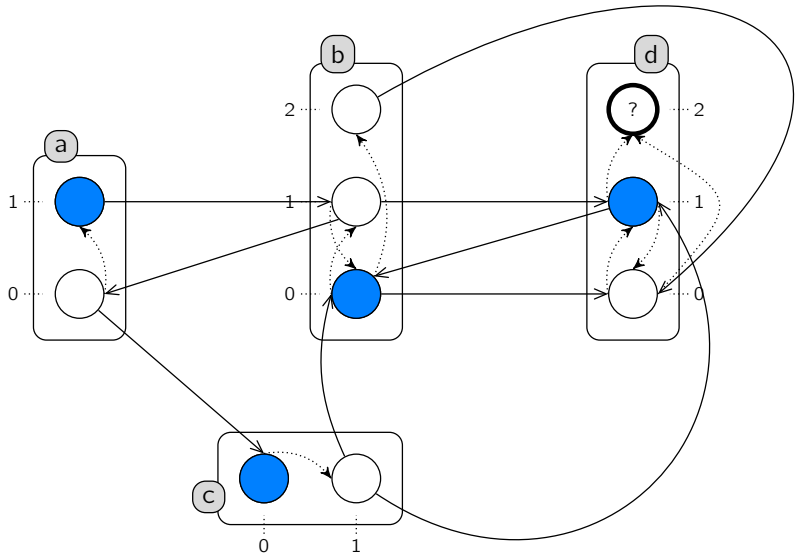
Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

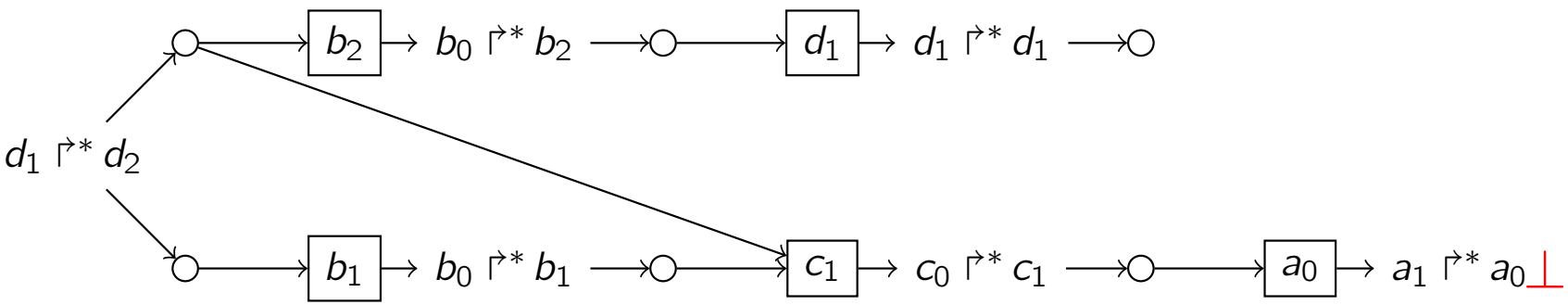
Un-ordered Over-approximation

Example



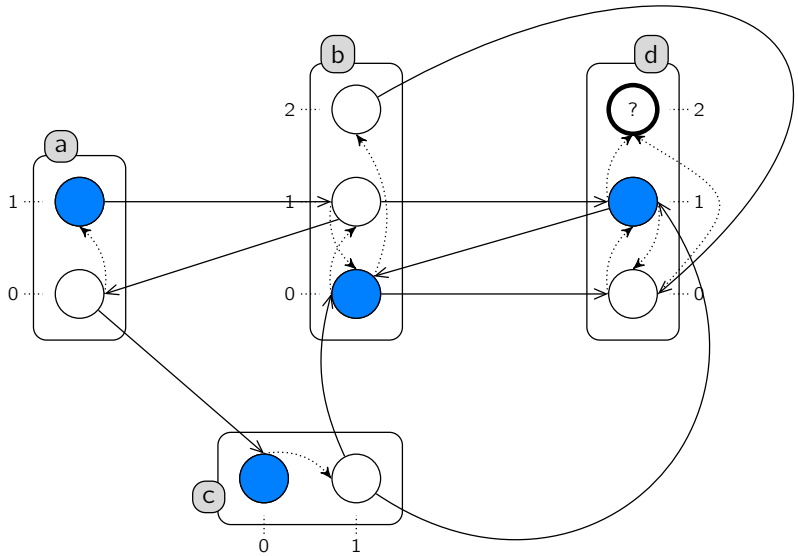
Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}_\varsigma^\omega$ such that:

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



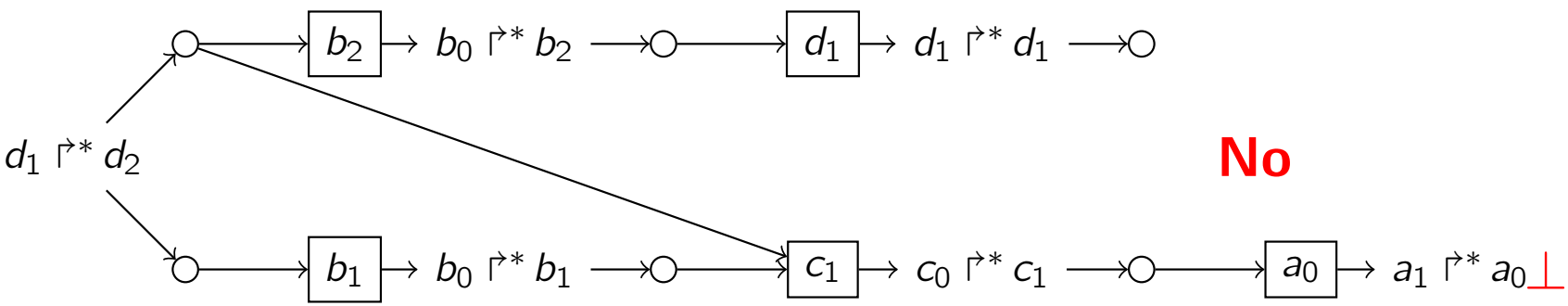
Un-ordered Over-approximation

Example



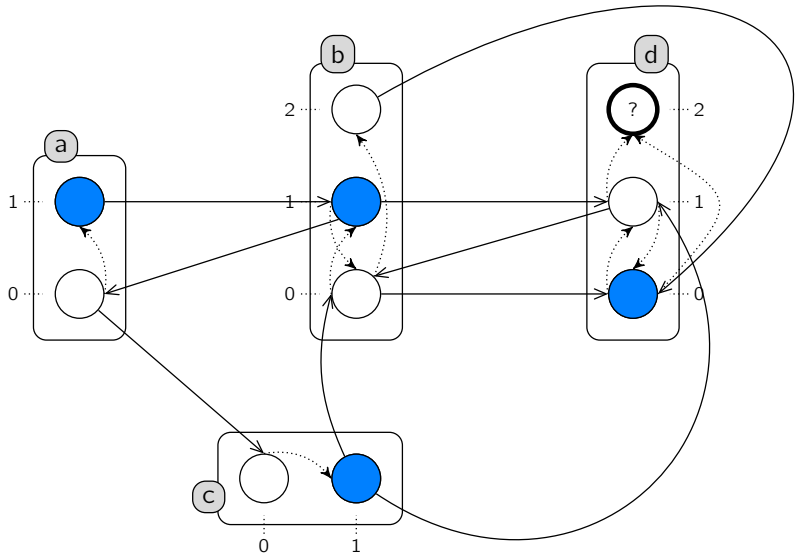
Necessary condition for $\gamma_\varsigma(\omega) \neq \emptyset$:
From each objective within ω, there exists a traversal of $\mathcal{A}_\varsigma^\omega$ such that:

