Abstract Interpretation of Dynamics of Biological Regulatory Networks

Loïc Paulevé, Morgan Magnin, Olivier Roux

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

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

IRCCyN / MOVES Team, Nantes (France)

ENS / Working group: Computational Biology
18th January 2010

Overview

Computer science for systems biology

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

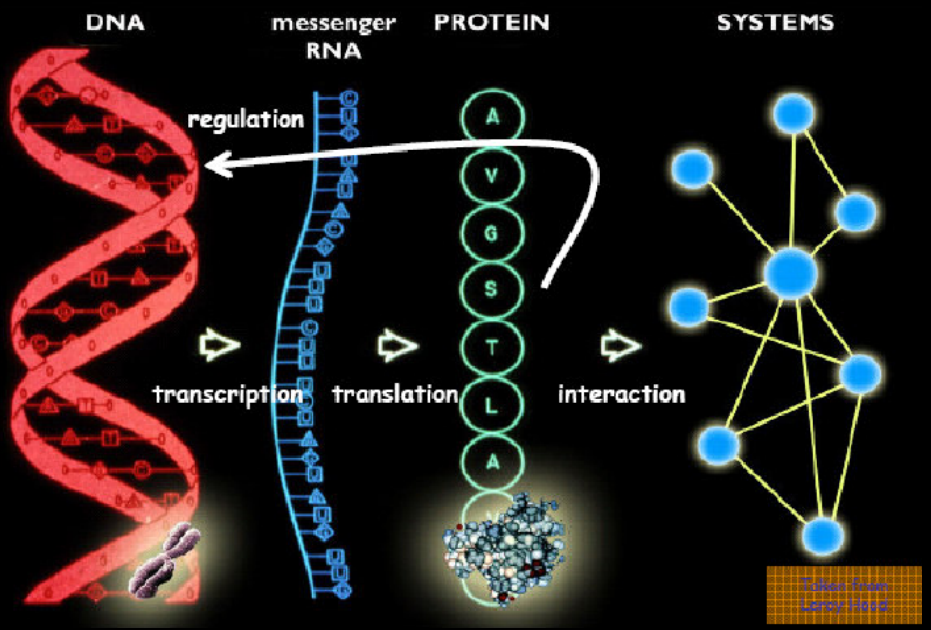
Large scale model checking (dynamical properties)

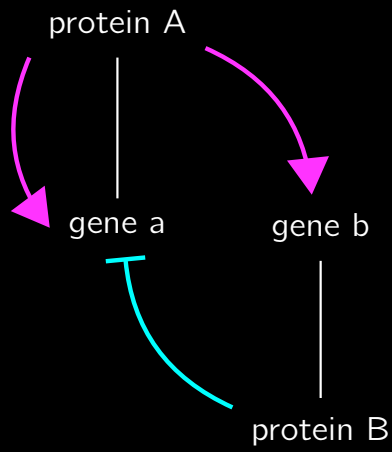
- Cope with state space explosion.
- Our approach: static analysis of the model.

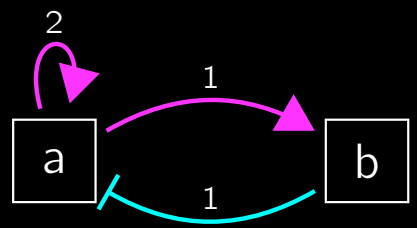
Outline

1. Introduction to BRNs.
2. The Process Hitting.
3. Abstract interpretation and static analysis of reachability properties.
4. Applications to large BRNs + on-going work.

Biological Regulatory Networks (BRNs)





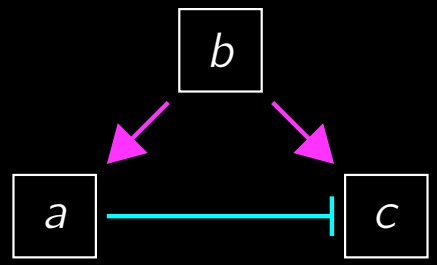


Loïc Paulevé, Morgan Magnin, Olivier Roux

Biological Regulatory Networks (BRNs)

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

Interaction graph



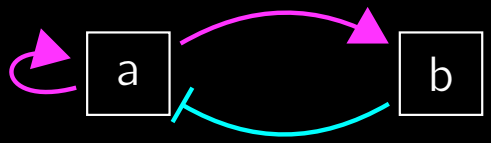
Cooperation

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

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

Dynamics Properties from Interaction Graph

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



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

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

Scalable Dynamics Analysis

Objectives

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

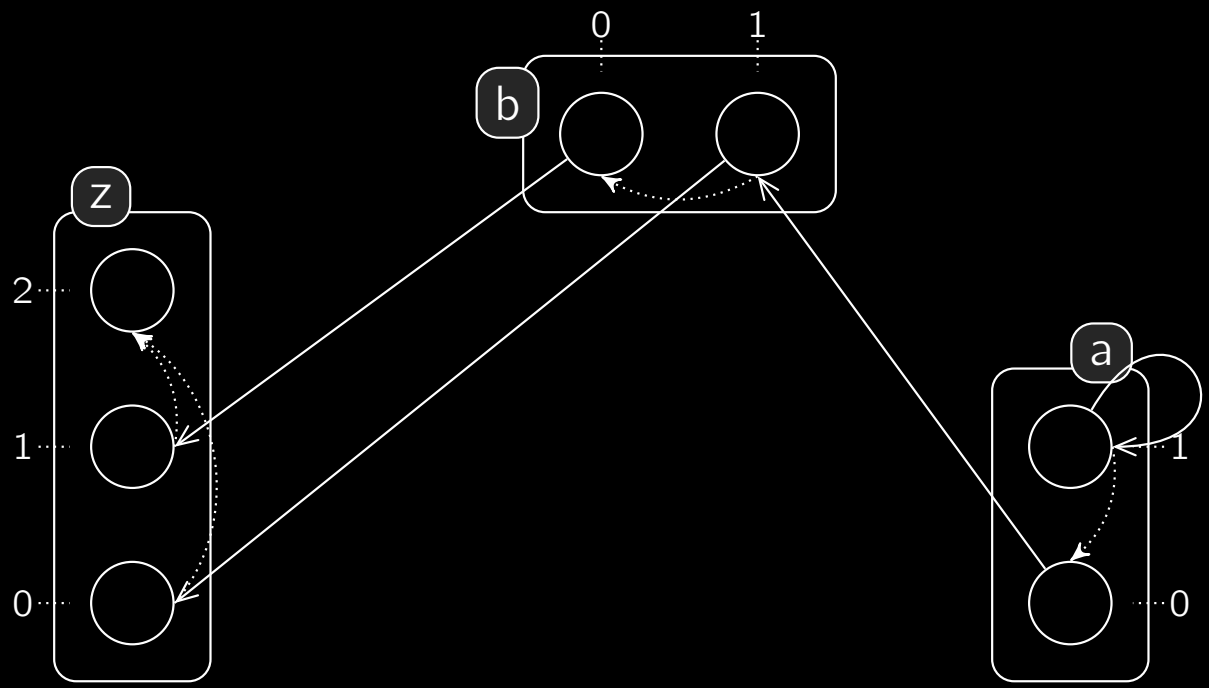
Methods

- Static analysis + Abstract interpretation.
- Focus on the Process Hitting.
- Take advantage of Process Hitting simple structures.

Outline

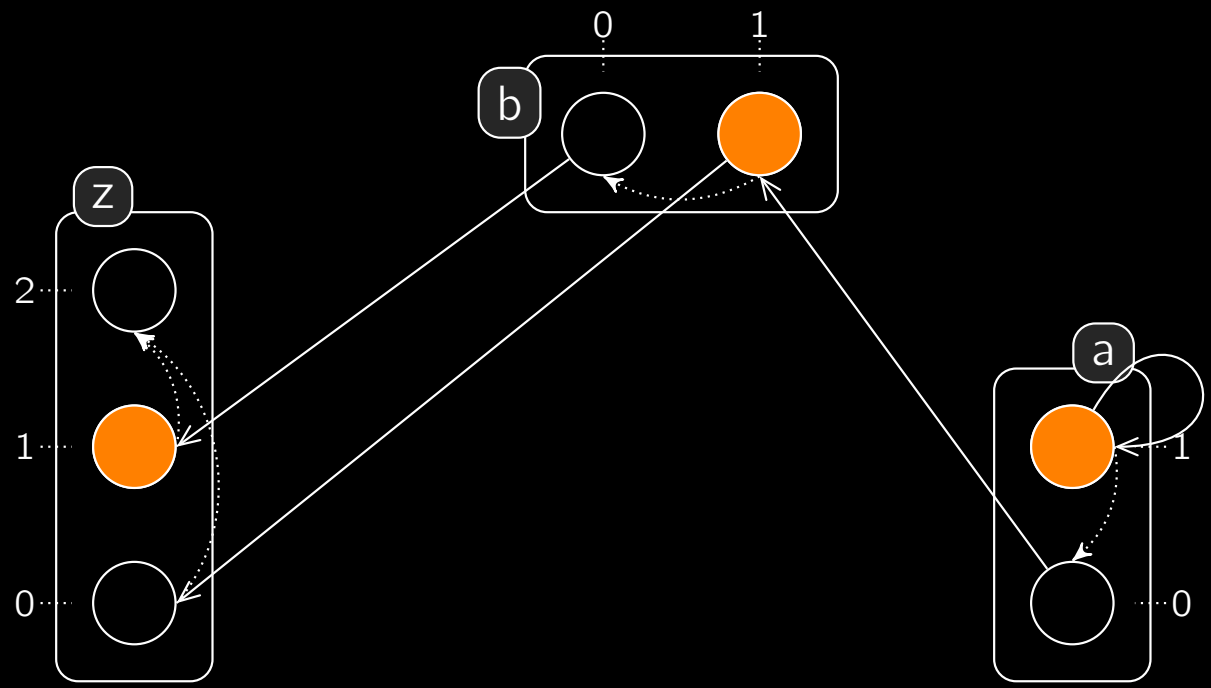
- The Process Hitting.
- Interpretation of BRNs in Process Hitting.
- Static analysis of reachability properties (over- and under-approximations).
- Application to BRNs of 94 and 104 components.

The Process Hitting Framework



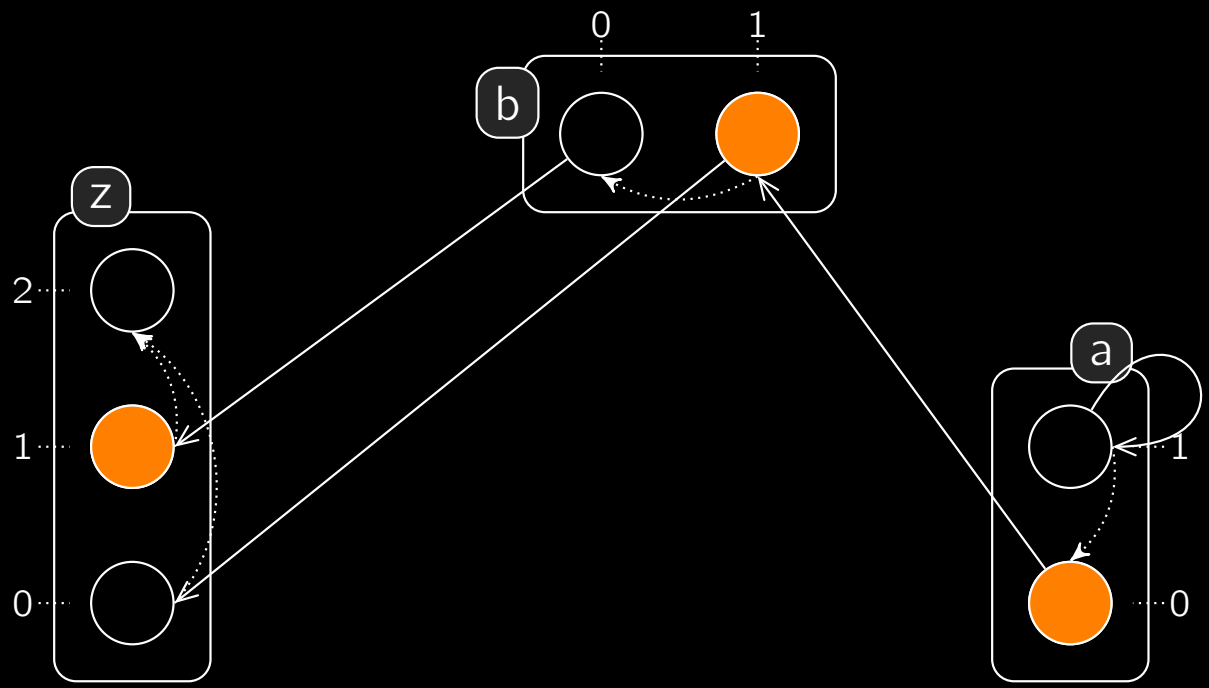
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines.

The Process Hitting Framework



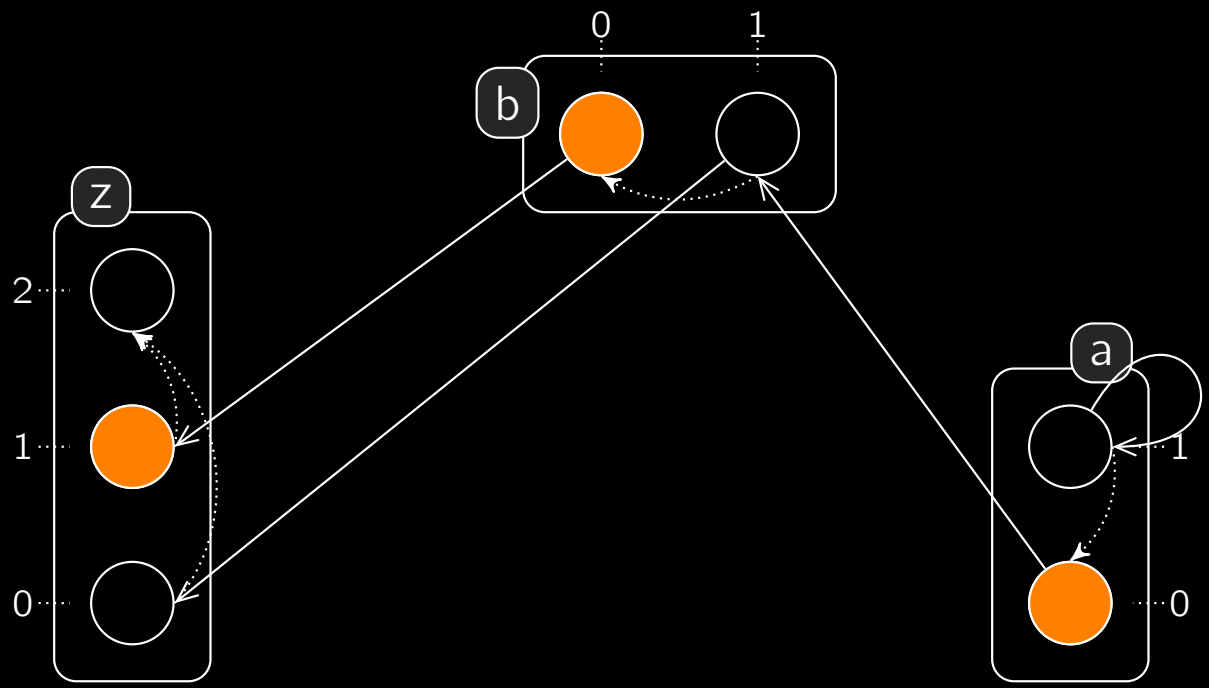
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines.

The Process Hitting Framework



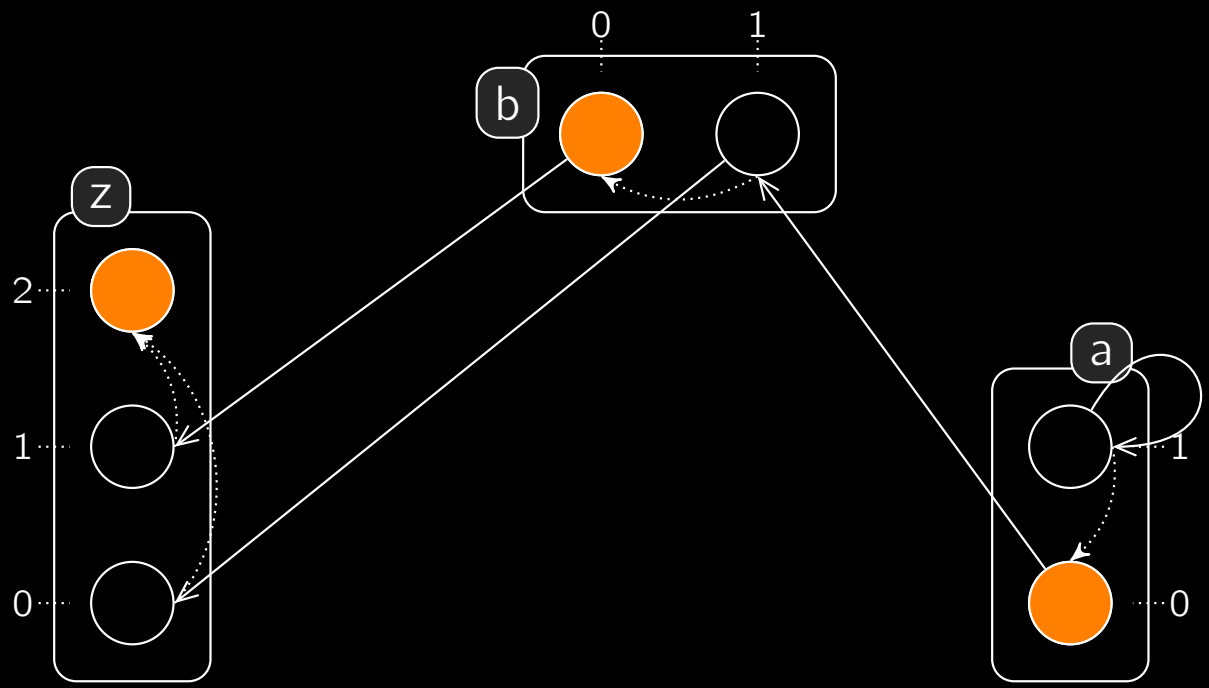
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines.

The Process Hitting Framework



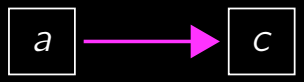
- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines.

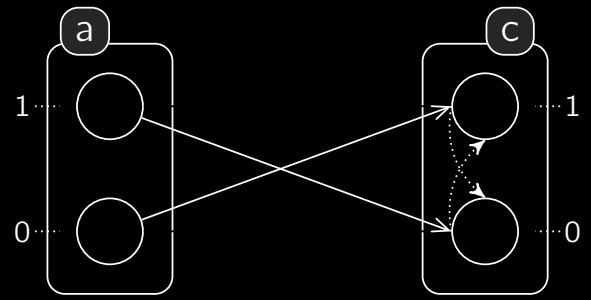
The Process Hitting Framework

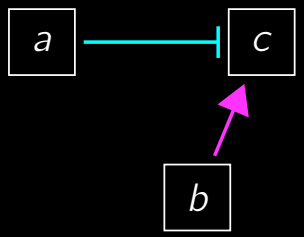


- Sorts: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Communicating Finite-State Machines.