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



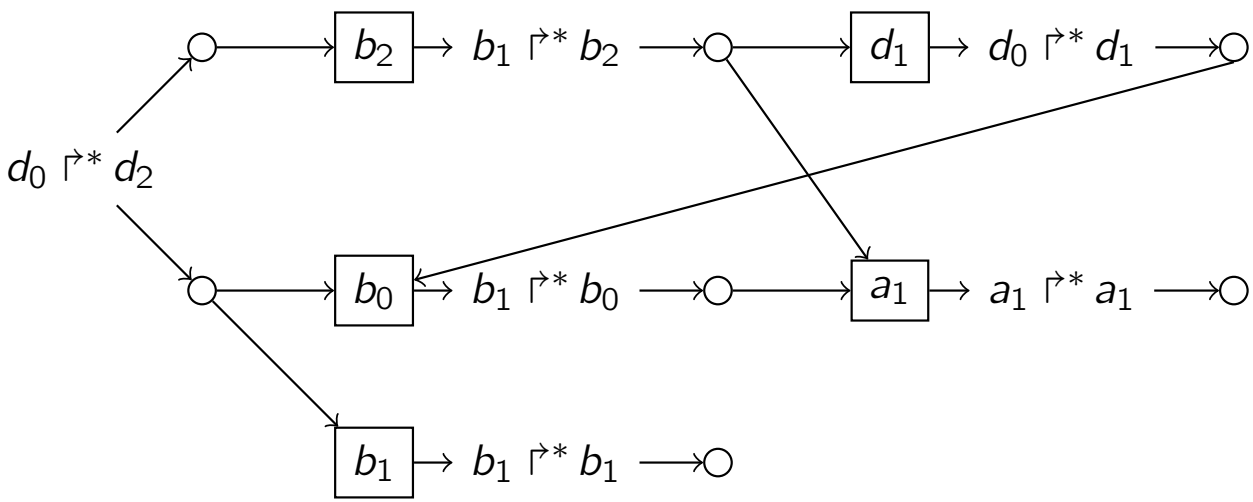
Un-ordered Over-approximation

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:

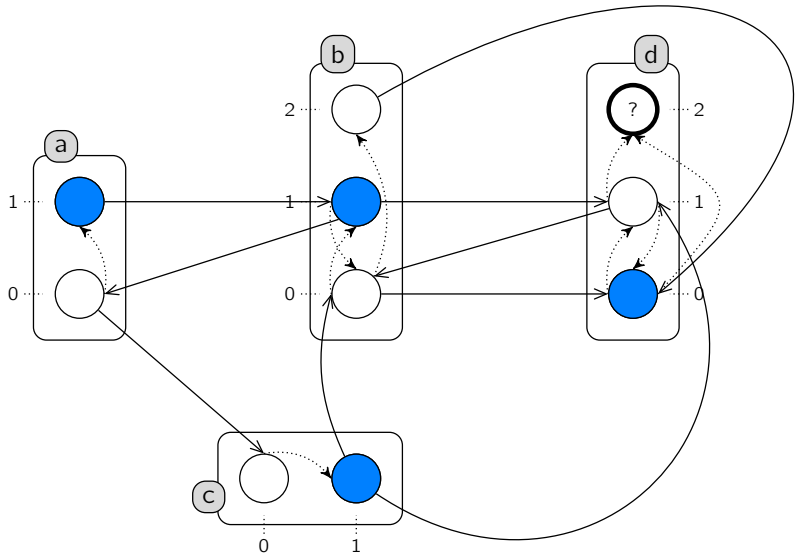
- objective $\rightarrow$ follow at least one solution;
- process $\rightarrow$ follow all objectives;
- no cycle.



Inconc

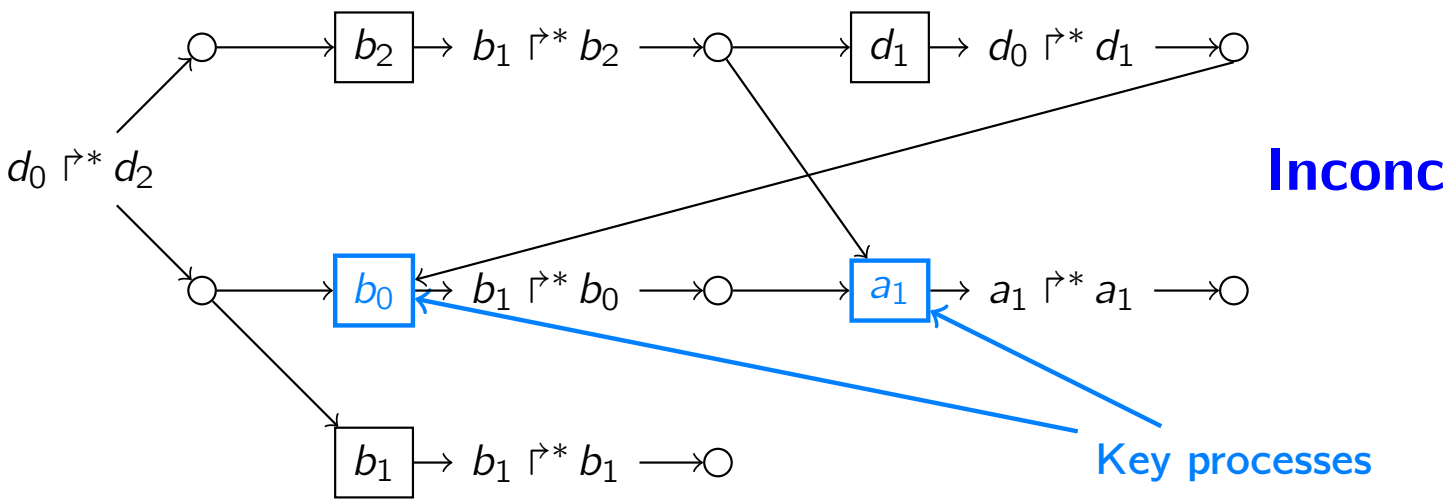
Un-ordered Over-approximation

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 → follow at least one solution;
- process → follow all objectives;
- no cycle.



Approximations of Successive Reachability

Over-approximations

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

No / Inconc

Successive Reachability

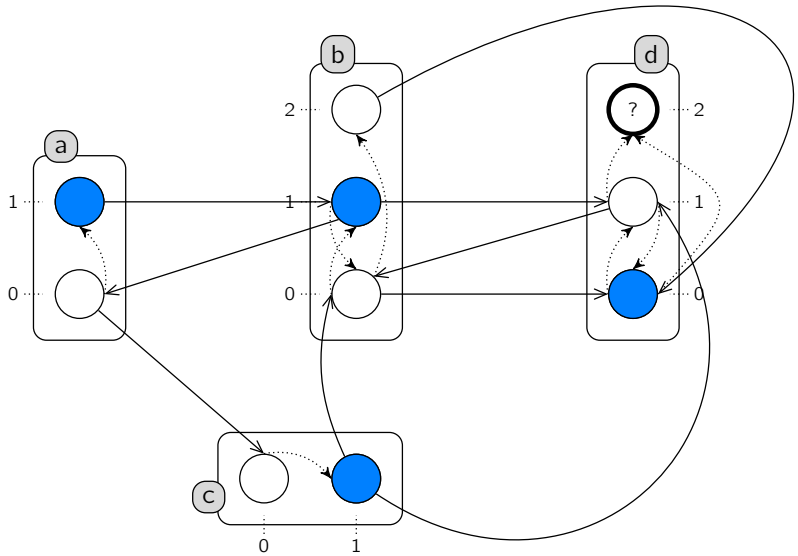
Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

Un-ordered Under-approximation

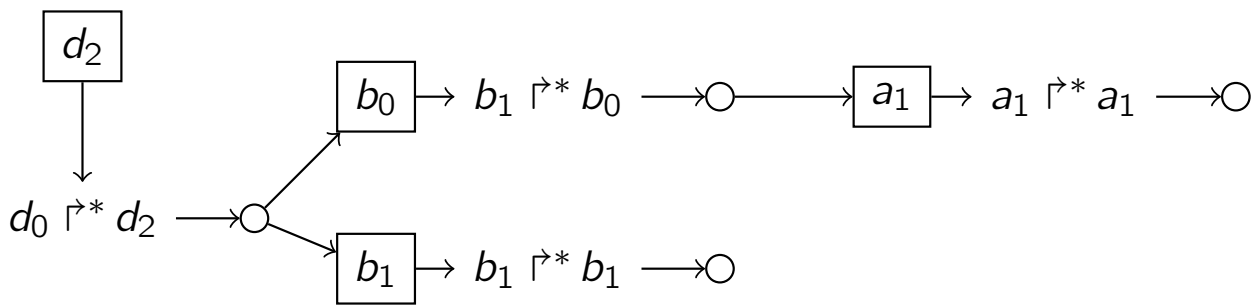
Example



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

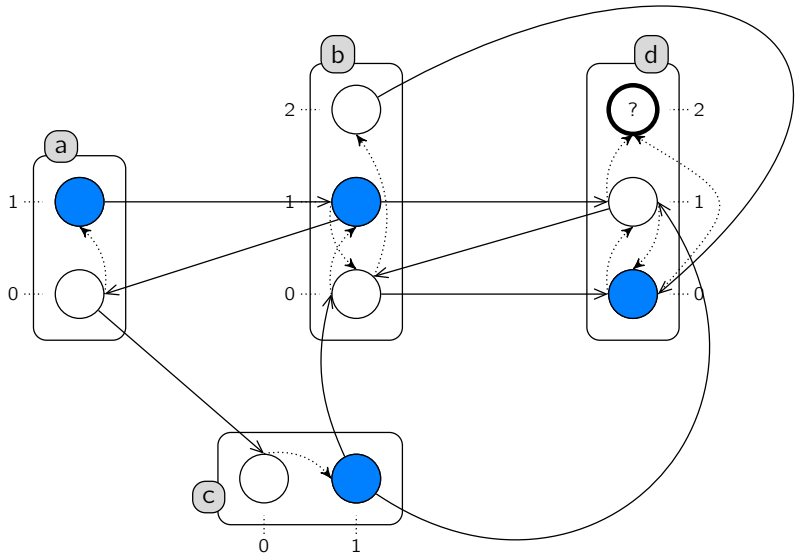
- $\lceil \mathcal{B}^\omega_\varsigma \rceil$ has no cycle;
- each objective has at least one solution.