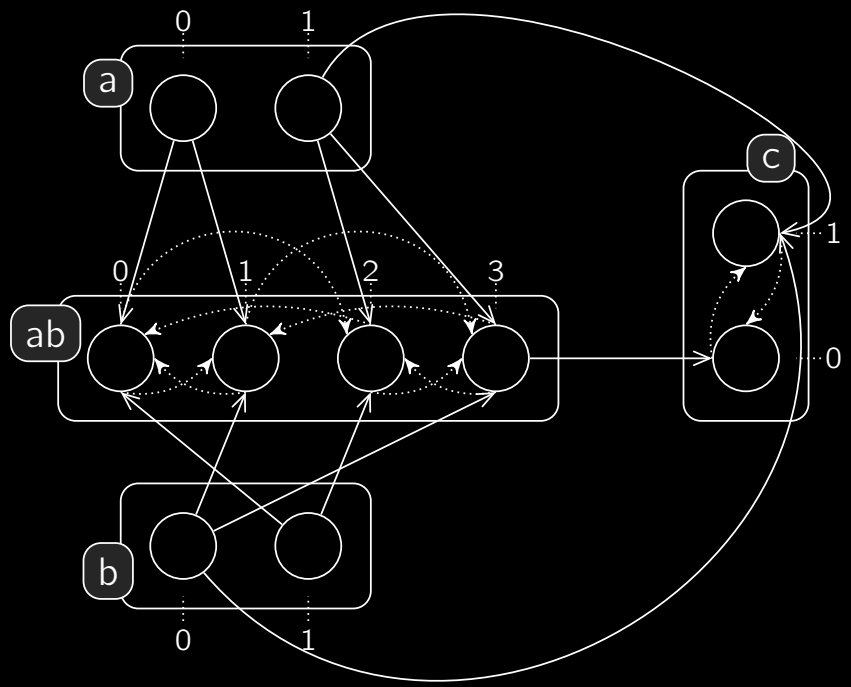
From BRNs to Process Hittings





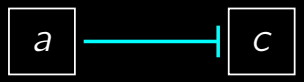


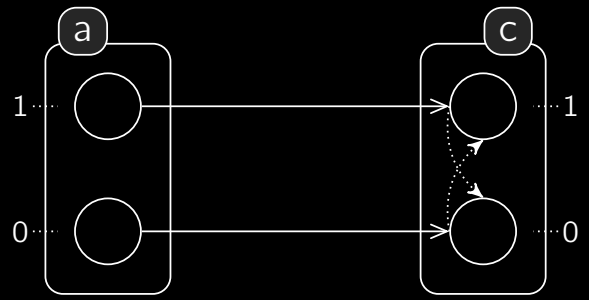
$c = \neg a \wedge b$

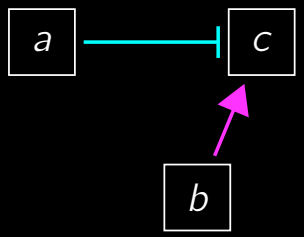


[Paulevé, Magnin, Roux on Trans. on Computational Systems Biology, 2010]

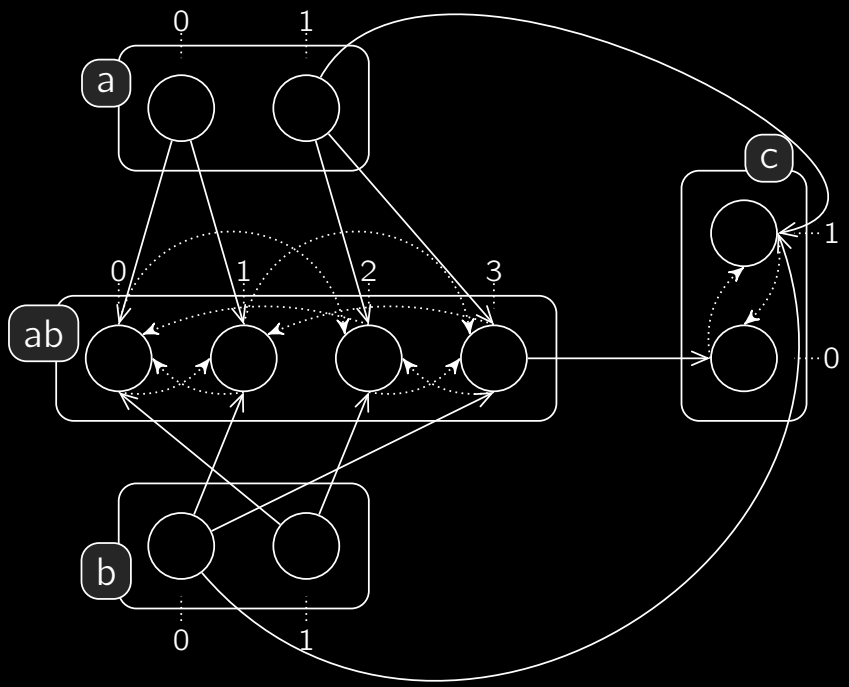
From BRNs to Process Hittings







$c = \neg a \wedge b$



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

The Process Reachability Problem

Scenario: sequence of successively playable actions.

$$b_1 \rightarrow a_0 \,\lceil\!\!\nearrow\, a_1\,,\; a_1 \rightarrow b_1 \,\lceil\!\!\nearrow\, b_0\,,\; b_0 \rightarrow d_0 \,\lceil\!\!\nearrow\, d_1\,,\; c_0 \rightarrow d_1 \,\lceil\!\!\nearrow\, d_2$$

The Process Reachability Problem

Scenario: sequence of successively playable actions.

$$b_1 \rightarrow a_0 \lceil\!\!\nearrow a_1,\ a_1 \rightarrow b_1 \lceil\!\!\nearrow b_0,\ b_0 \rightarrow d_0 \lceil\!\!\nearrow d_1,\ c_0 \rightarrow d_1 \lceil\!\!\nearrow d_2$$

Abstraction by objective sequences

- $a_0 \lceil\!\!\nearrow^* a_1,\ b_1 \lceil\!\!\nearrow^* b_0,\ d_0 \lceil\!\!\nearrow^* d_1,\ d_1 \lceil\!\!\nearrow^* d_2$

The Process Reachability Problem

Scenario: sequence of successively playable actions.

$$b_1 \rightarrow a_0 \lceil\!\!\nearrow a_1,\ a_1 \rightarrow b_1 \lceil\!\!\nearrow b_0,\ b_0 \rightarrow d_0 \lceil\!\!\nearrow d_1,\ c_0 \rightarrow d_1 \lceil\!\!\nearrow d_2$$

Abstraction by objective sequences

- $a_0 \lceil\!\!\nearrow^* a_1,\ b_1 \lceil\!\!\nearrow^* b_0,\ d_0 \lceil\!\!\nearrow^* d_1,\ d_1 \lceil\!\!\nearrow^* d_2$
- $b_1 \lceil\!\!\nearrow^* b_0,\ d_0 \lceil\!\!\nearrow^* d_2$

The Process Reachability Problem

Scenario: sequence of successively playable actions.

$$b_1 \rightarrow a_0 \,\vec{\lceil}\, a_1,\ a_1 \rightarrow b_1 \,\vec{\lceil}\, b_0,\ b_0 \rightarrow d_0 \,\vec{\lceil}\, d_1,\ c_0 \rightarrow d_1 \,\vec{\lceil}\, d_2$$

Abstraction by objective sequences

- $a_0 \,\vec{\lceil}^*\, a_1,\ b_1 \,\vec{\lceil}^*\, b_0,\ d_0 \,\vec{\lceil}^*\, d_1,\ d_1 \,\vec{\lceil}^*\, d_2$
- $b_1 \,\vec{\lceil}^*\, b_0,\ d_0 \,\vec{\lceil}^*\, d_2$
- $d_0 \,\vec{\lceil}^*\, d_2$

The Process Reachability Problem

Scenario: sequence of successively playable actions.

$$b_1 \rightarrow a_0 \,\lceil\!\!\nearrow\, a_1\,,\; a_1 \rightarrow b_1 \,\lceil\!\!\nearrow\, b_0\,,\; b_0 \rightarrow d_0 \,\lceil\!\!\nearrow\, d_1\,,\; c_0 \rightarrow d_1 \,\lceil\!\!\nearrow\, d_2$$

Abstraction by objective sequences

- $a_0 \,\lceil\!\!\nearrow^* a_1\,,\; b_1 \,\lceil\!\!\nearrow^* b_0\,,\; d_0 \,\lceil\!\!\nearrow^* d_1\,,\; d_1 \,\lceil\!\!\nearrow^* d_2$
- $b_1 \,\lceil\!\!\nearrow^* b_0\,,\; d_0 \,\lceil\!\!\nearrow^* d_2$
- $d_0 \,\lceil\!\!\nearrow^* d_2$
- ...

Abstraction by bounce sequences

- $a_1 \rightarrow b_1 \,\lceil\!\!\nearrow\, b_0 \;\; (b_1 \,\lceil\!\!\nearrow^* b_0)$

The Process Reachability Problem

Scenario: sequence of successively playable actions.

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

Abstraction by objective sequences

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

Abstraction by bounce sequences

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

The Process Reachability Problem

Scenario: sequence of successively playable actions.

$$b_1 \rightarrow a_0 \,\lceil\!\!\nearrow\, a_1,\ a_1 \rightarrow b_1 \,\lceil\!\!\nearrow\, b_0,\ b_0 \rightarrow d_0 \,\lceil\!\!\nearrow\, d_1,\ c_0 \rightarrow d_1 \,\lceil\!\!\nearrow\, d_2$$

Abstraction by objective sequences

- $a_0 \,\lceil\!\!\nearrow^*\, a_1,\ b_1 \,\lceil\!\!\nearrow^*\, b_0,\ d_0 \,\lceil\!\!\nearrow^*\, d_1,\ d_1 \,\lceil\!\!\nearrow^*\, d_2$
- $b_1 \,\lceil\!\!\nearrow^*\, b_0,\ d_0 \,\lceil\!\!\nearrow^*\, d_2$
- $d_0 \,\lceil\!\!\nearrow^*\, d_2$
- ...

Abstraction by bounce sequences

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

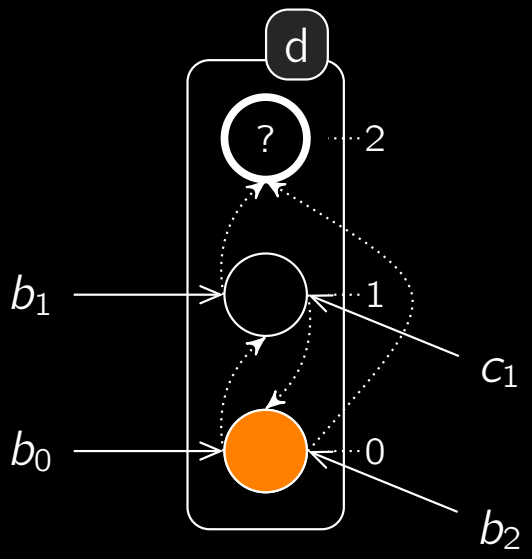
Process Reachability

Is an objective sequence $a_i \,\lceil\!\!\nearrow^*\, a_j, \ldots, z_k \,\lceil\!\!\nearrow^*\, z_l$ concretizable in a state s?

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

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

(Abstracted) Bounce Sequences



Objectives

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

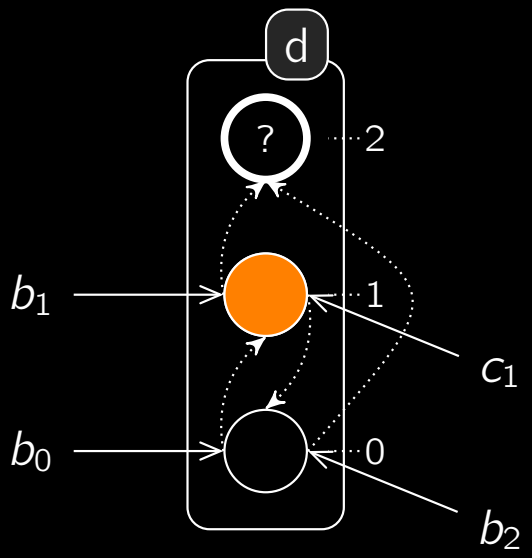
Bounce Sequences

Only minimal sequences are kept.

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

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

(Abstracted) Bounce Sequences



Objectives

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

Bounce Sequences

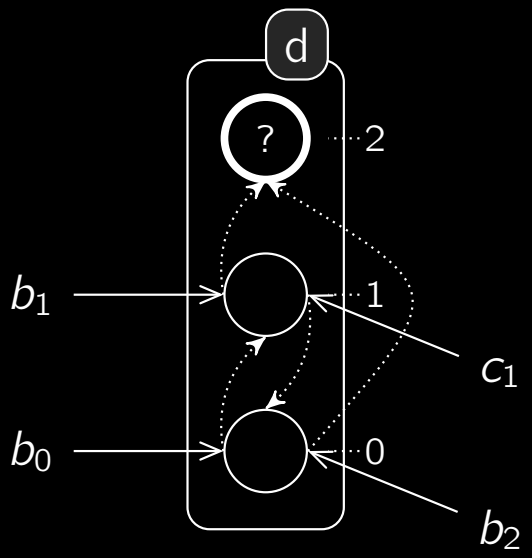
Only minimal sequences are kept.

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

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

Loïc Paulevé, Morgan Magnin, Olivier Roux

(Abstracted) Bounce Sequences



Objectives

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

Bounce Sequences

Only minimal sequences are kept.

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

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

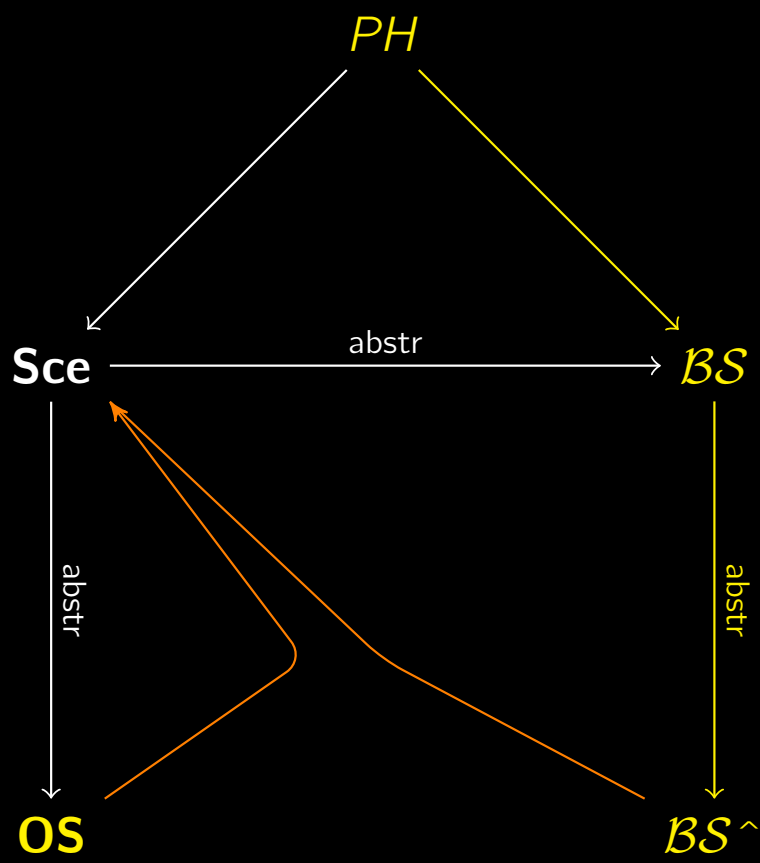
Abstracted Bounce Sequences

Only hitters are kept, and the order is forgotten.

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

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

Overall Approach



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

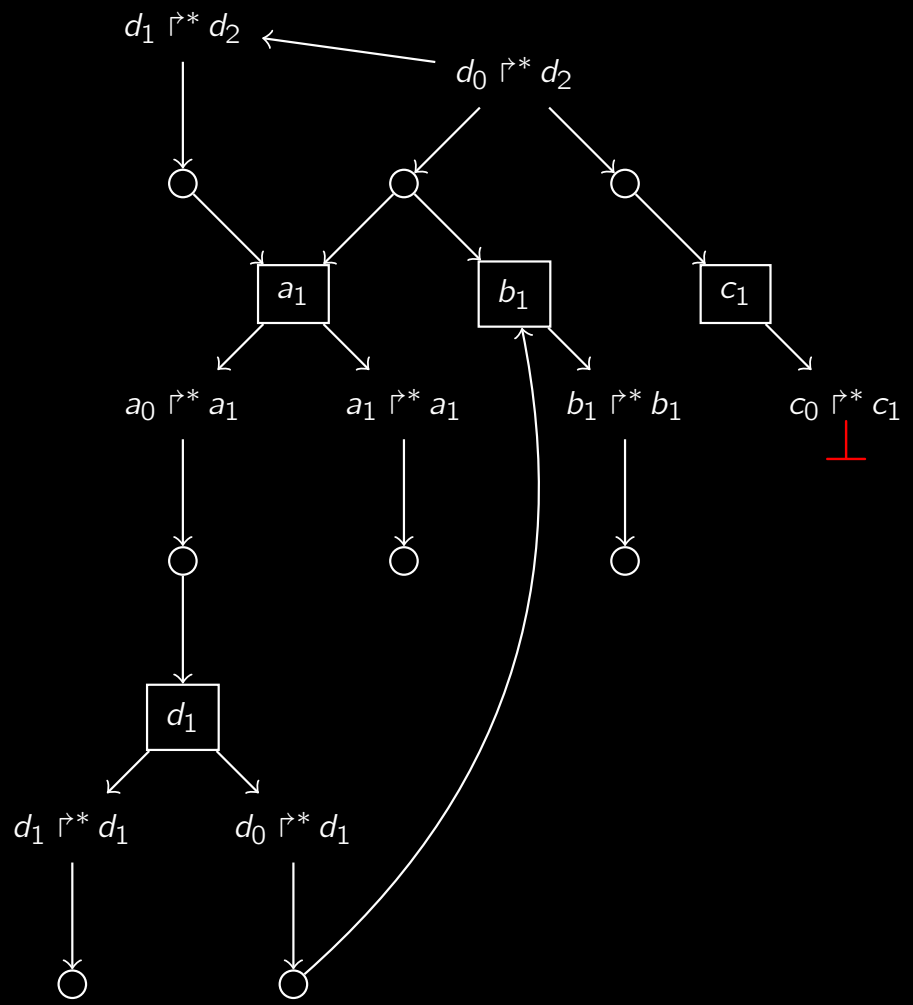
$\omega = a_0 \Rsh^* a_1, d_0 \Rsh^* d_2$

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

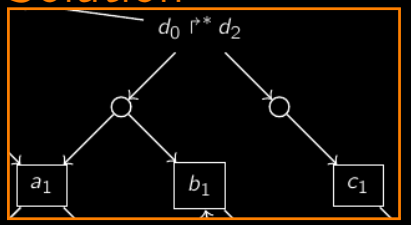
$\mathcal{BS}^\wedge(a_0 \Rsh^* a_1) = \{\{b_1\}\}$
$\mathcal{BS}^\wedge(d_0 \Rsh^* d_2) = \{\{a_1, b_0\}\}$

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

Process Hitting Abstract Structure

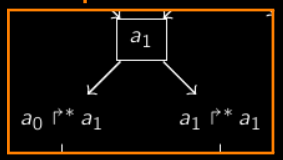


Solution



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

Requirement



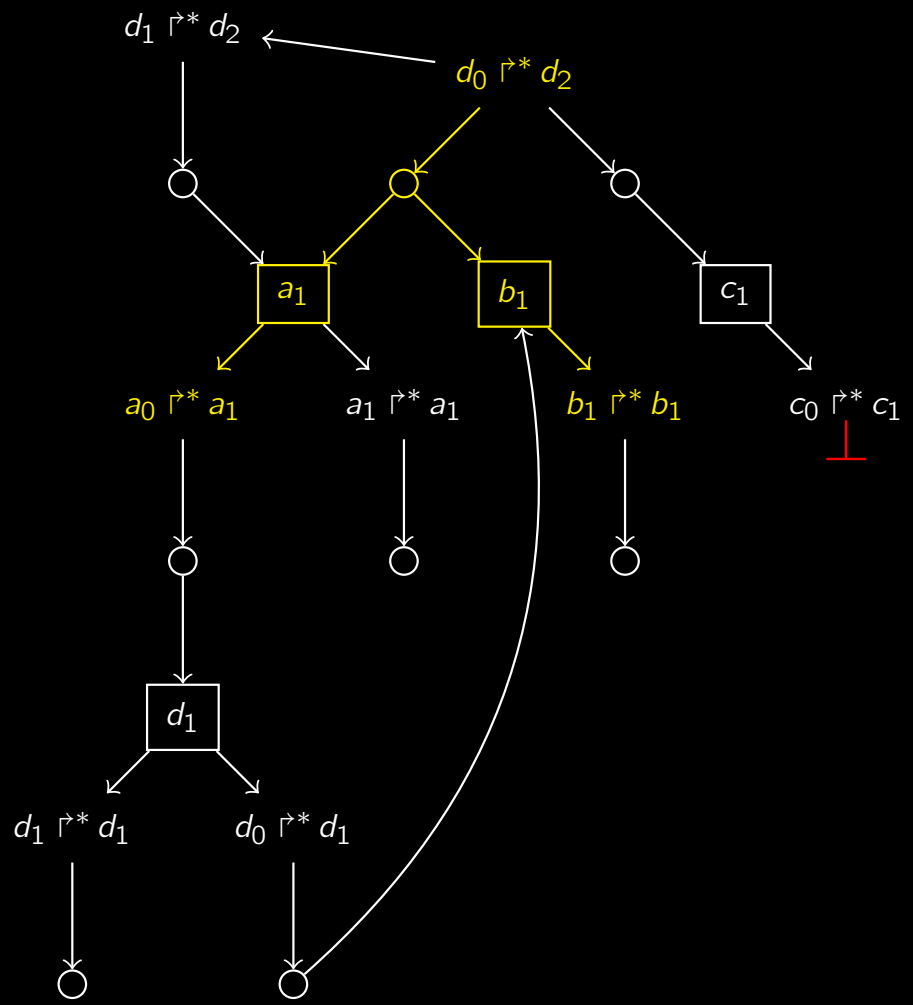
Objective to resolve (from current state).