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



Un-ordered Under-approximation

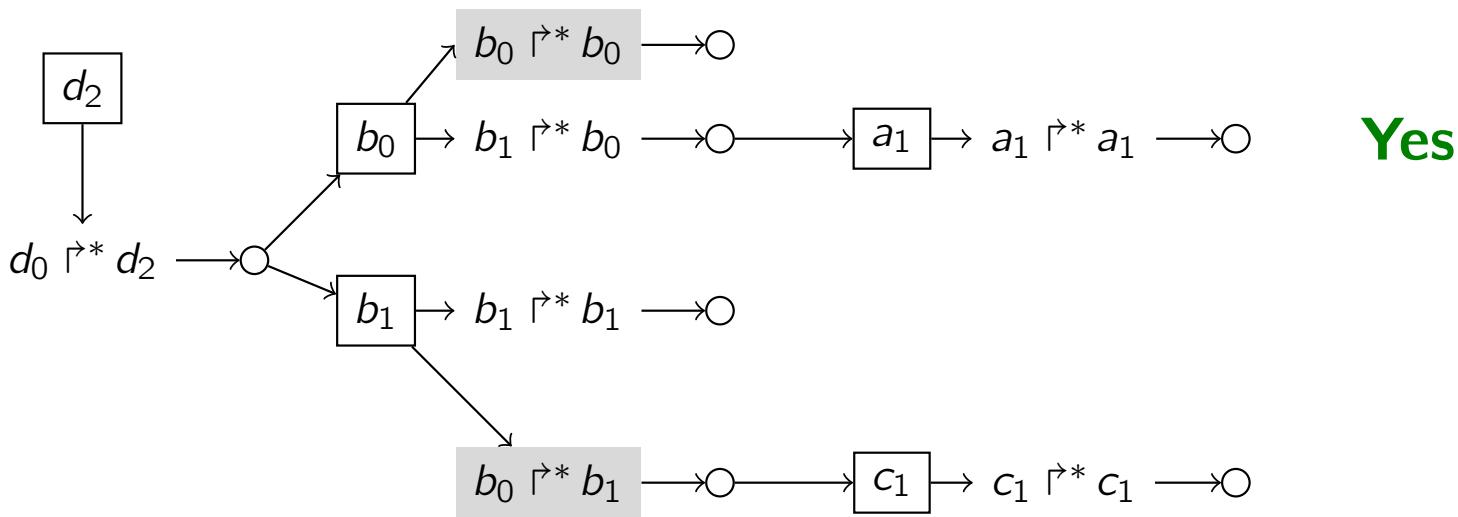
Example



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

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

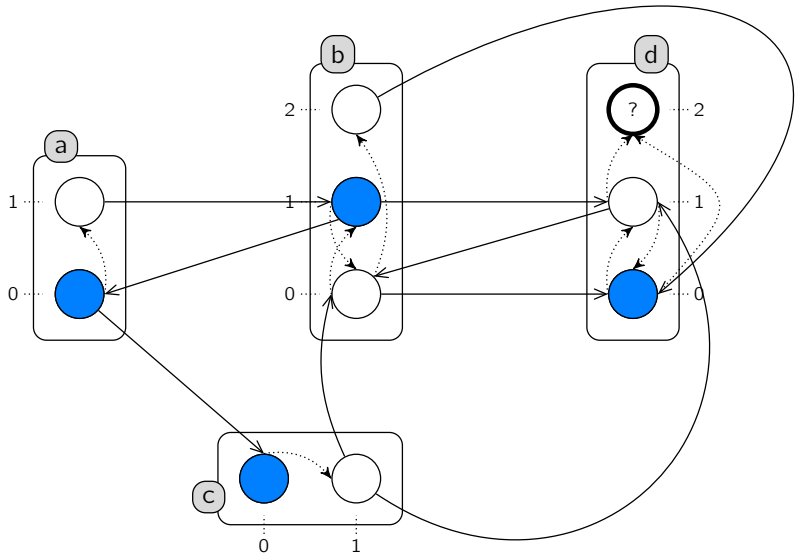
$\lceil \mathcal{B}^\omega_\varsigma \rceil$: saturated $\mathcal{A}^\omega_\varsigma$.



Yes

Un-ordered Under-approximation

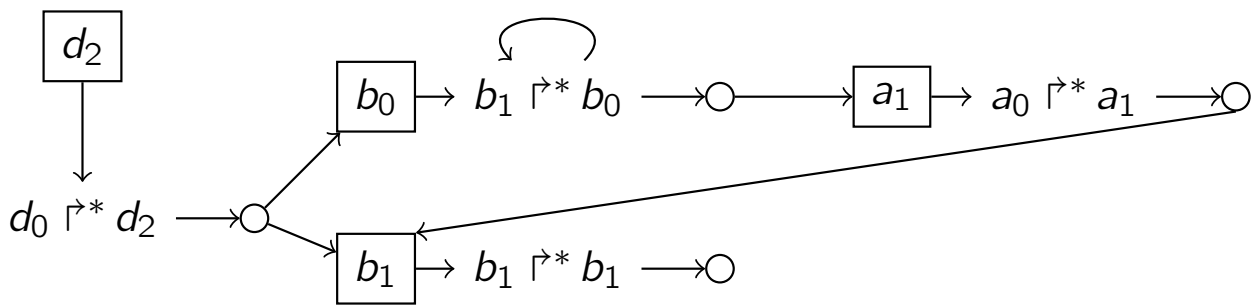
Example



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

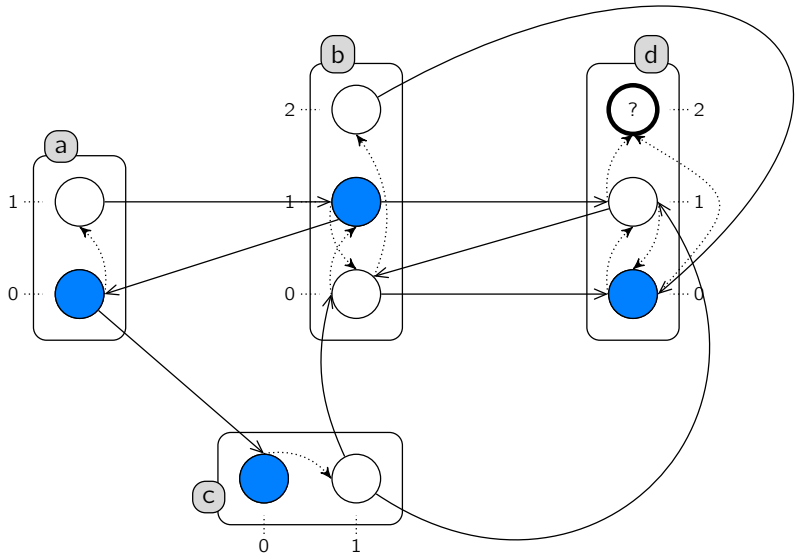
- $\lceil \mathcal{B}^\omega_\varsigma \rceil$ has no cycle;
- each objective has at least one solution.

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



Un-ordered Under-approximation

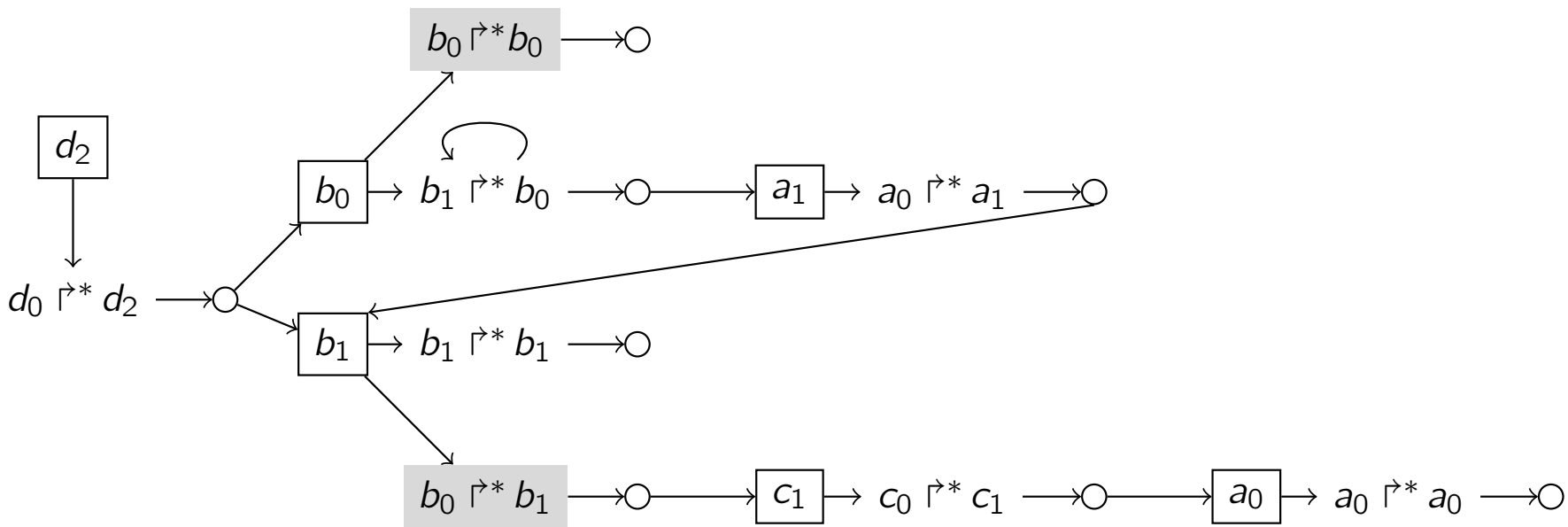
Example



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

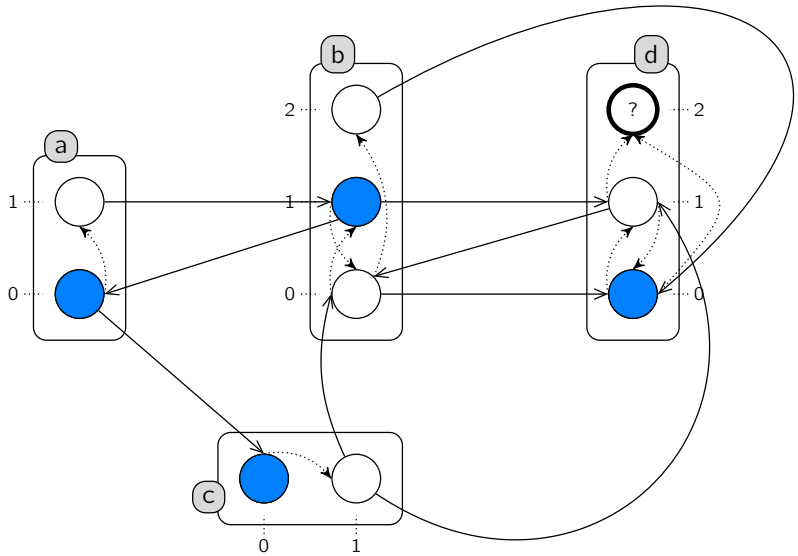
- $\lceil \mathcal{B}^\omega_\varsigma \rceil$ has no cycle;
- each objective has at least one solution.

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



Un-ordered Under-approximation

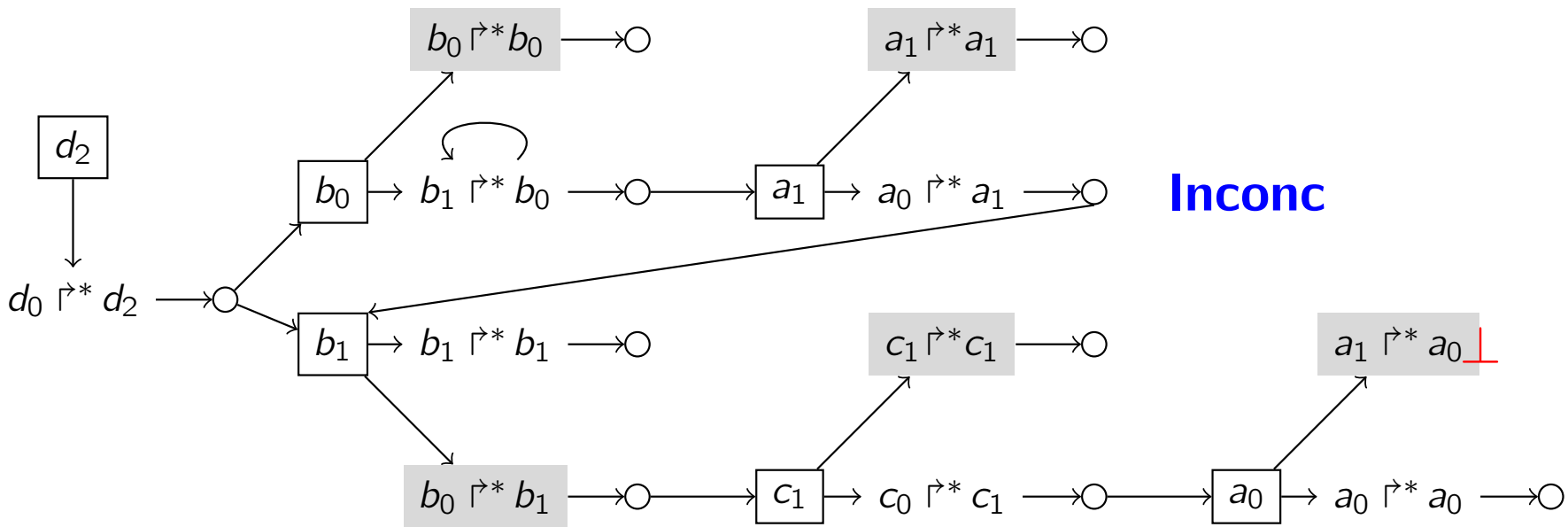
Example



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 $\mathcal{A}^\omega_\varsigma$.

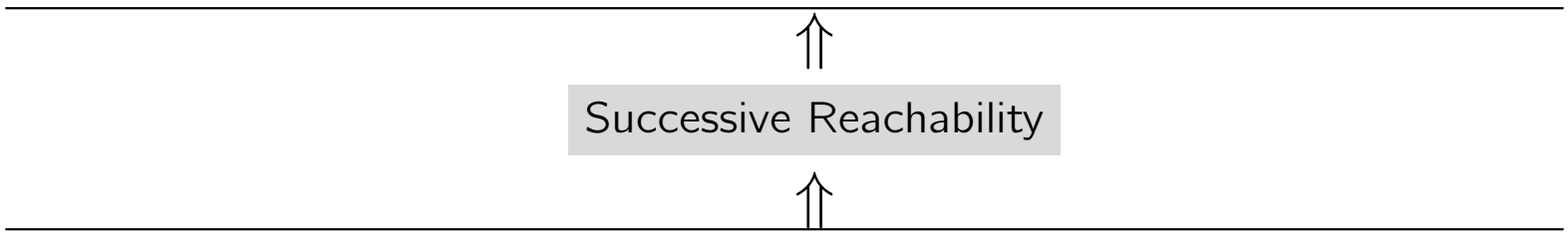


Approximations of Successive Reachability

Over-approximations

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

No / Inconc



Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

Ordered Over-approximation

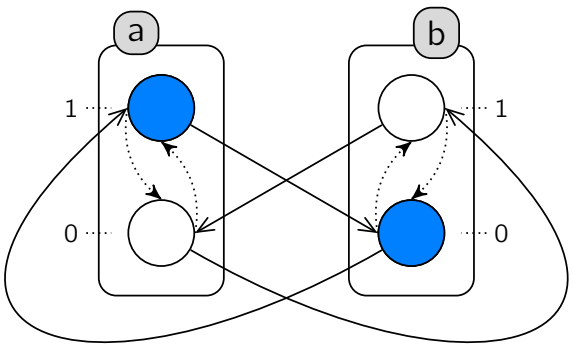
Goal: $\gamma_\varsigma(a_1 \Rsh^* a_0 :: \omega) \neq \emptyset \Longrightarrow \gamma_{max\varsigma}(\omega) \neq \emptyset$

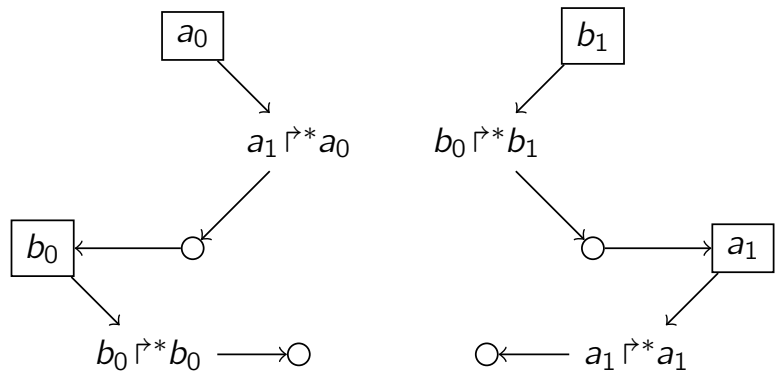
- By default, use the saturated context of $\lceil \mathcal{A}^\omega_\varsigma \rceil$:
$$\lceil \mathcal{A}^\omega_\varsigma \rceil = \mathrm{lfp}\ \left(\mathcal{A}^\omega_\varsigma \mapsto \mathcal{A}^\omega_{\varsigma \Cap \mathbf{procs}(\mathcal{A}^\omega_\varsigma)} \right) .$$
- $a_0 \notin \varsigma$,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