Continuity

Objective resolution split.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Process Hitting Abstract Structure



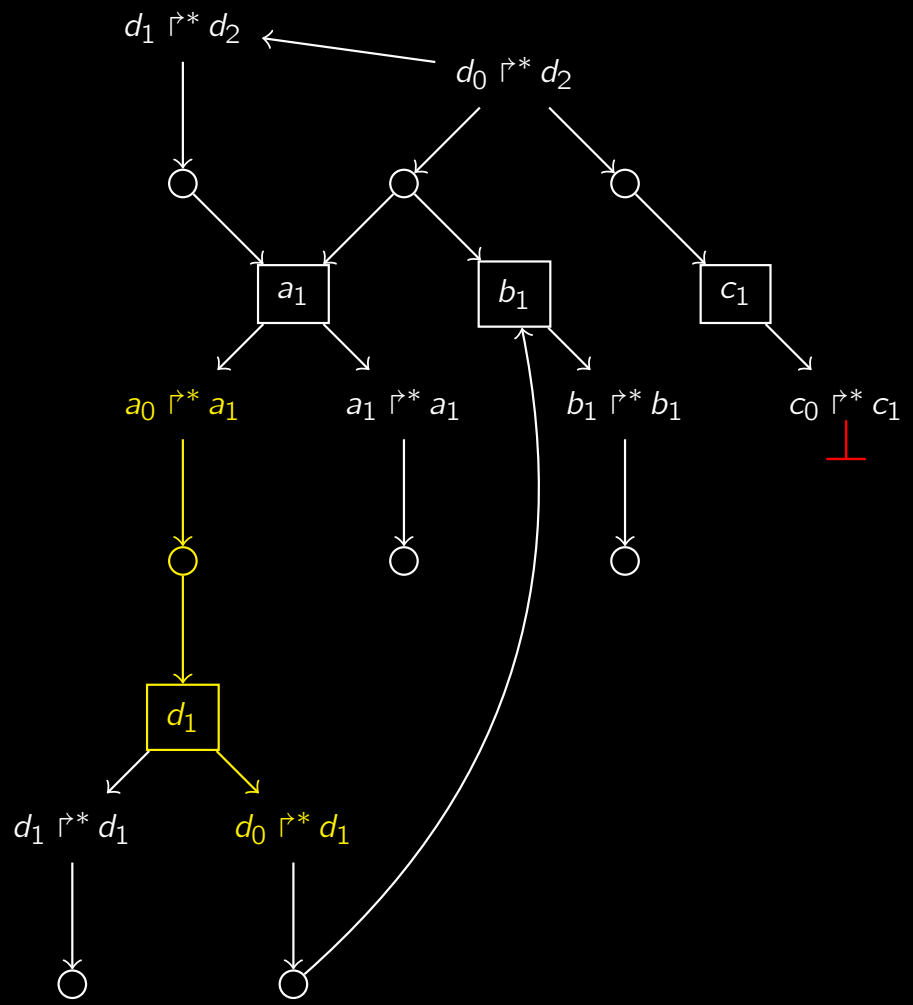
Starting with state $\langle a_0, b_1, c_0, d_0 \rangle$:

$$d_0 \nearrow^* d_2$$

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

Loïc Paulevé, Morgan Magnin, Olivier Roux

Process Hitting Abstract Structure



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

$$d_0 \nearrow^* d_2$$

$$\wr$$

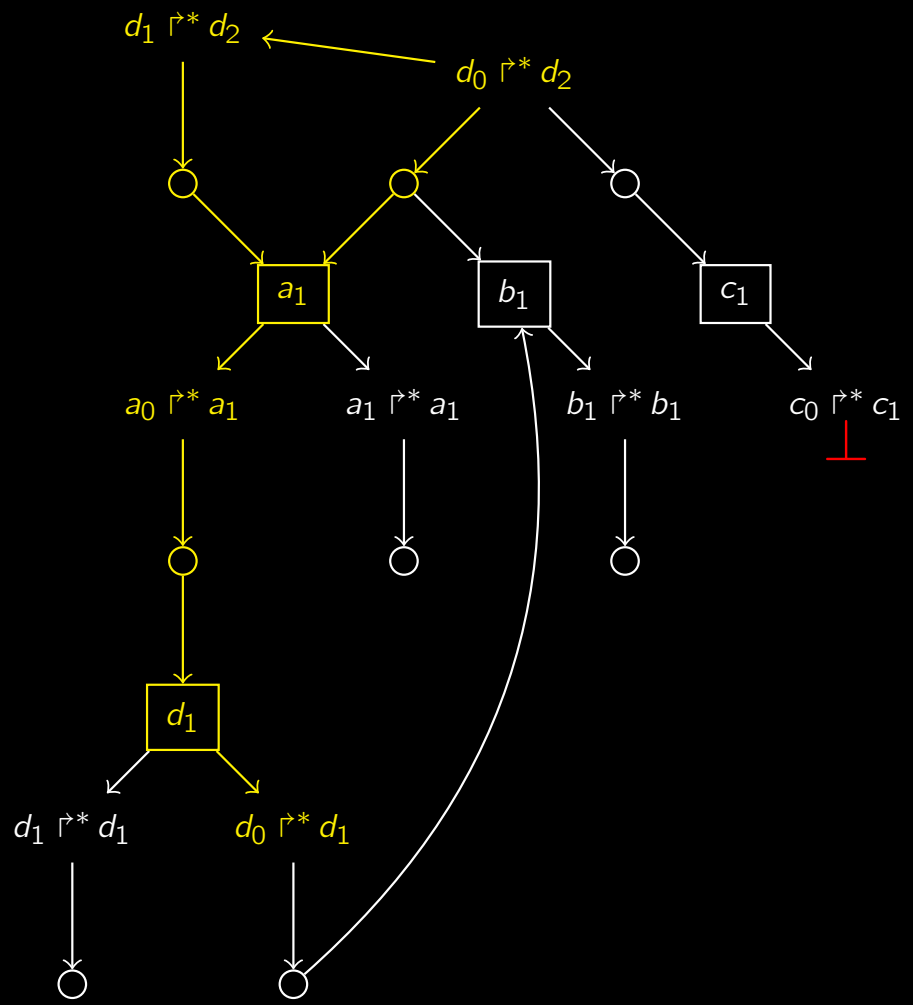
$$\begin{cases} a_0 \nearrow^* a_1, b_1 \nearrow^* b_1, d_0 \nearrow^* d_2 \\ b_1 \nearrow^* b_1, a_0 \nearrow^* a_1, d_0 \nearrow^* d_2 \end{cases}$$

$$a_0 \nearrow^* a_1$$

$$\wr$$

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

Process Hitting Abstract Structure



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

$$d_0 \nearrow^* d_2$$

$$\downarrow$$

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

$$a_0 \nearrow^* a_1$$

$$\downarrow$$

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

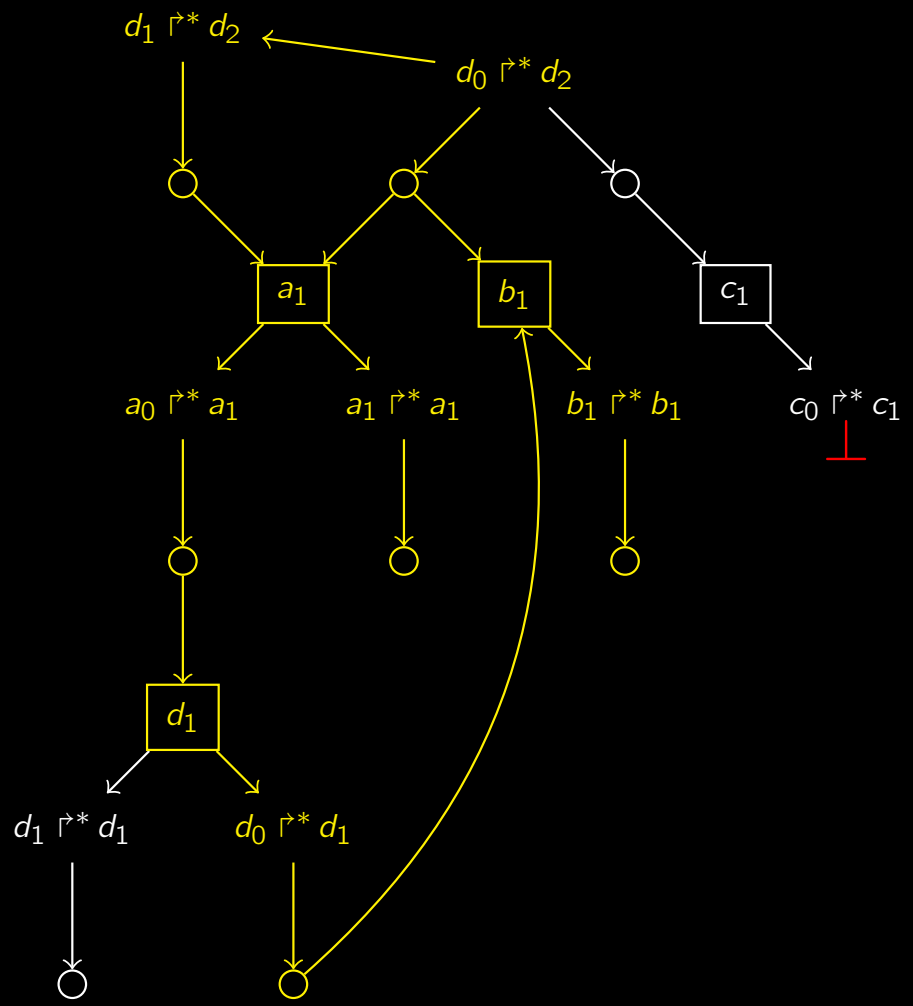
$$a_0 \nearrow^* a_1, d_0 \nearrow^* d_2$$

$$\downarrow$$

$$d_0 \nearrow^* d_1, a_0 \nearrow^* a_1, d_1 \nearrow^* d_2$$

Loïc Paulevé, Morgan Magnin, Olivier Roux

Process Hitting Abstract Structure



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

$$d_0 \nearrow^* d_2$$

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

$$a_0 \nearrow^* a_1$$

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

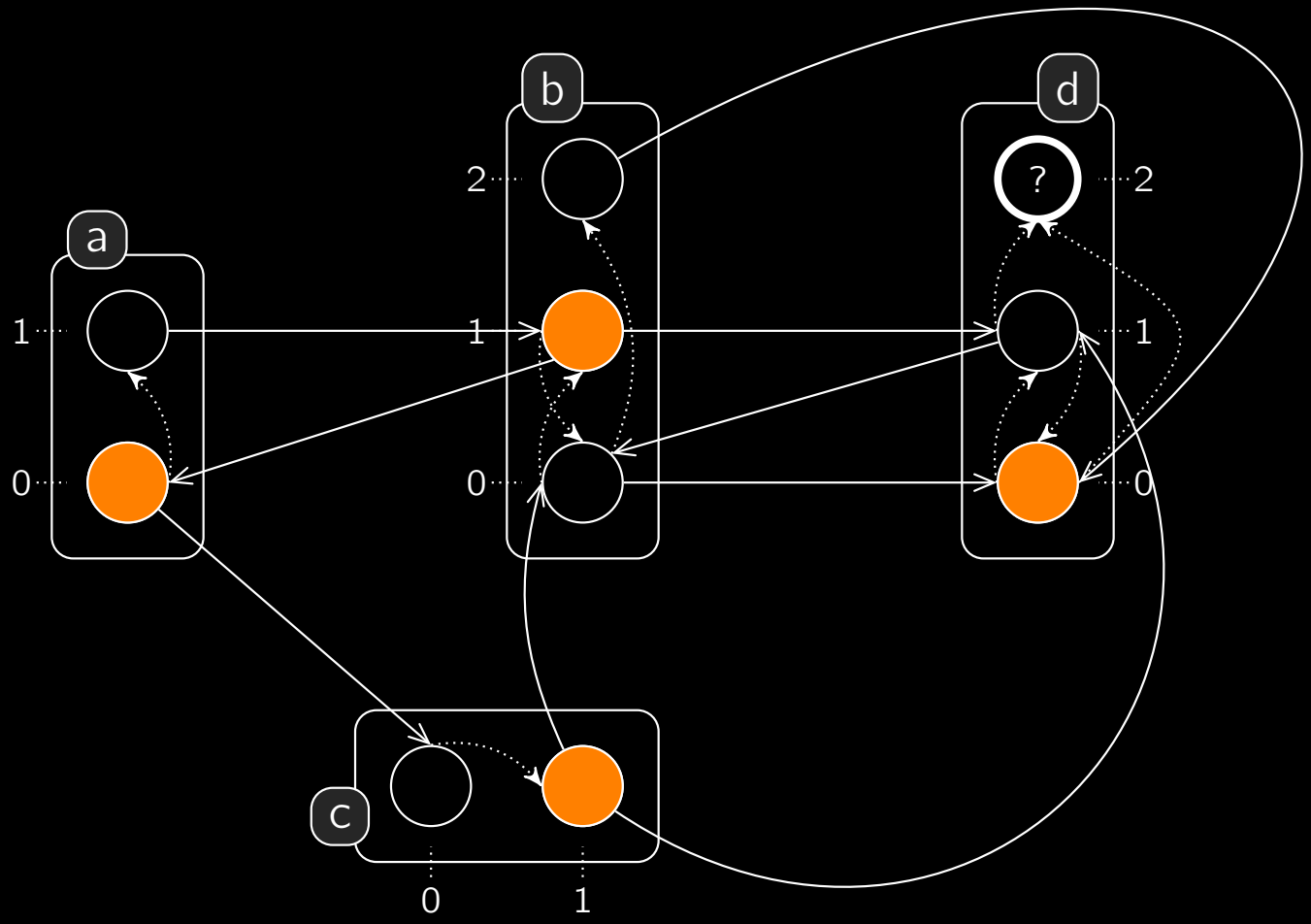
$$a_0 \nearrow^* a_1, d_0 \nearrow^* d_2$$

$$d_0 \nearrow^* d_1, a_0 \nearrow^* a_1, d_1 \nearrow^* d_2$$

$$d_0 \nearrow^* d_2$$

$$b_1 \nearrow^* b_1, d_0 \nearrow^* d_1, a_0 \nearrow^* a_1, d_1 \nearrow^* d_2$$

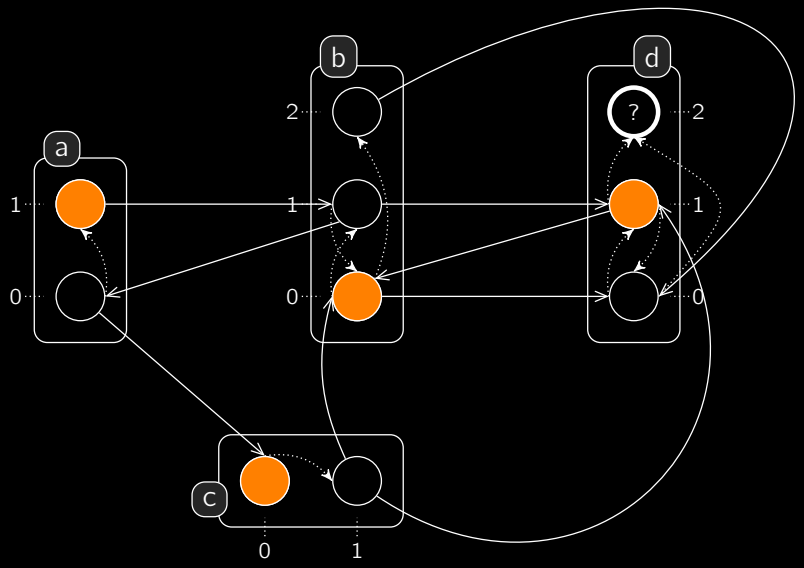
Running Example



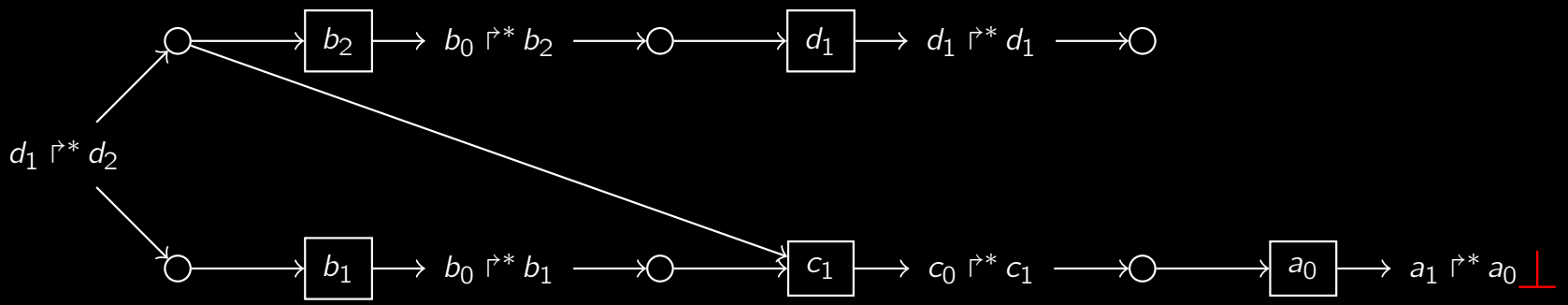
Is objective sequence $d_0 \curvearrowright^* d_2$ concretizable?

Loïc Paulevé, Morgan Magnin, Olivier Roux

Over-approximation of Process Reachability

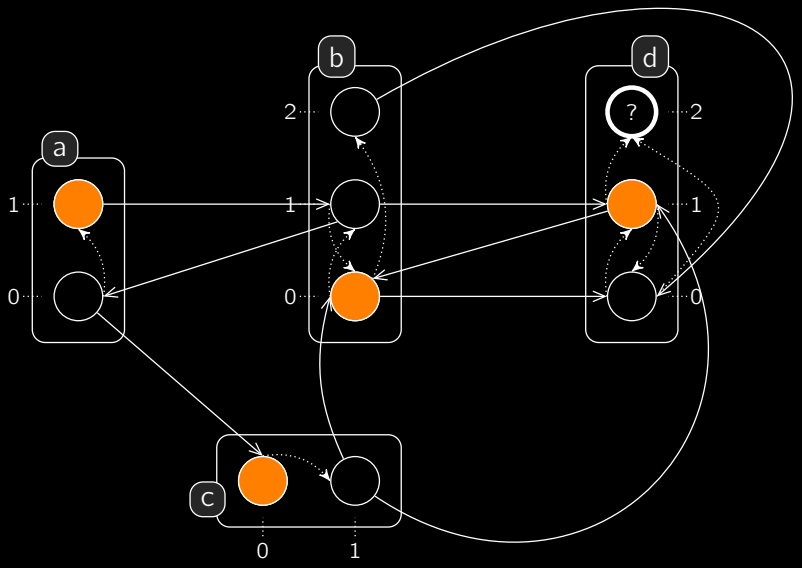


- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailed);
- Necessary condition: there always exists a solution ending with a trivial objective.