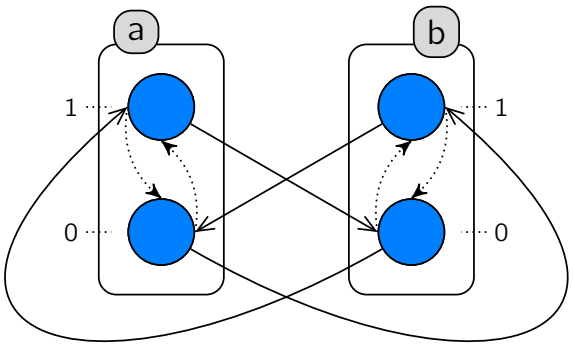


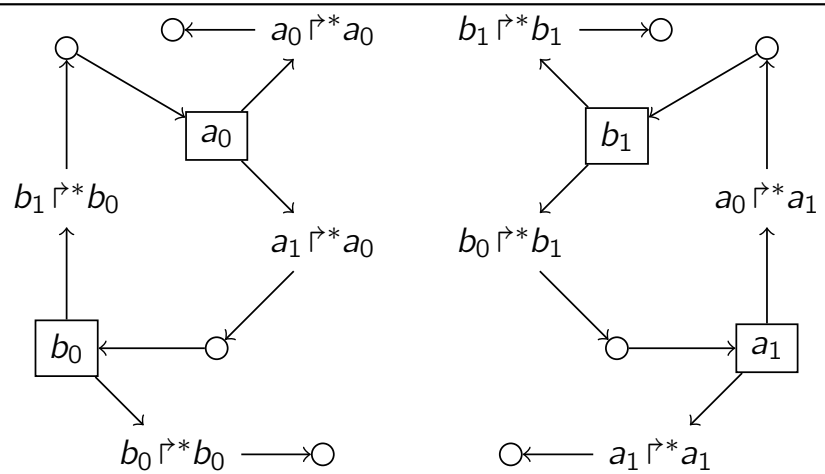
$\gamma_{\langle a_1, b_0 \rangle}(a_1 \Rsh^* a_0 :: b_0 \Rsh^* b_1) \neq \emptyset$

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



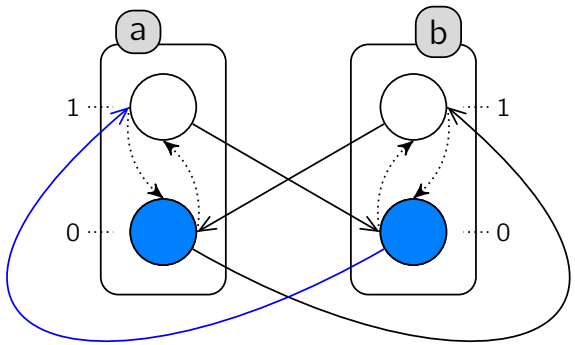


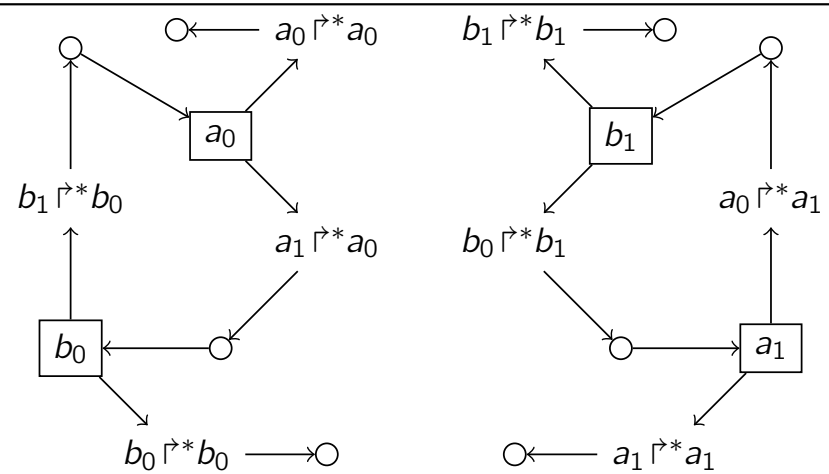
$\gamma_{\langle a_1, b_0 \rangle}(a_1 \Rsh^* a_0 :: b_0 \Rsh^* b_1) \neq \emptyset$

Ordered Over-approximation

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

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



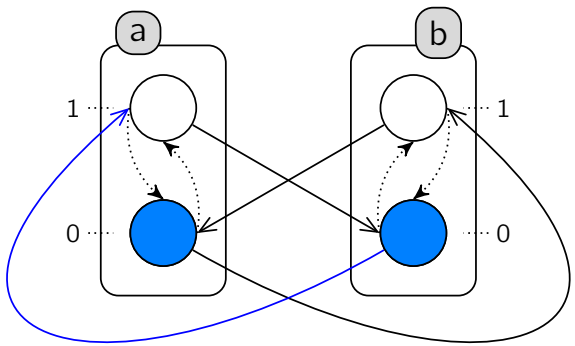


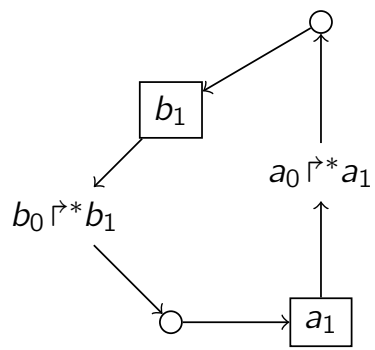
$\gamma_{\langle a_1, b_0 \rangle}(a_1 \restriction^* a_0 :: b_0 \restriction^* b_1) \neq \emptyset \implies \gamma_{\langle a_0, b_0 \rangle}(b_0 \restriction^* b_1) \neq \emptyset$

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





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

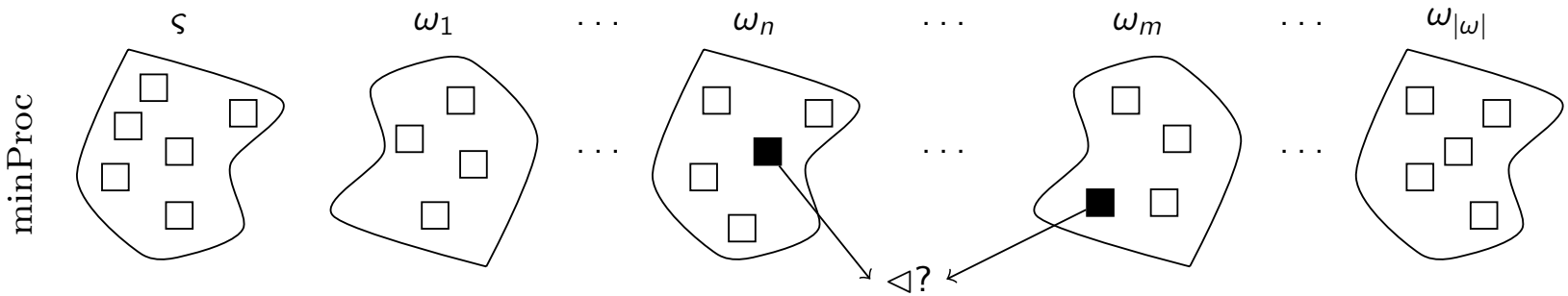
Process Occurrences Order Constraints

$a_j \lhd a_i \iff$ no scenario can be abstracted by $a_i \not\curvearrowright^* a_j$.