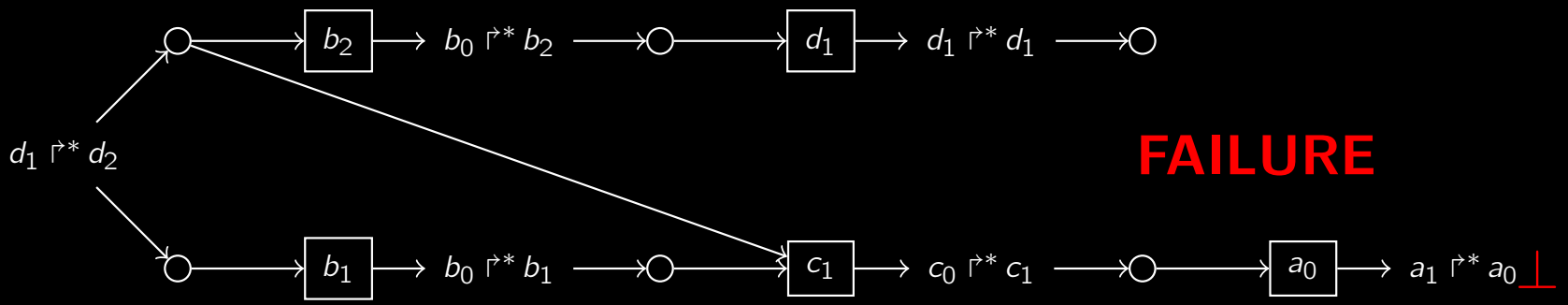


Loïc Paulevé, Morgan Magnin, Olivier Roux

Over-approximation of Process Reachability

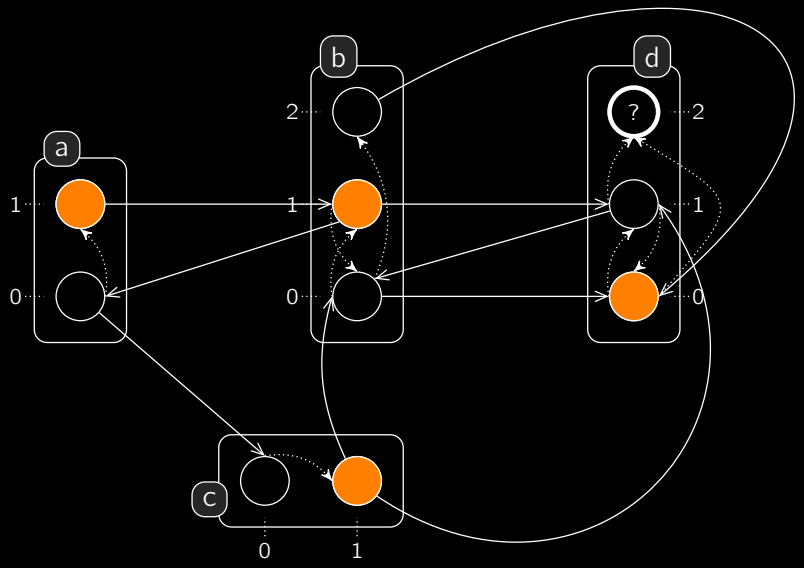


- Focus on objectives starting from the initial state;
- Add required objective redirections (not detailed);
- Necessary condition: there always exists a solution ending with a trivial objective.

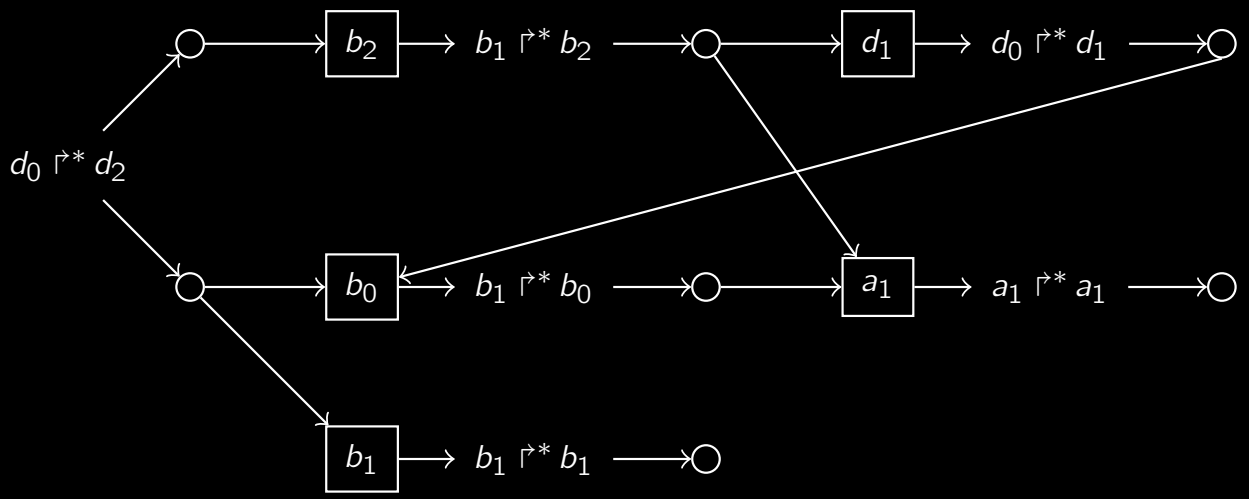


Loïc Paulevé, Morgan Magnin, Olivier Roux

Over-approximation of Process Reachability



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



OK

Loïc Paulevé, Morgan Magnin, Olivier Roux

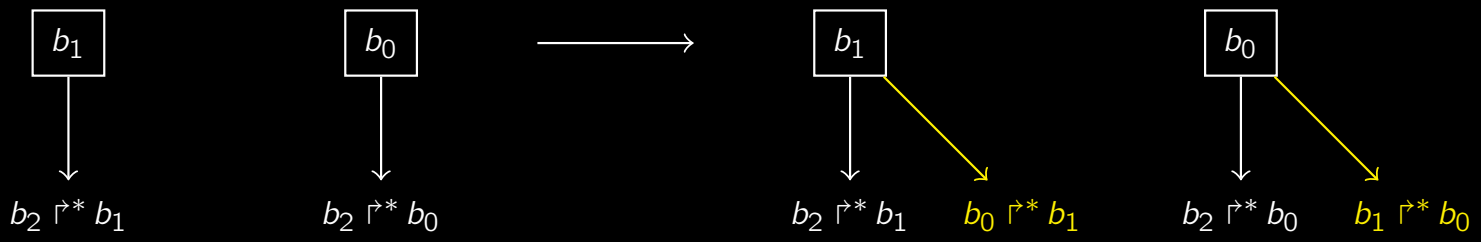
Under-Approximation of Process Reachability

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

Conditions

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

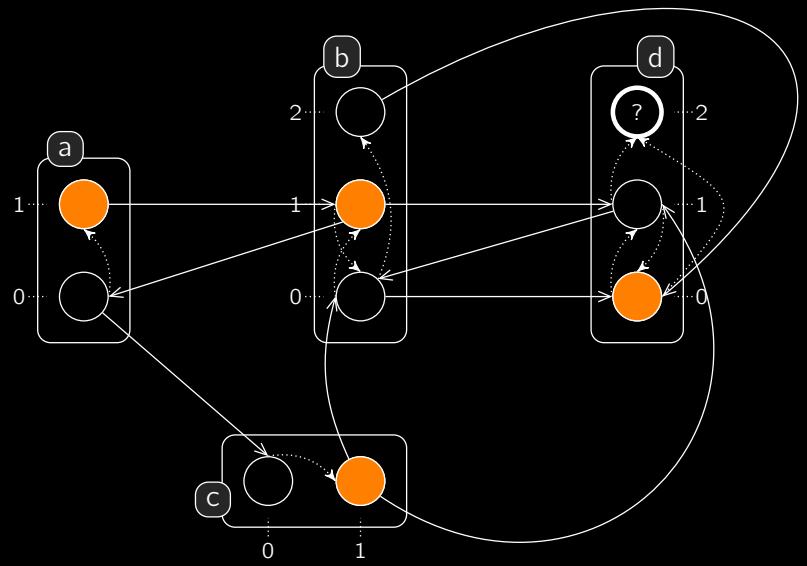
Requirements saturation

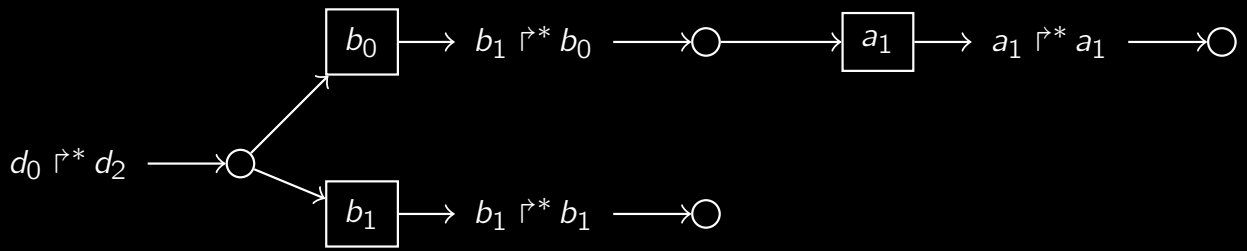


Remark

To increase conclusiveness, one can arbitrarily select objective solutions.

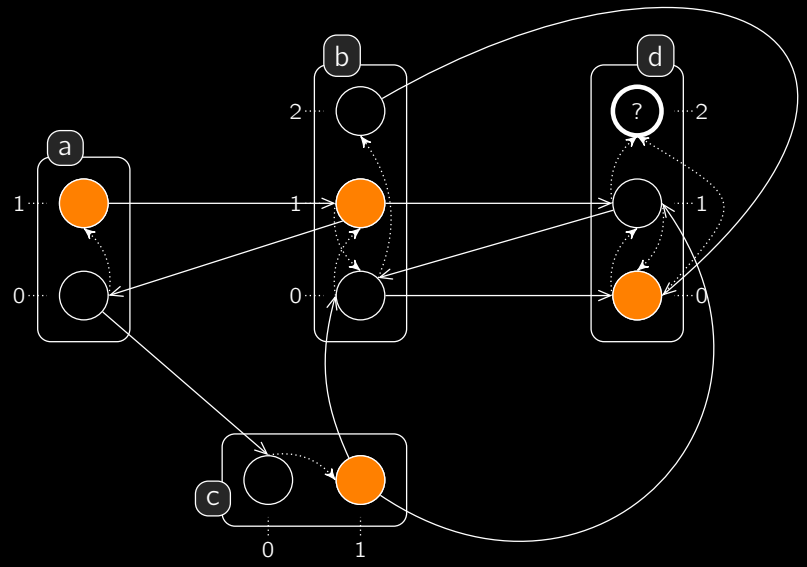
Under-Approximation of Process Reachability

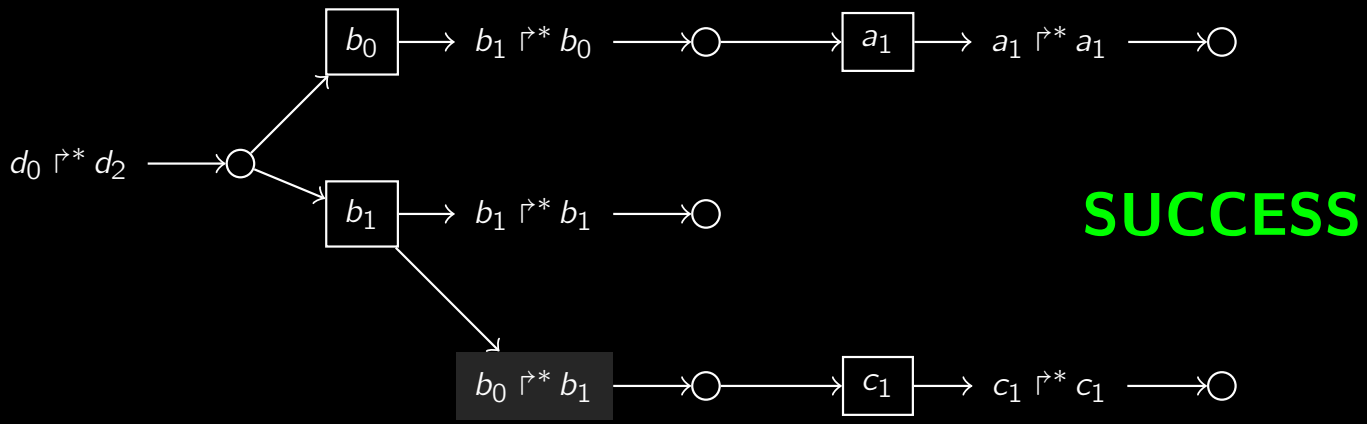




Loïc Paulevé, Morgan Magnin, Olivier Roux

Under-Approximation of Process Reachability

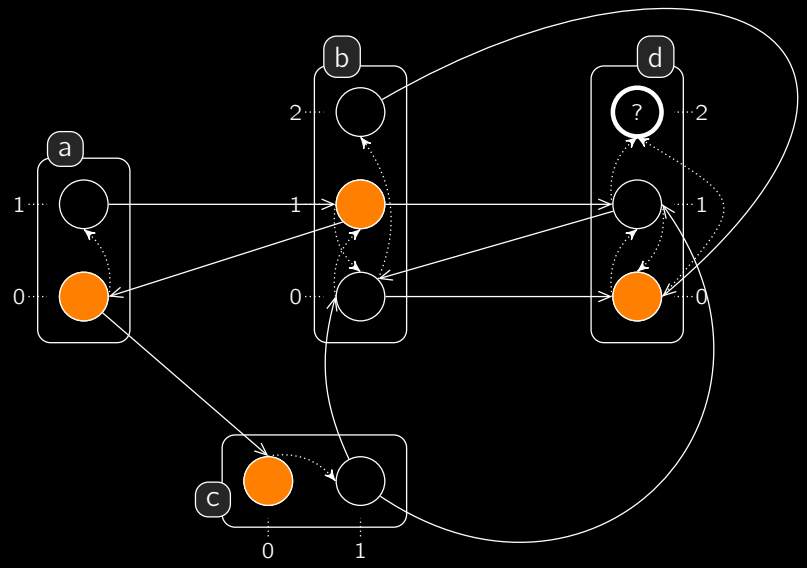


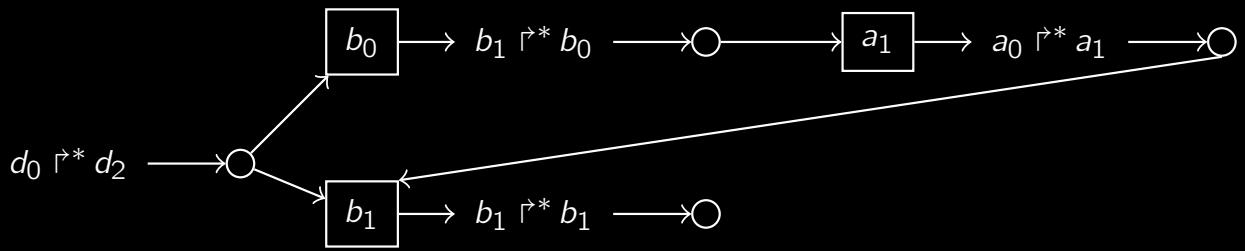


SUCCESS

Loïc Paulevé, Morgan Magnin, Olivier Roux

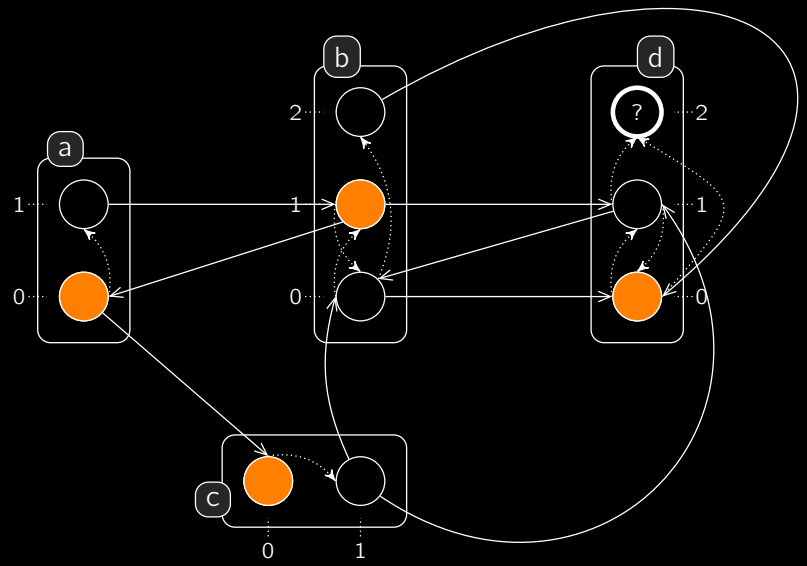
Under-Approximation of Process Reachability

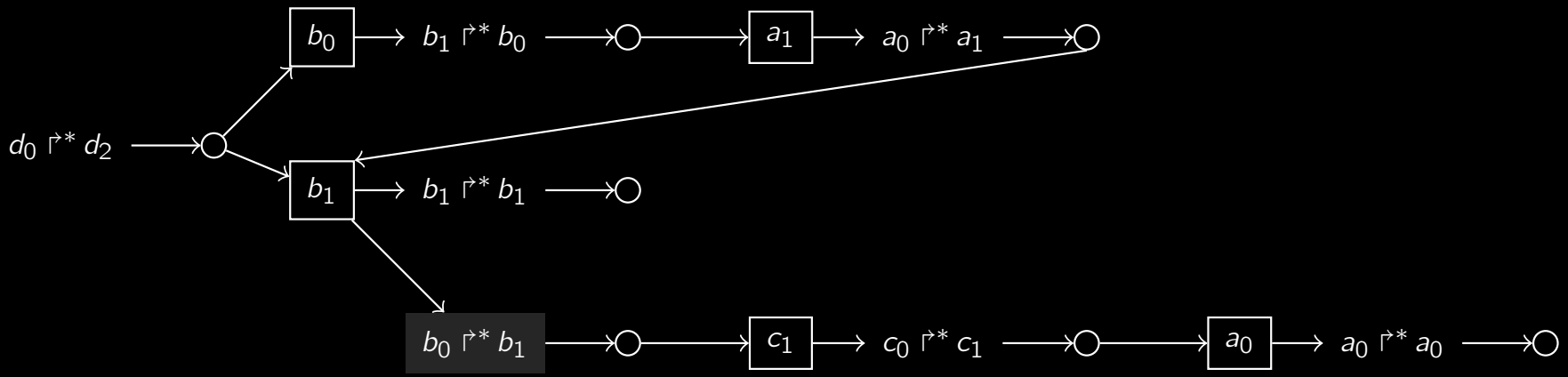




Loïc Paulevé, Morgan Magnin, Olivier Roux

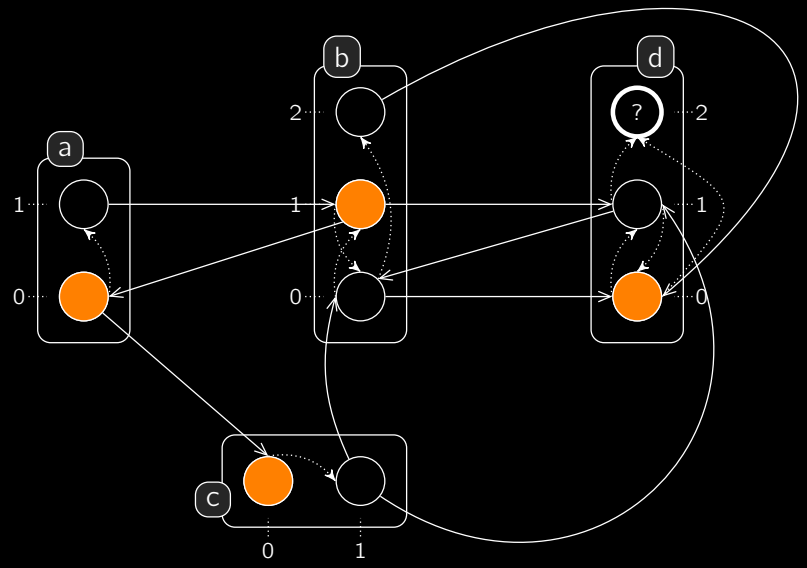
Under-Approximation of Process Reachability

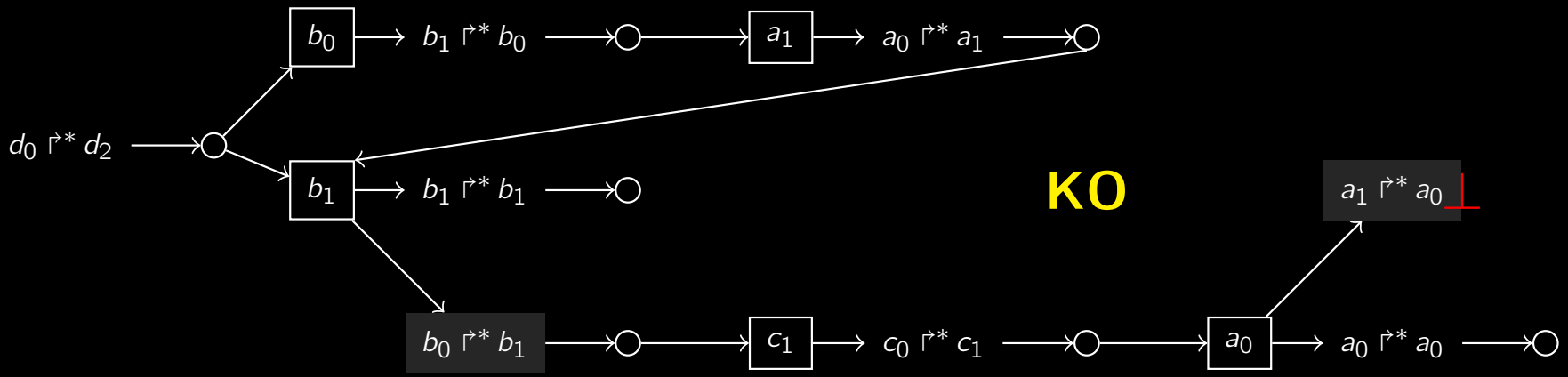




Loïc Paulevé, Morgan Magnin, Olivier Roux

Under-Approximation of Process Reachability





Loïc Paulevé, Morgan Magnin, Olivier Roux

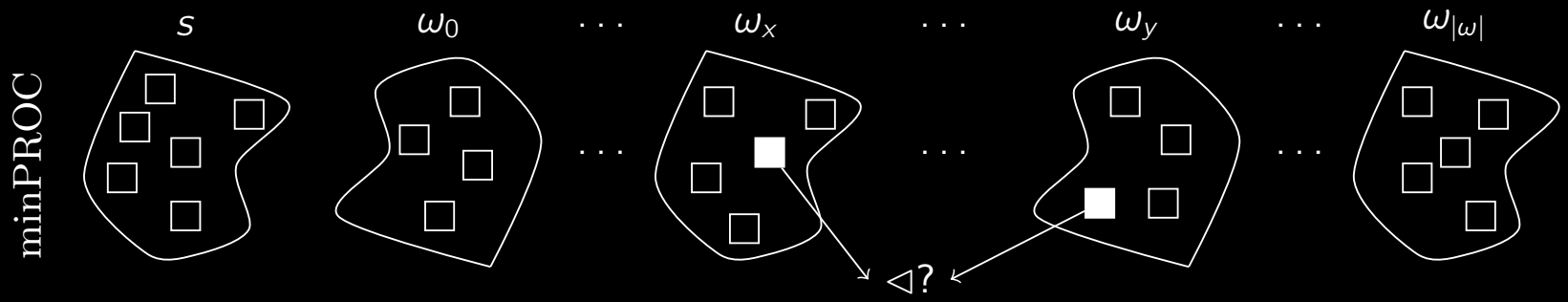
Process Appearance Order Constraints

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

Uncovering Order Constraints

$\mathcal{BS}(a_i \upharpoonright^* a_j) = \emptyset \implies a_j \lhd a_i$

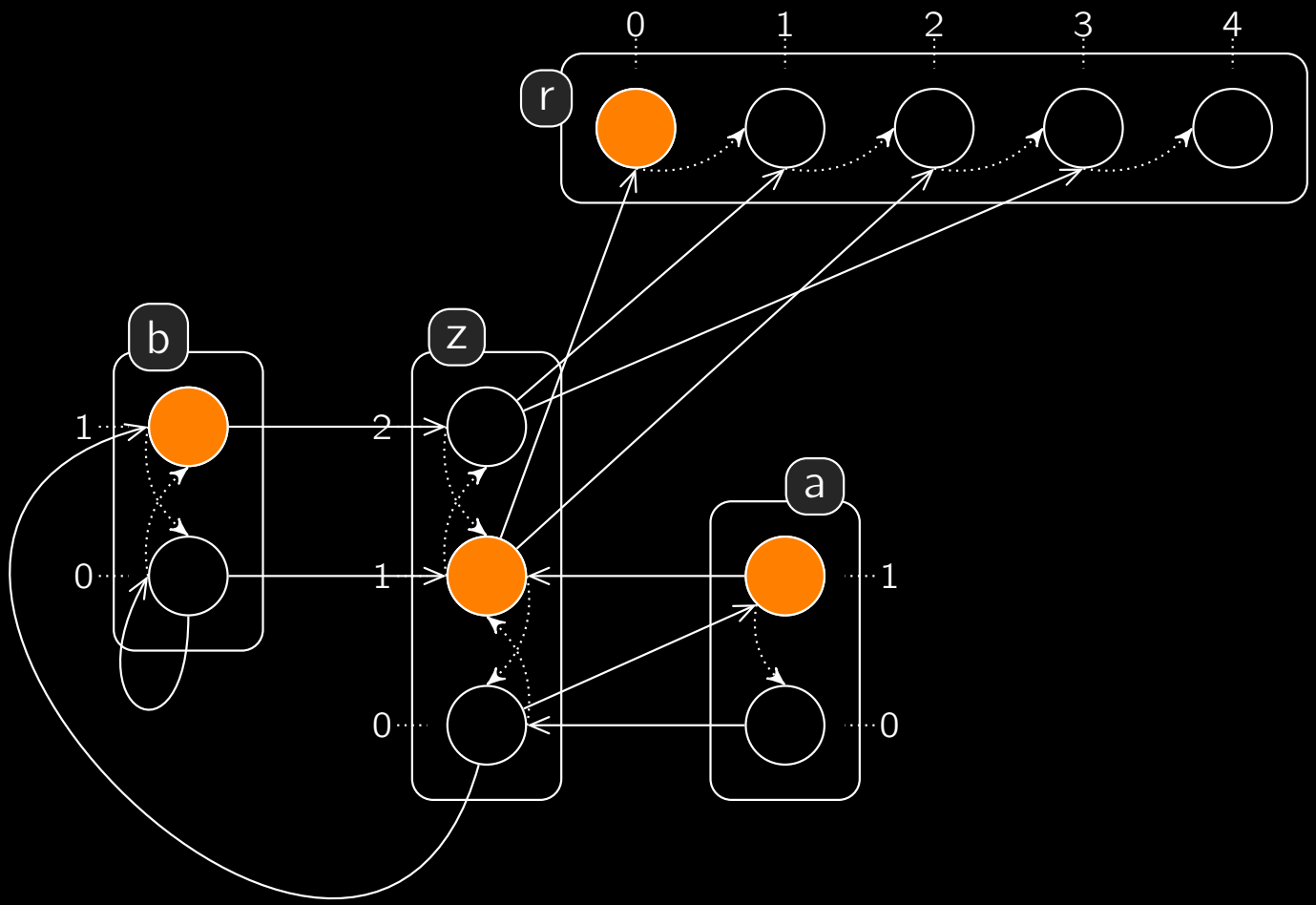
Idea of Over-Approximation



- Context: set of potentially present processes.
- Computing minimal processes at step i needs to compute the maximal context at step $i-1$.
- Work on the complete saturated abstract structure.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