Uncovering Order Constraints

$\mathbf{BS}(a_i \not\curvearrowright^* a_j) = \emptyset \implies a_j \lhd a_i$

Idea of Over-Approximation



Based on the ordered over-approximation:

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

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

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
- $\Rightarrow$ drive refinements of ω.

Complexity

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

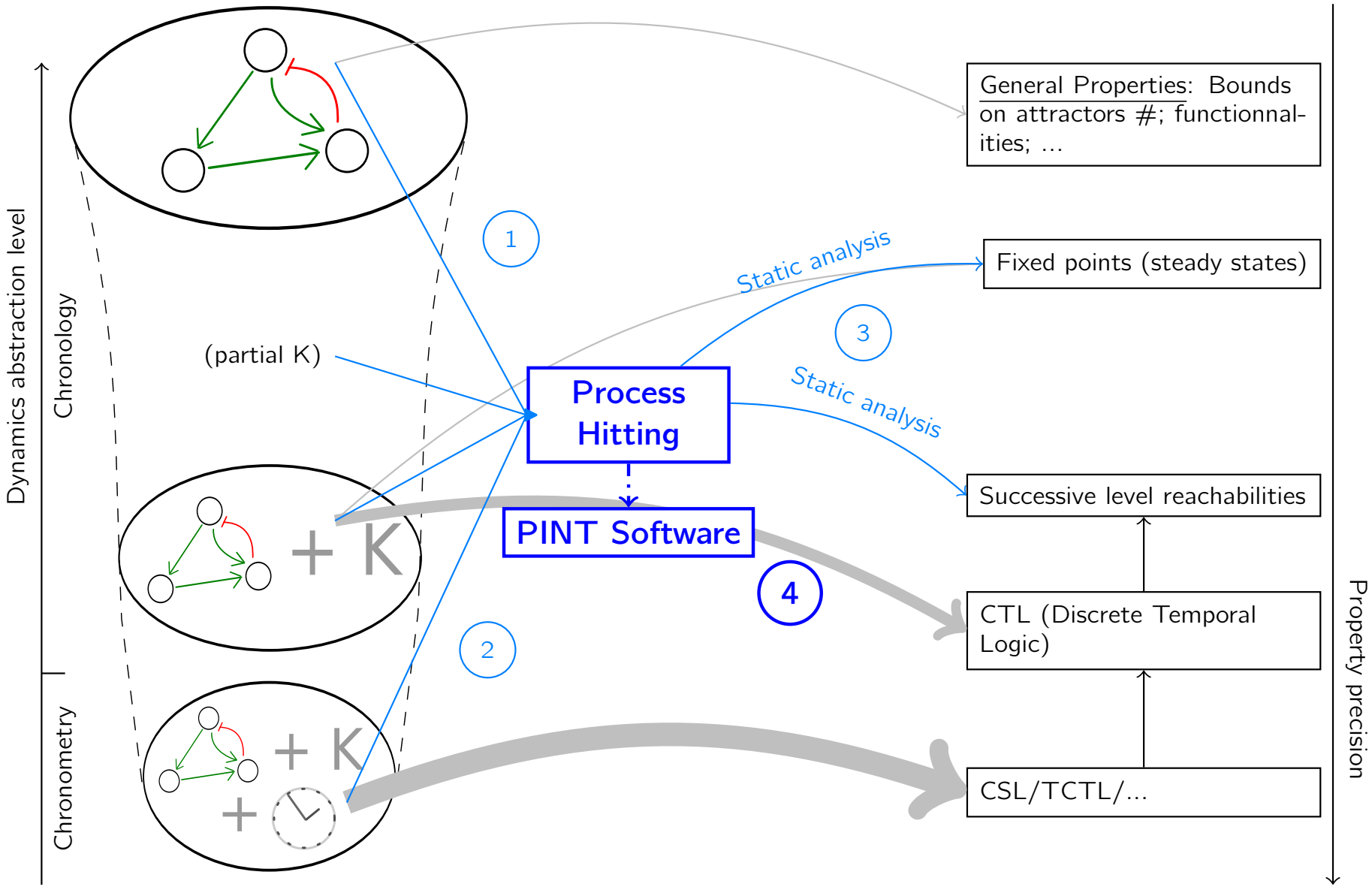
- $\mathbf{BS}^{\wedge}$ computation: exponential in the number of processes within a single sort.
- Size of $\mathbf{BS}^{\wedge}$: combinaisons of $|\mathbf{Proc}_a|$ processes $\binom{|\mathbf{Proc}|}{|\mathbf{Proc}_a|}$.
- Size of $\mathcal{A}^{\omega}_{\varsigma}$ (and $\lceil \mathcal{B}^{\omega}_{\varsigma}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

Analyses

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

Outline


Dynamics abstraction level
Chronology
Chronometry
Property precision
(partial K)
+ K
+ K
+
1
2
3
4
Static analysis
Static analysis
Process Hitting
PINT Software
General Properties: Bounds on attractors #; functionnalities; ...
Fixed points (steady states)
Successive level reachabilities
CTL (Discrete Temporal Logic)
CSL/TCTL/...