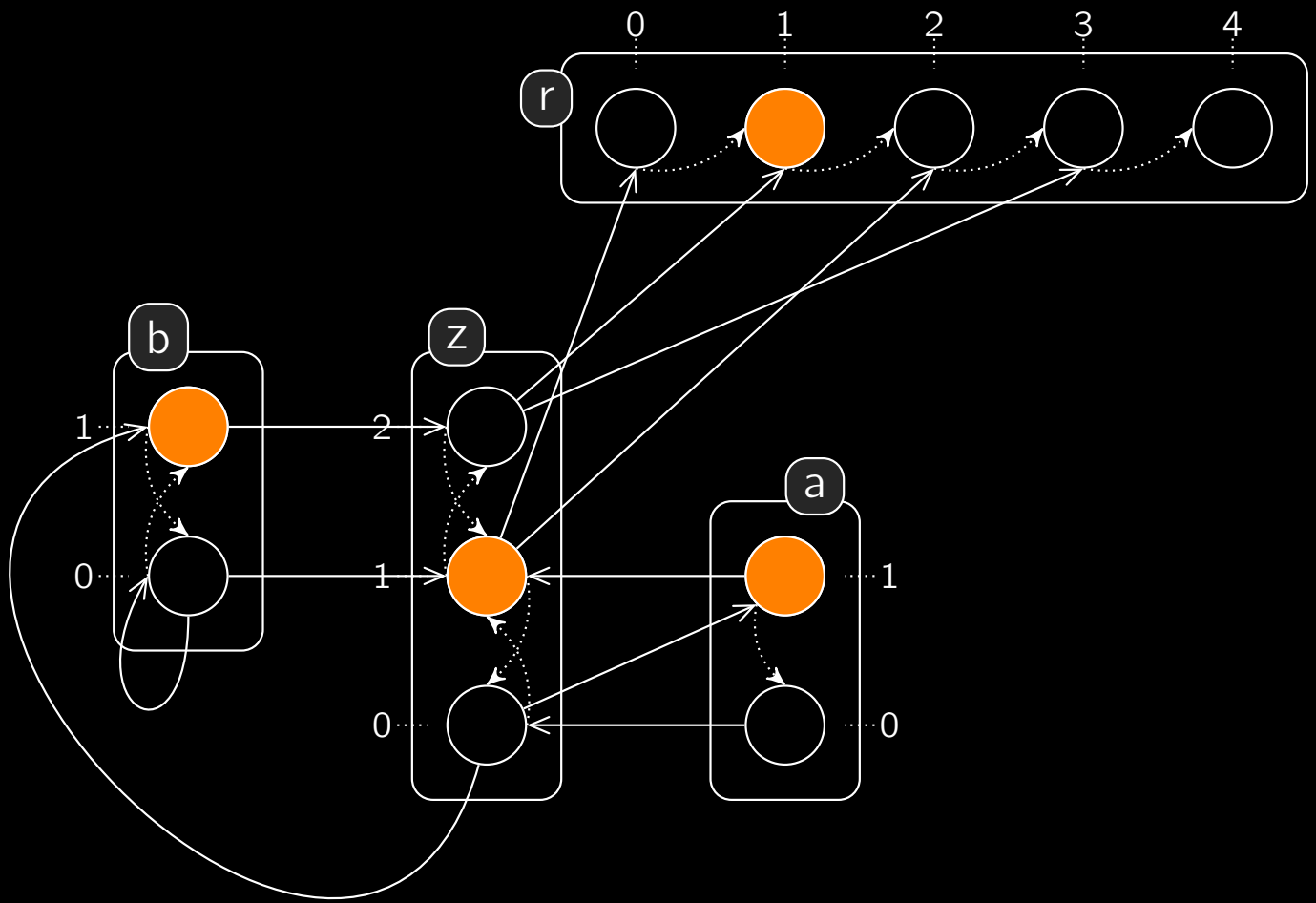


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

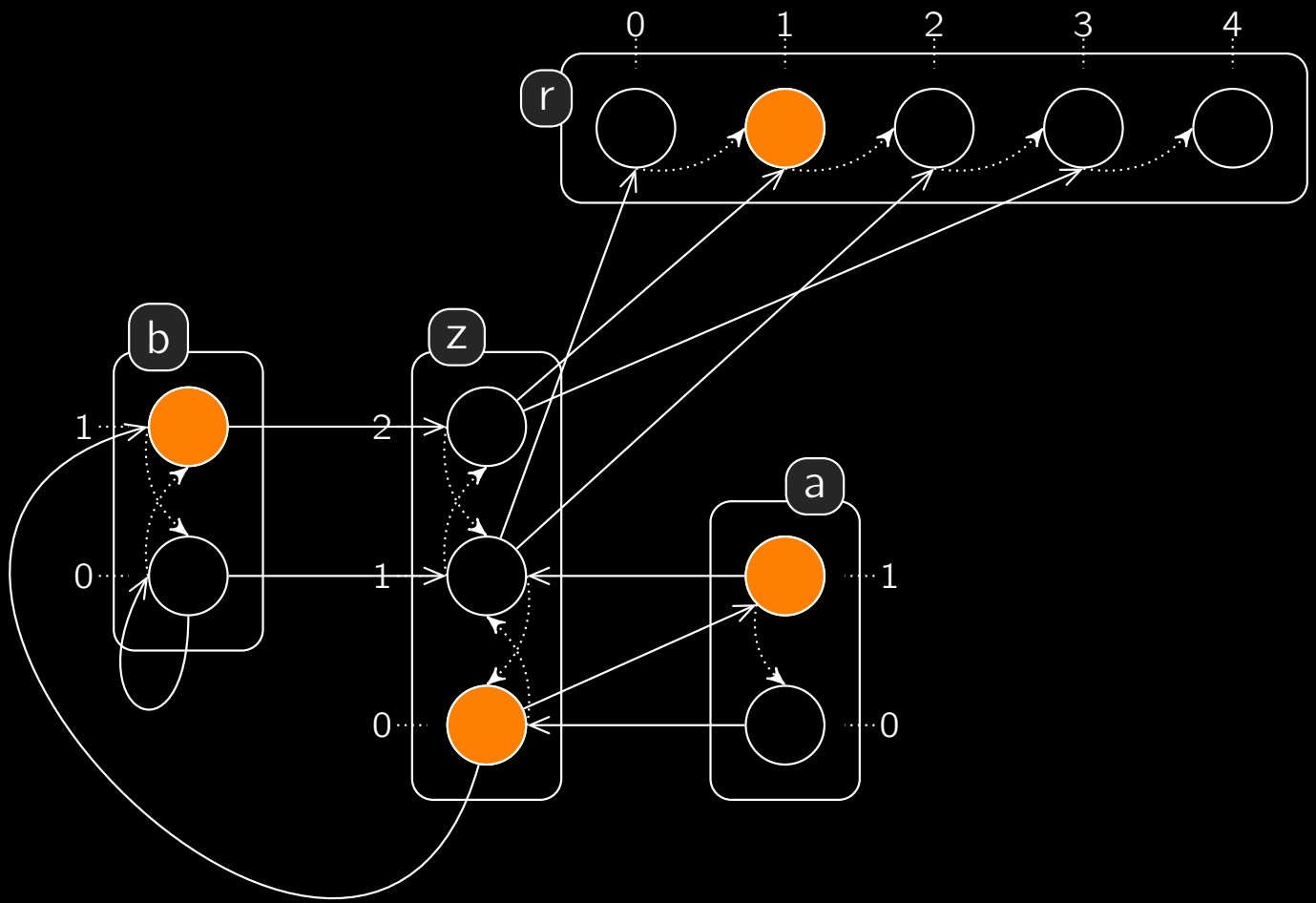


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

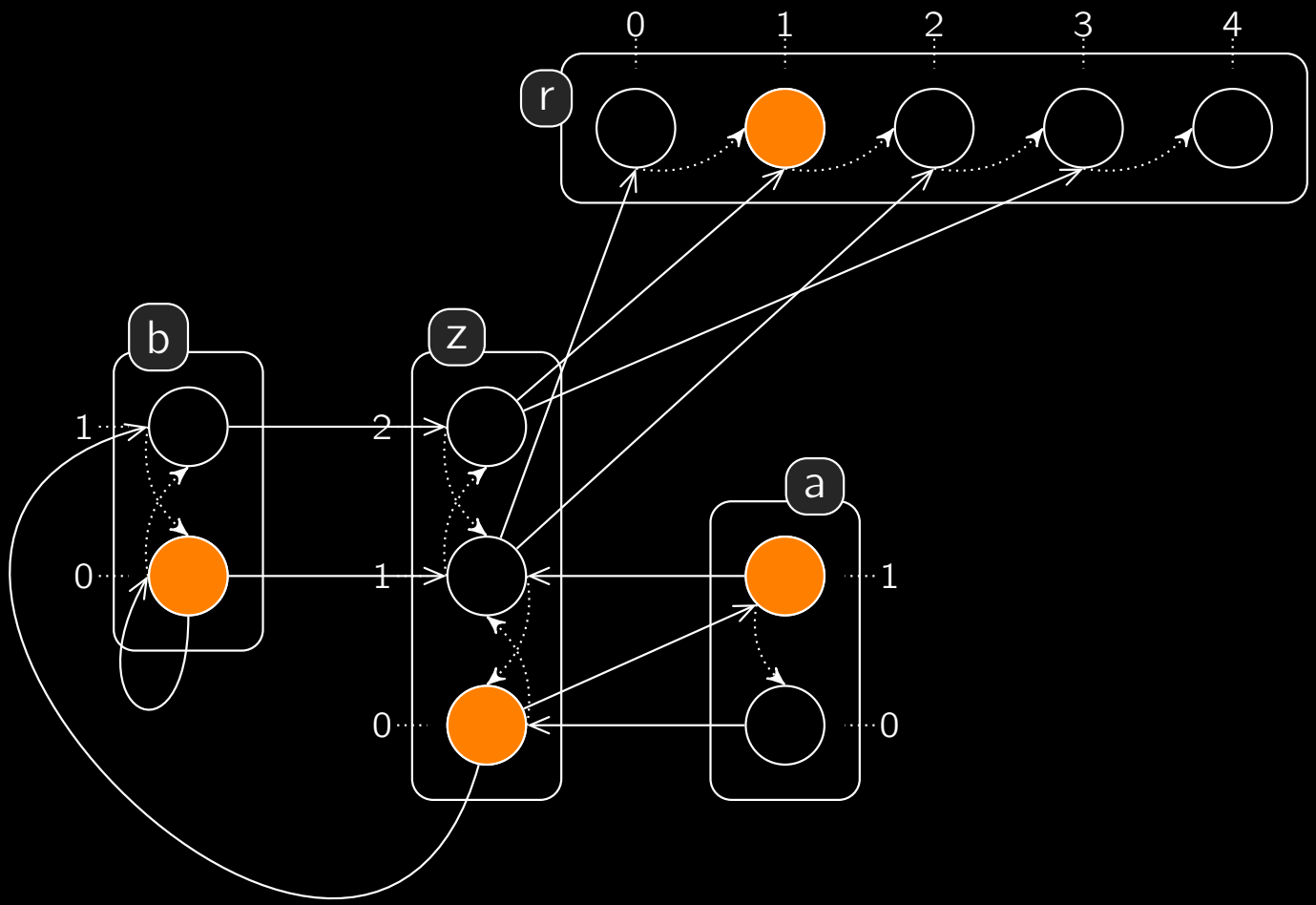


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

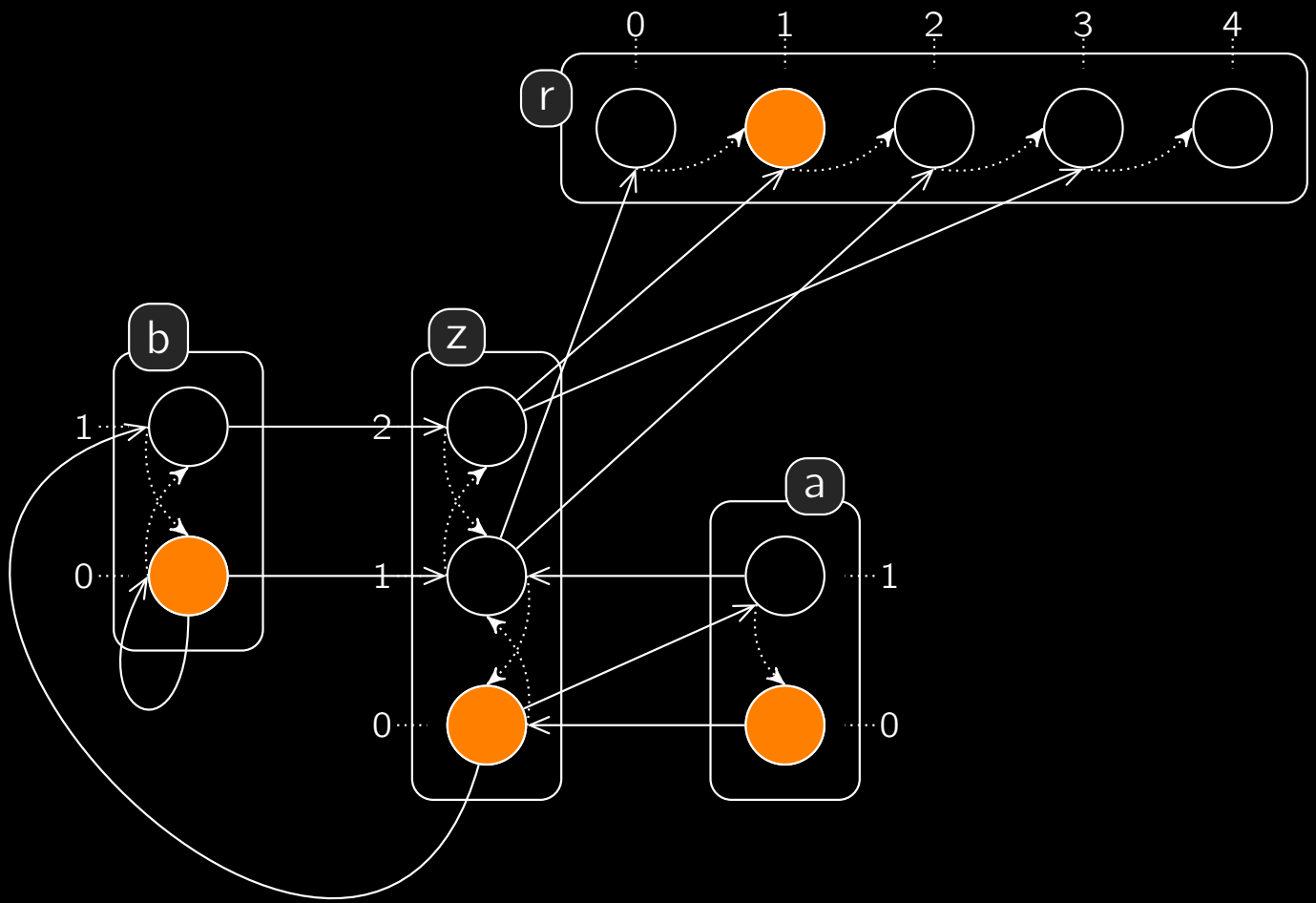


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

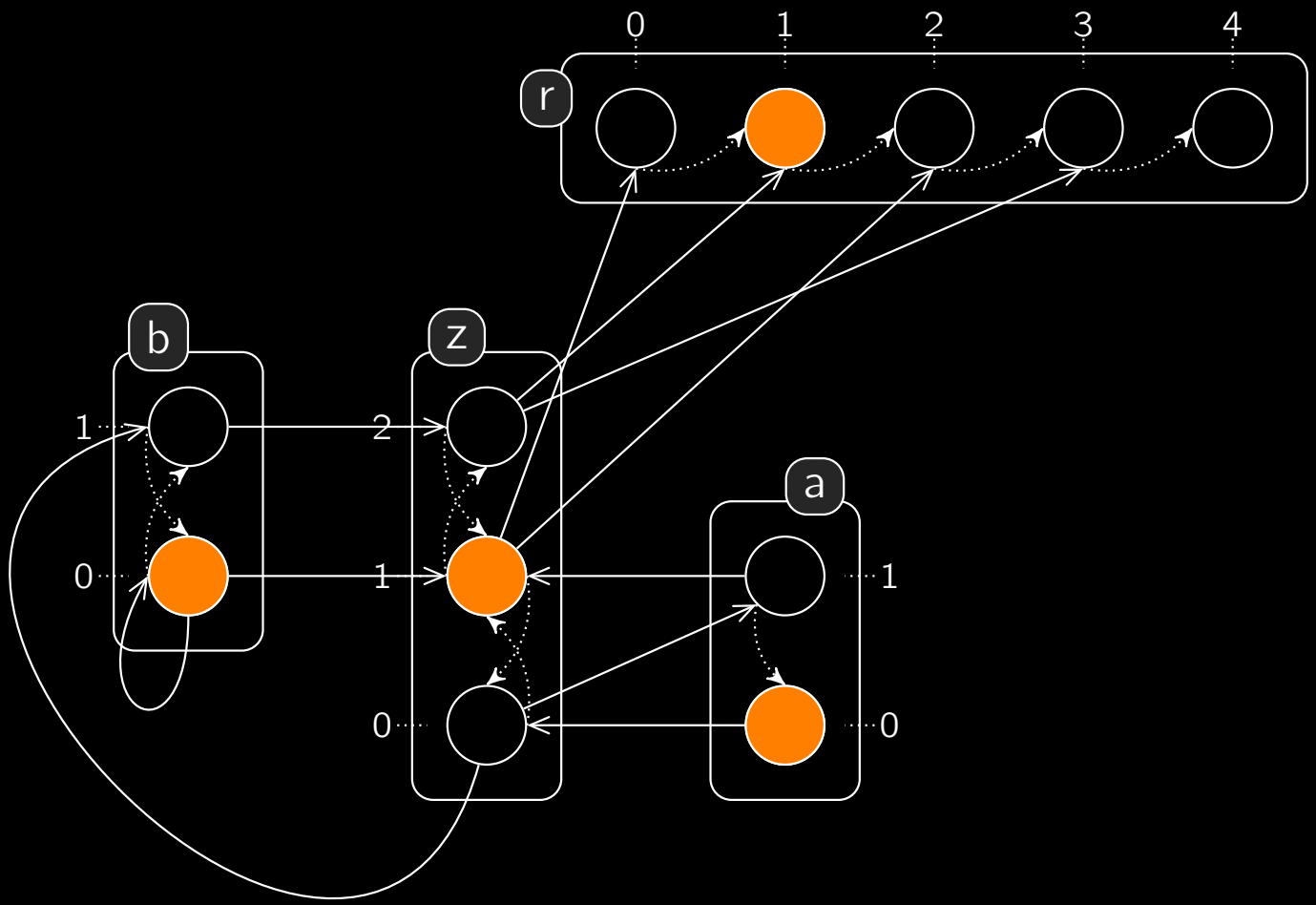


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

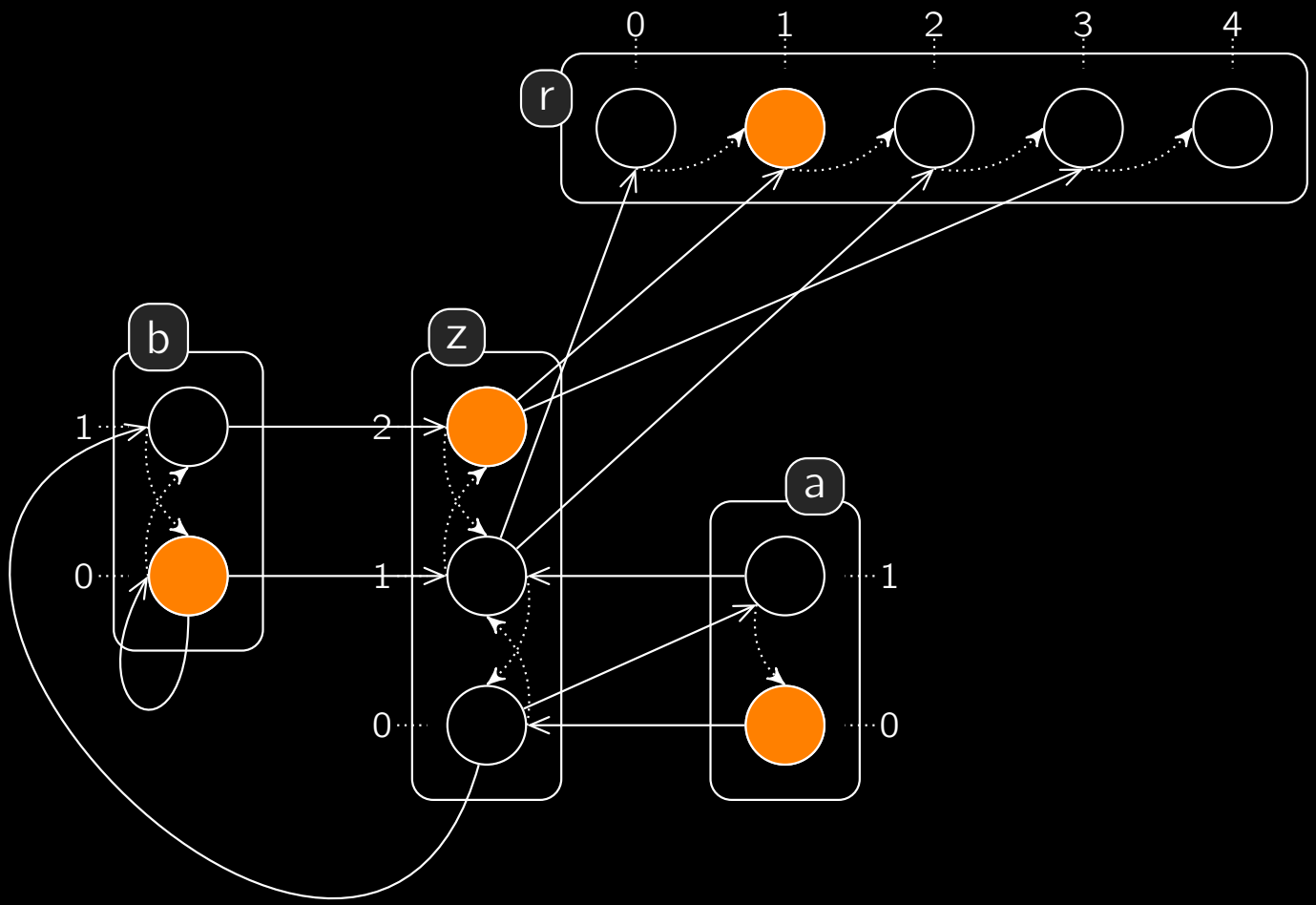


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

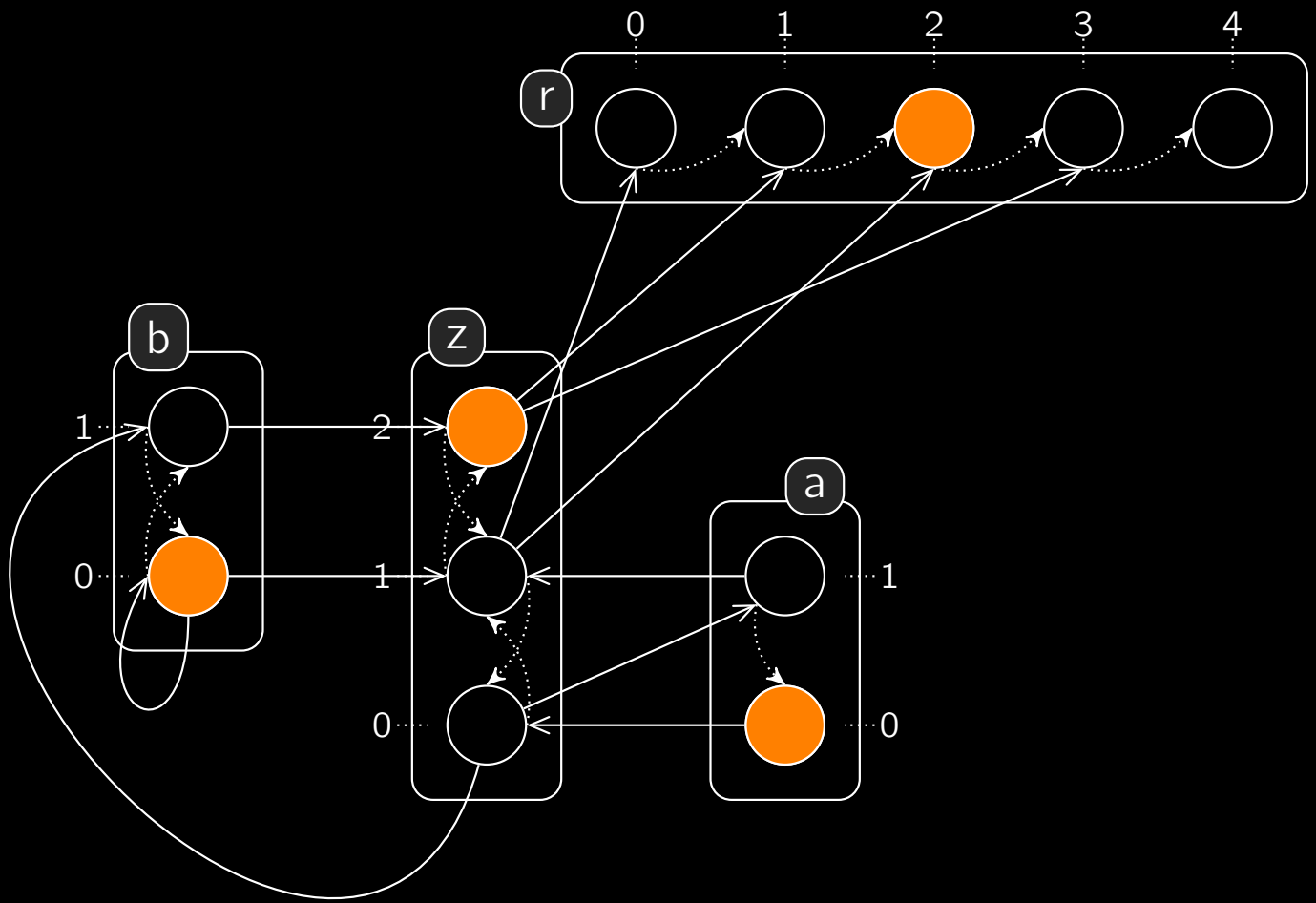


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

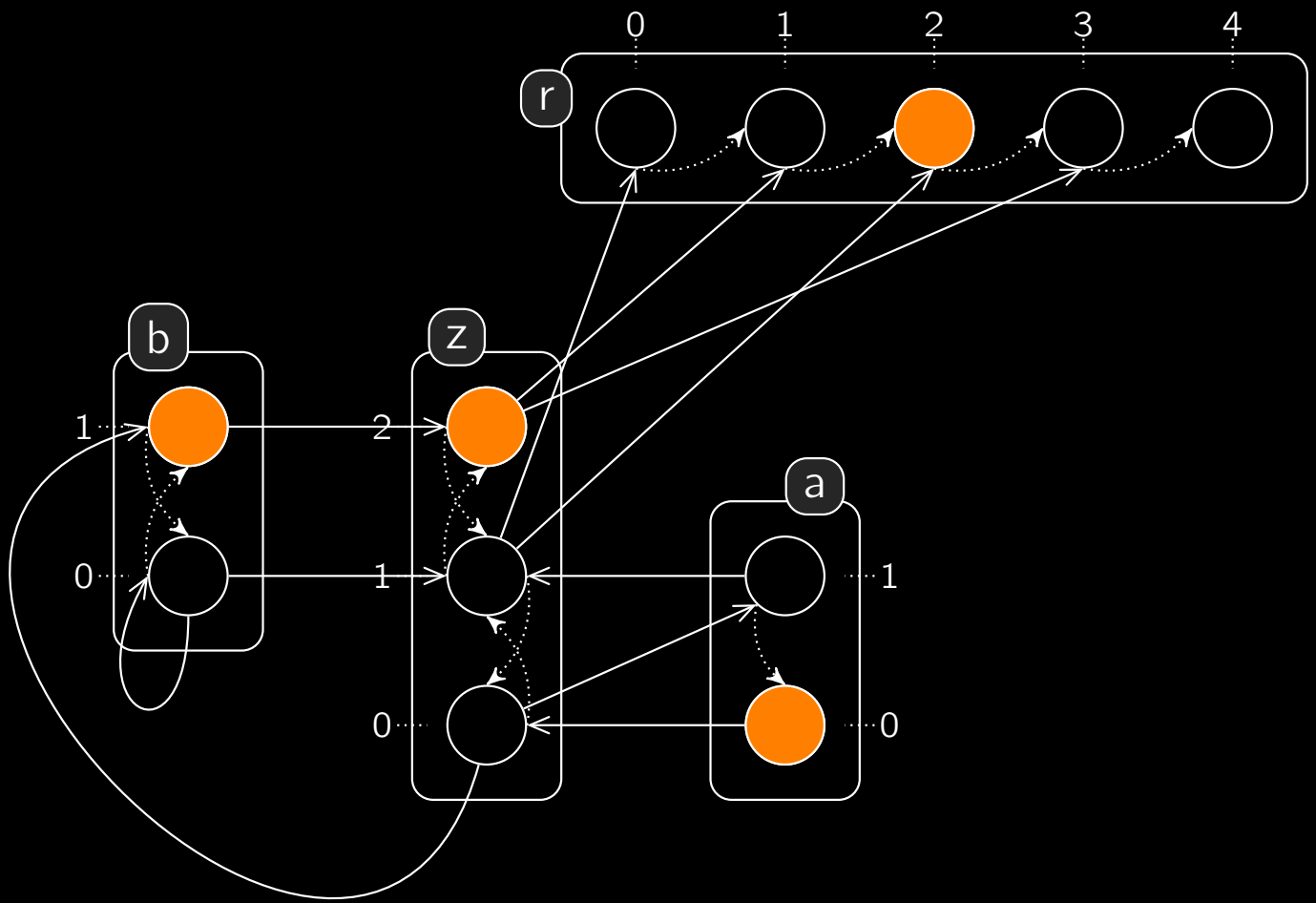


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

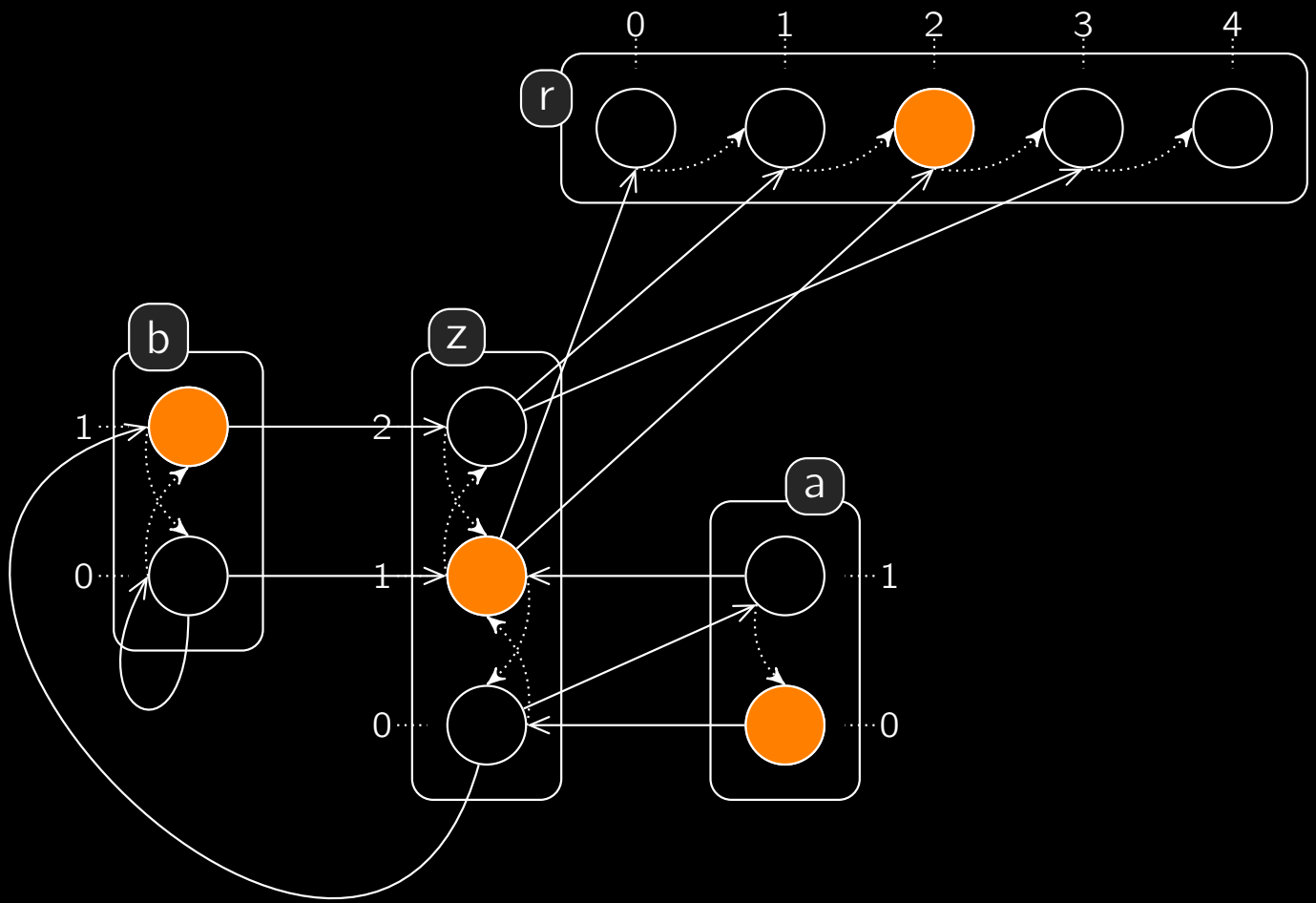


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

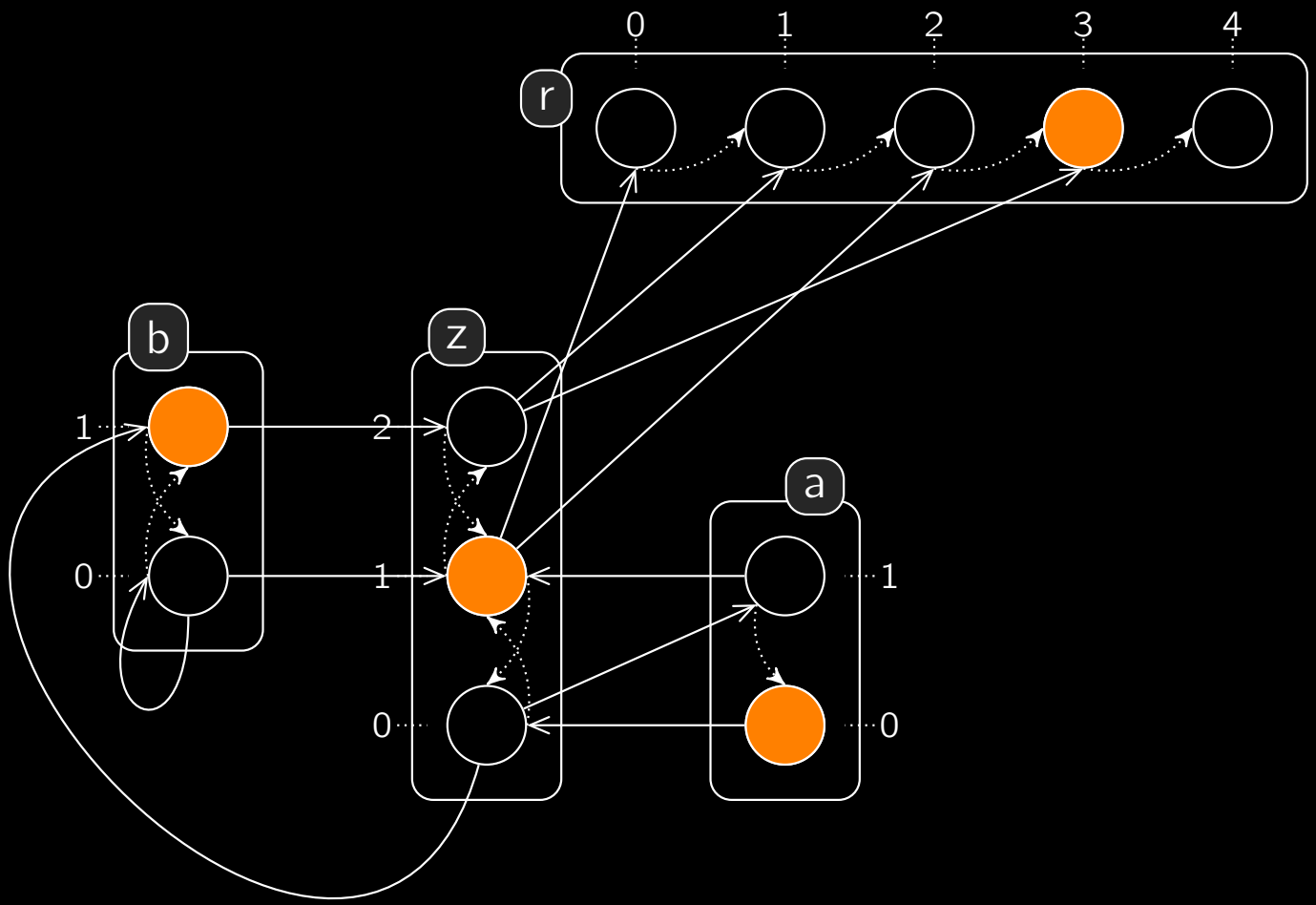