Software: PINT

Features

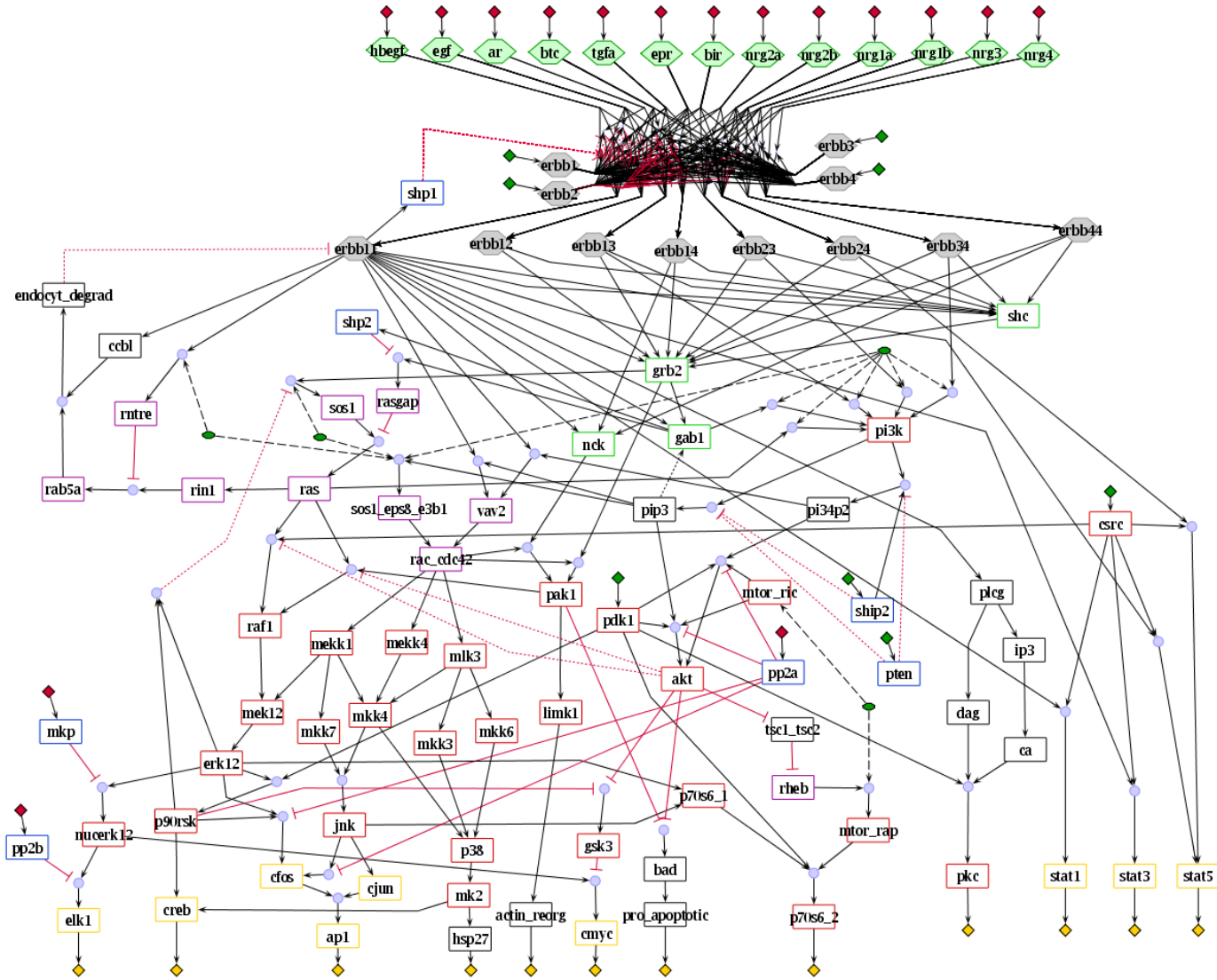
- Textual language describing Process Hittings.
- Simulation;
- Fixed points lister;
- Successive reachability analysis;
- Importations/exportations from/to various formats
  (including bio-oriented softwares: GINsim, CellNetAnalyzer, Biocham).

Technical points

- Implemented in OCaml; library; free software.
- Available at `http://process.hitting.free.fr` .
- Documentations + model repository.

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

For various reachability analysis:

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

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

Conclusion

The Process Hitting

- New formalism for abstract modelling of complex systems.
- Simple (one action type) but not simplist.
- Suitable for modelling Biological Regulatory Networks.

Conclusion

The Process Hitting

- New formalism for abstract modelling of complex systems.
- Simple (one action type) but not simplist.
- Suitable for modelling Biological Regulatory Networks.

Time and stochastic parameters

- Rate + stochasticity absorption factor (variance controller).
- Specification of action duration using confidence interval.
- Simulation (easy); Quantitative model-checking (difficult).

Conclusion

The Process Hitting

- New formalism for abstract modelling of complex systems.
- Simple (one action type) but not simplist.
- Suitable for modelling Biological Regulatory Networks.

Time and stochastic parameters

- Rate + stochasticity absorption factor (variance controller).
- Specification of action duration using confidence interval.
- Simulation (easy); Quantitative model-checking (difficult).

Static discrete analysis of Process Hittings

- Static listing of fixed points.
- Very efficient approximations of successive reachability properties.
- Key processes uncovering (necessary to a given reachability) (towards control).
- Make tractable the formal analysis of large BRNs.

Outlook (1/2)

Applicability extension

Biochemical reactions networks (e.g., Biocham)

- $A + B \rightarrow C + D$, etc.
- Involve transformations and complexations;
- Can be abstracted into Process Hitting (similar to Boolean semantics in Biocham).

Process Hitting with Priorities

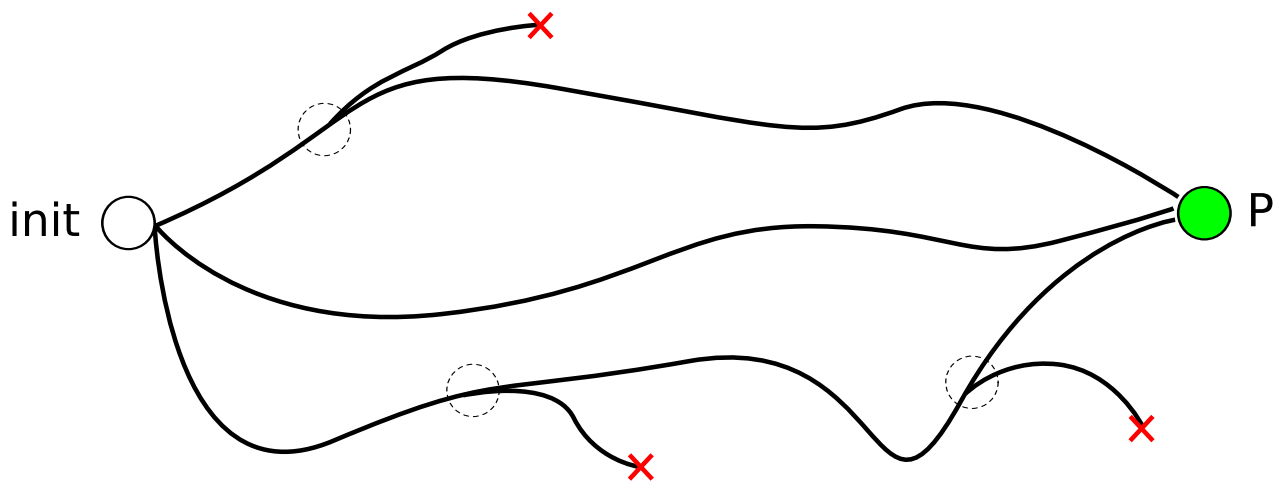
- Actions have (static) priorities of application.
- Allows a weak bisimulation of Petri Nets, etc.
- . . . breaks under-approximations: TODO.

Outlook (2/2)

Analysis extension

Quantitative analysis

- Assign probabilities or time to behaviours;
- Static bifurcation analysis.
- Key actions? (controlling bifurcations)



+ Attractors, automatic refining using bio-data, parameters inference, etc.

Thank you for your attention.