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Toy Example

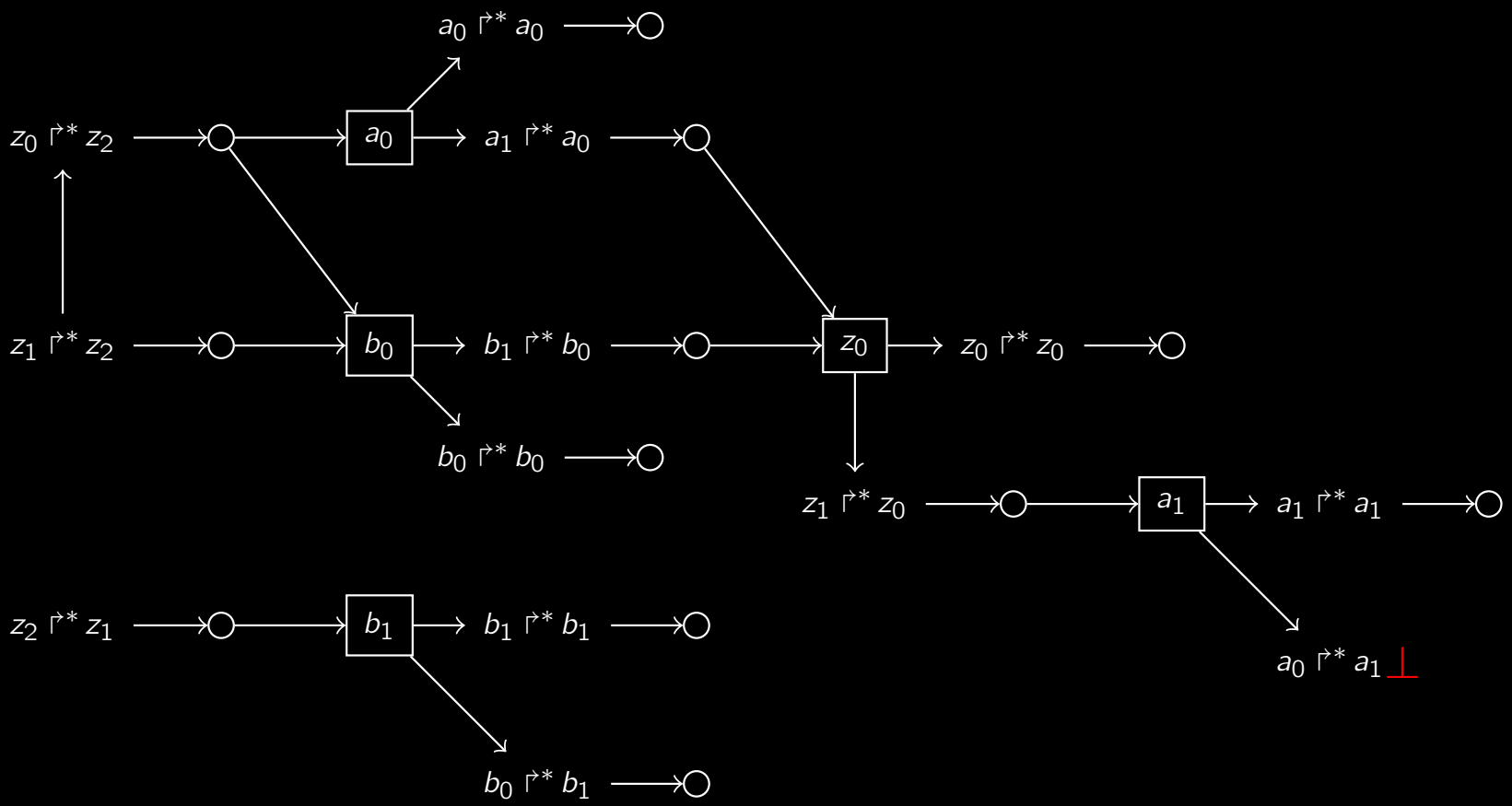


$\mathcal{BS}(r_0 \upharpoonright^* r_4) = \{[z_1 \rightarrow r_0 \upharpoonright r_1, z_2 \rightarrow r_1 \upharpoonright r_2, z_1 \rightarrow r_2 \upharpoonright r_3, z_2 \rightarrow r_3 \upharpoonright r_4]\}$

Objective sequence to check: $z_1 \upharpoonright^* z_2, z_2 \upharpoonright^* z_1, z_1 \upharpoonright^* z_2$.

Loïc Paulevé, Morgan Magnin, Olivier Roux

Saturated Abstract Structure of Example

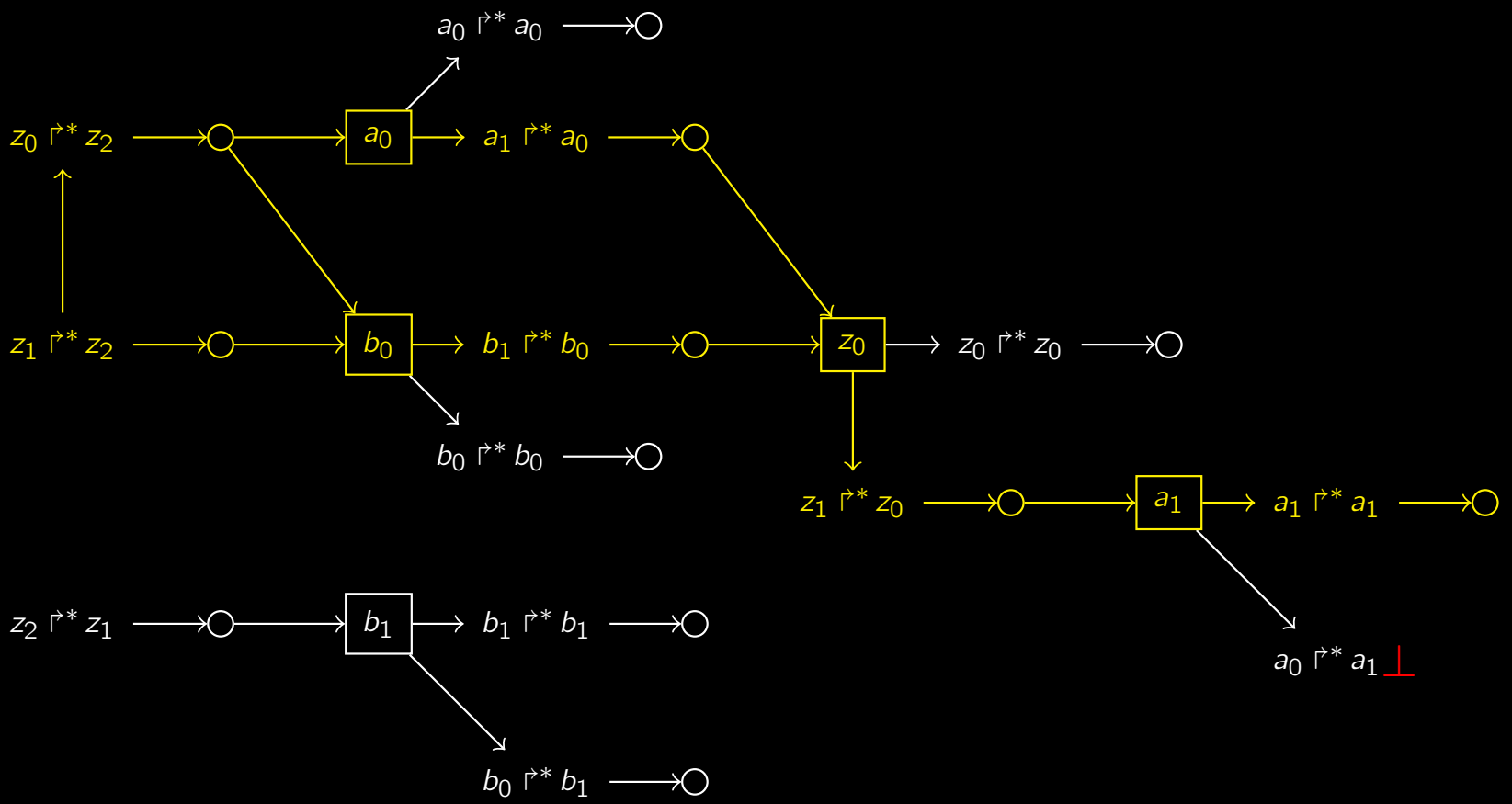


$a_1 \lhd a_0$

	s	$z_1 \Rsh^* z_2$	$z_2 \Rsh^* z_1$	$z_1 \Rsh^* z_2$
minPROC	a_1, b_1, z_1			
endPROC	a_1, b_1, z_1			

Loïc Paulevé, Morgan Magnin, Olivier Roux

Saturated Abstract Structure of Example

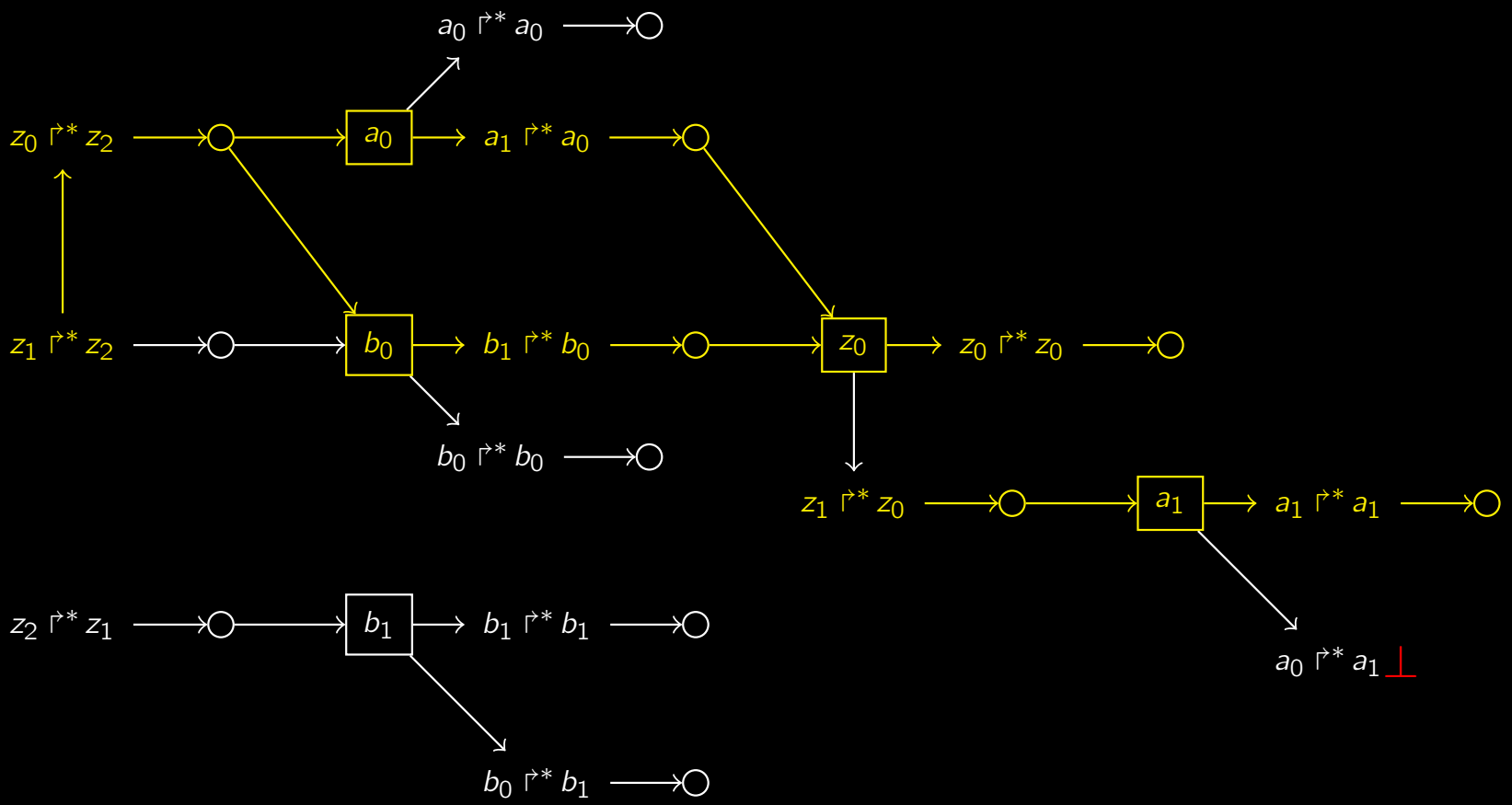


$a_1 \lhd a_0$

	s	$z_1 \Rsh^* z_2$	$z_2 \Rsh^* z_1$	$z_1 \Rsh^* z_2$
minPROC	a_1, b_1, z_1			
endPROC	a_1, b_1, z_1			

Loïc Paulevé, Morgan Magnin, Olivier Roux

Saturated Abstract Structure of Example

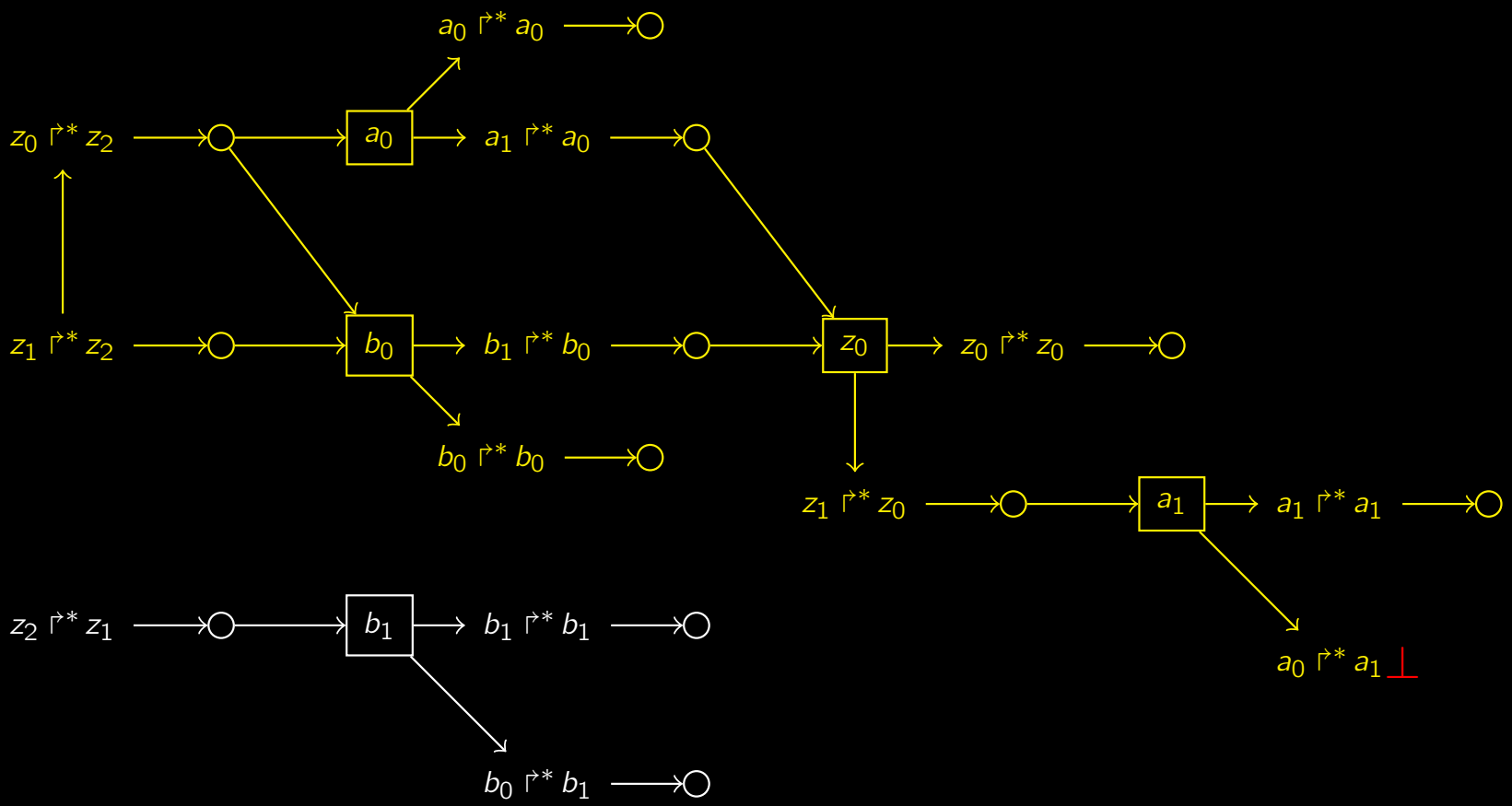


$a_1 \lhd a_0$

	s	$z_1 \Rsh^* z_2$	$z_2 \Rsh^* z_1$	$z_1 \Rsh^* z_2$
minPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_0, z_2		
endPROC	a_1, b_1, z_1			

Loïc Paulevé, Morgan Magnin, Olivier Roux

Saturated Abstract Structure of Example

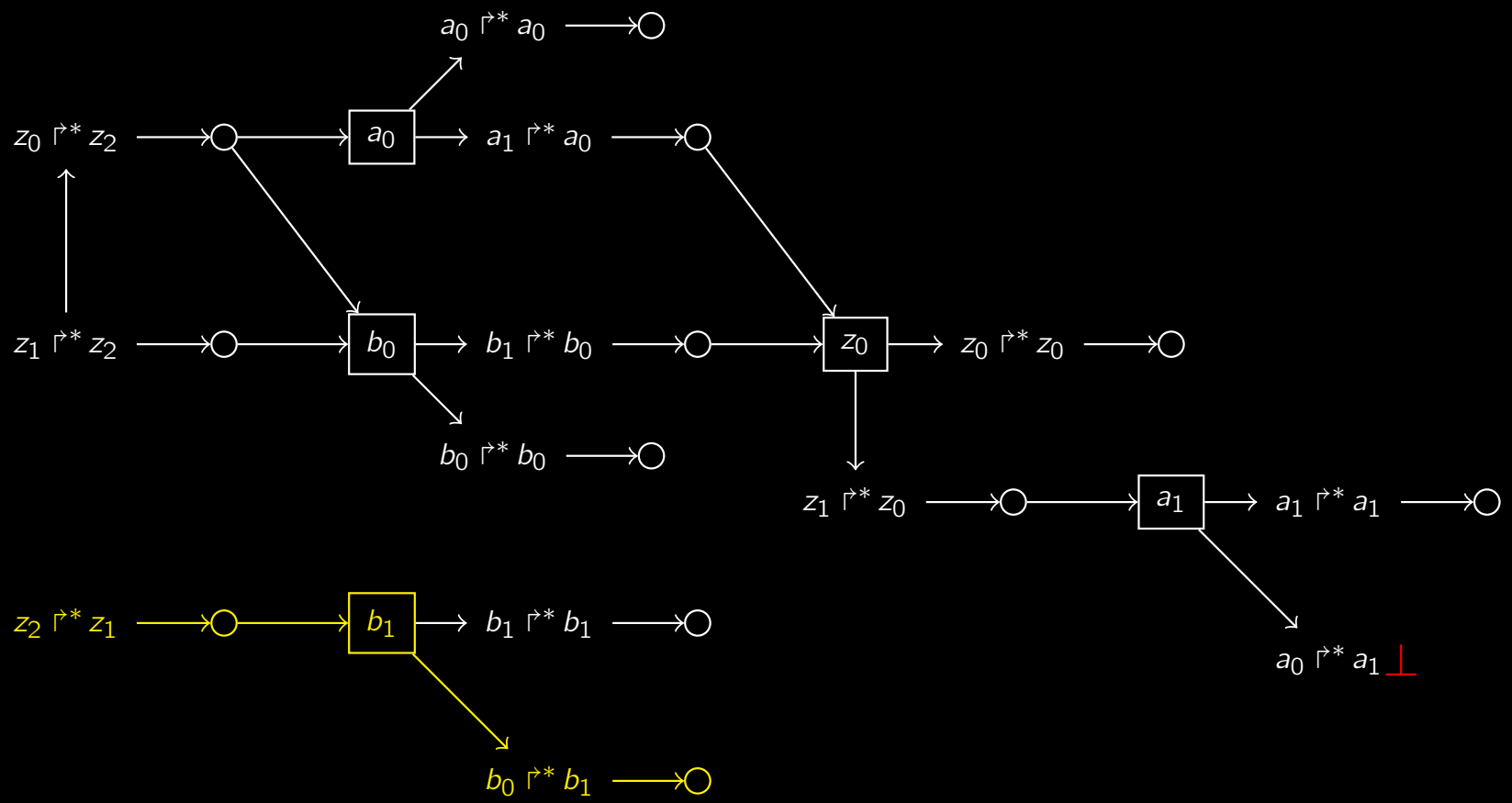


$a_1 \lhd a_0$

	s	$z_1 \Rsh^* z_2$	$z_2 \Rsh^* z_1$	$z_1 \Rsh^* z_2$
minPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_0, z_2		
endPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_2		

Loïc Paulevé, Morgan Magnin, Olivier Roux

Saturated Abstract Structure of Example

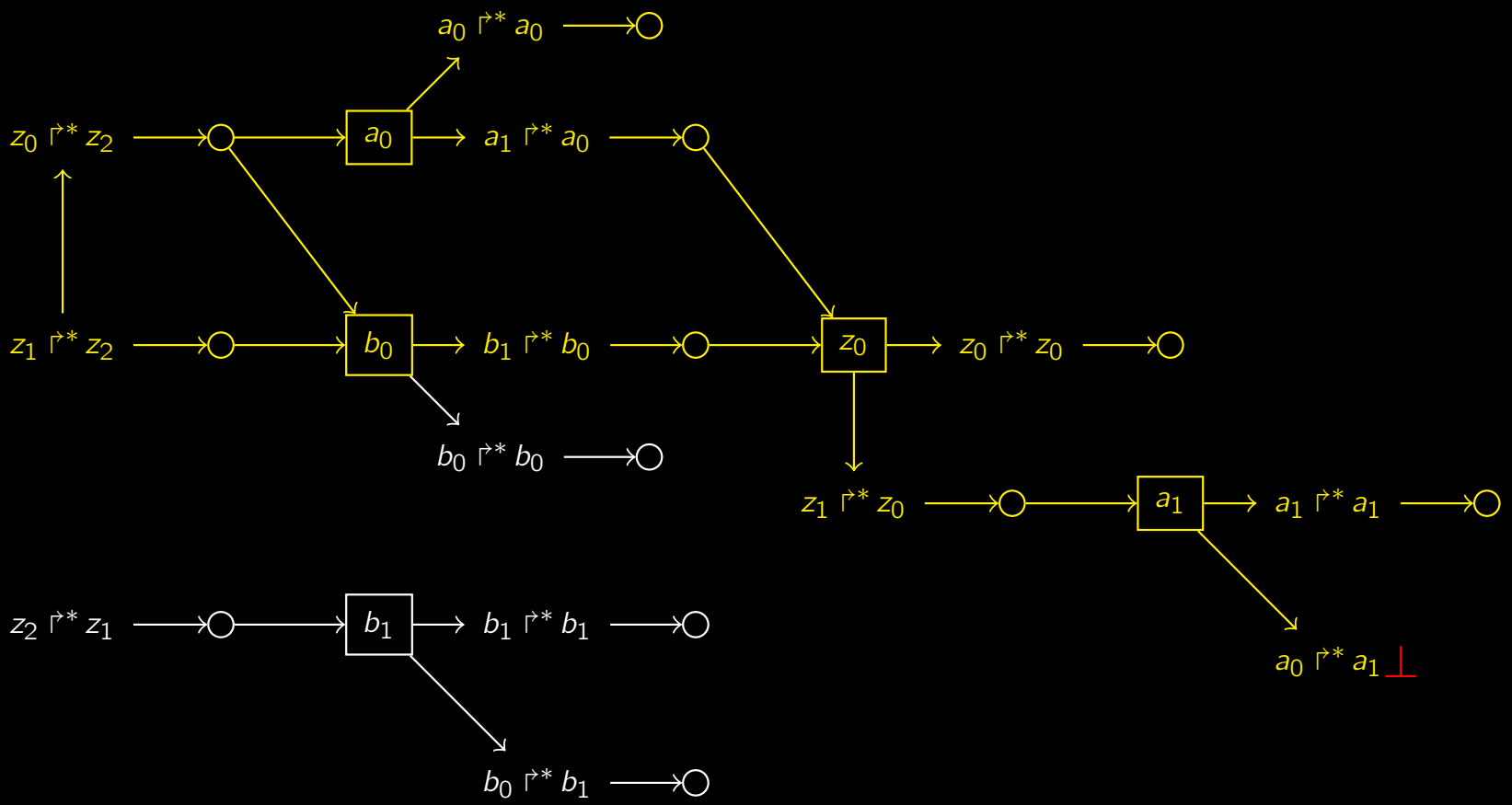


$a_1 \lhd a_0$

	s	$z_1 \lceil^* z_2$	$z_2 \lceil^* z_1$	$z_1 \lceil^* z_2$
minPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_0, z_2	b_1, z_1	
endPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_2	b_1, z_1	

Loïc Paulevé, Morgan Magnin, Olivier Roux

Saturated Abstract Structure of Example



$a_1 \lhd a_0$

	s	$z_1 \Rsh^* z_2$	$z_2 \Rsh^* z_1$	$z_1 \Rsh^* z_2$
minPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_0, z_2	b_1, z_1	a_0, a_1, b_0 z_0, z_2
endPROC	a_1, b_1, z_1	a_0, a_1, b_0 z_2	b_1, z_1	

Loïc Paulevé, Morgan Magnin, Olivier Roux

Pre-Conclusion

Comments

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

Complexities

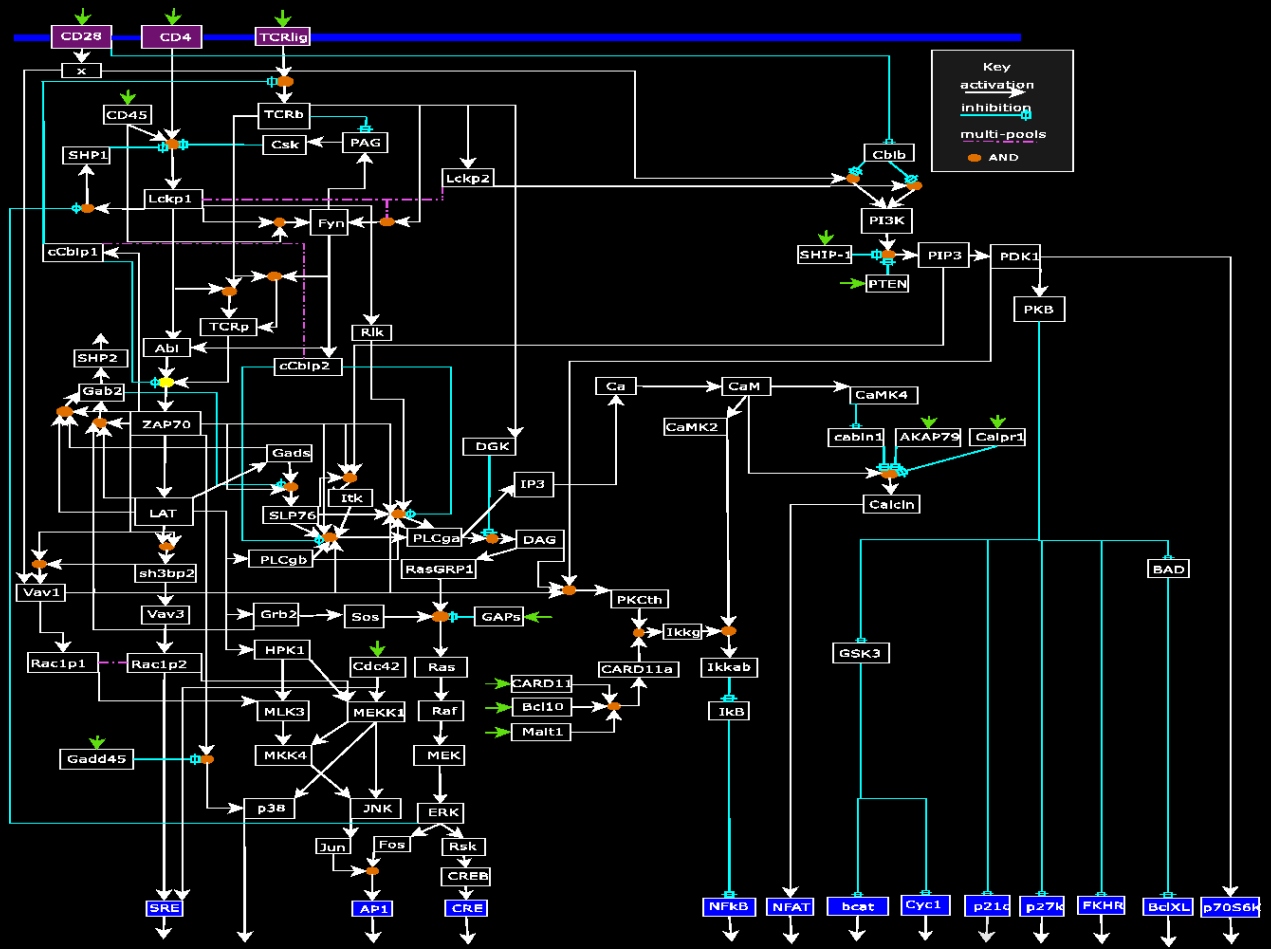
- Computing $\mathcal{BS}\hat{}$ is exponential in the size of the sort;
- other operations are $\approx$ polynomial in the number of processes.

Ideas for reducing inconclusiveness (future work)

- Inconclusiveness: relation cycles and/or excessive saturation.
- Maximum context to improve the first over-approximation (and break cycles).
- Backward strategy?

T-Cell Receptor Signalling Pathway

(94 components)



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

Process Hitting

133 sorts,

448 processes,

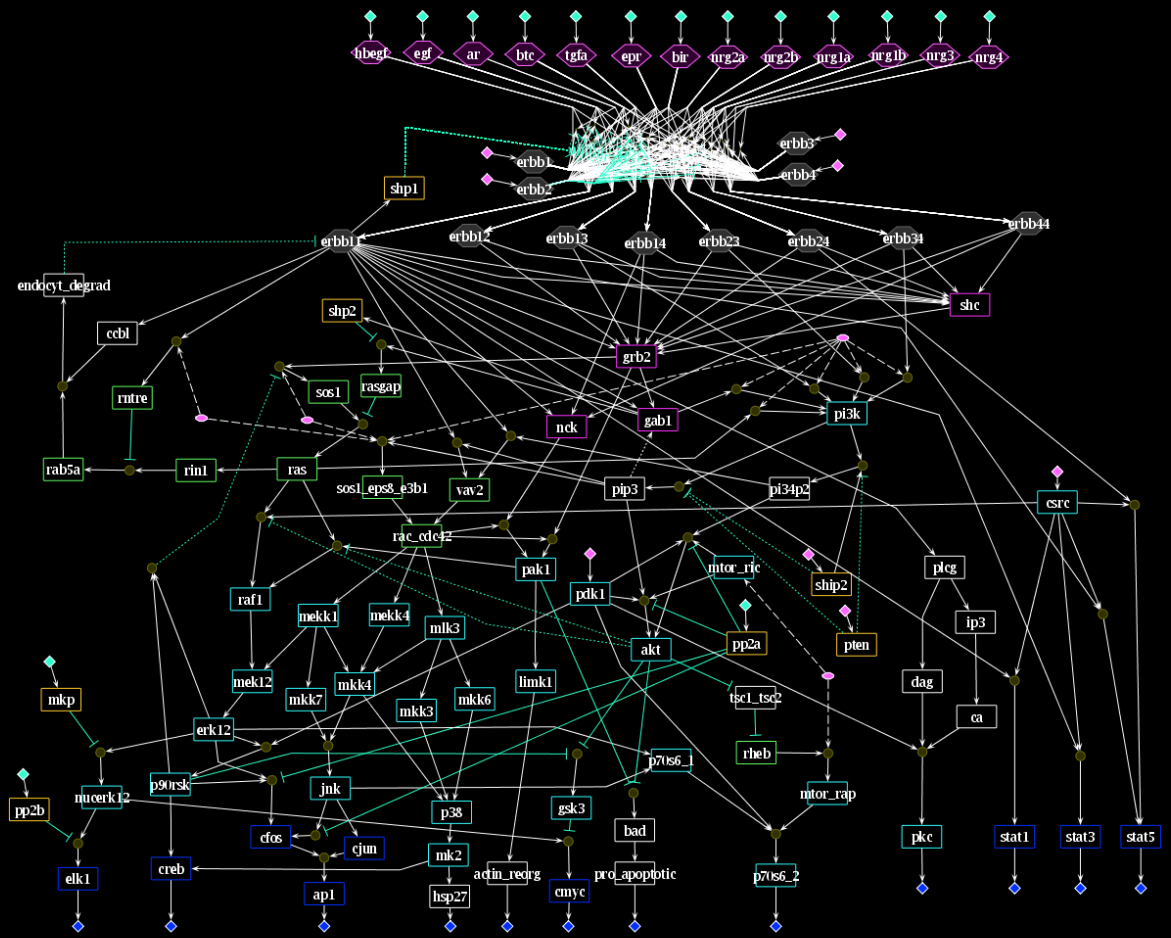
1124 actions:

$\approx 2 \cdot 10^{58}$ states.

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

EGFR/ErbB Signalling

(104 components)



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

Process Hitting

193 sorts,

748 processes,

2356 actions:

$\approx 2 \cdot 10^{96}$ states.

Reachability analysis always conclusive; around 0.05*s* (compared to *libddd*: out of memory).

Loïc Paulevé, Morgan Magnin, Olivier Roux

Conclusion and Outlook

Conclusion

- Efficient over- and under-approximation of process reachability decision;
- static analysis and abstract interpretation from the model;
- promising scalability for BRNs analysis.

Bleeding-edge results

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

On-going work

- Extension to the Process Hitting with Priorities.
- Two classes of actions: instantaneous and non-instantaneous.
- Towards quantitative analysis.