KUBIC-NII Joint Seminar on Bioinformatics 2014

Perturbations and Recovery Costs in Biological Regulatory Networks with Process Hitting

Maxime FOLSCHETTE

MeForBio / IRCCyN / École centrale de Nantes (Nantes, France)
maxime.folschette@irccyn.ec-nantes.fr
http://maxime.folschette.name/

Ongoing work with: **Morgan MAGNIN** and **Katsumi INOUE**

2014/11/17

Studying the Perturbations of a Biological Model

Biological models are well-known for being resilient

- Alternative pathways
- Restoration of oscillations

Studying the Perturbations of a Biological Model

Biological models are well-known for being resilient

- Alternative pathways
- Restoration of oscillations

Observe or measure this in qualitative models:

- Running the models → slow and inefficient
- Model checking → requires powerful methods
- Resilience times → requires timing data
- Observation of specific characteristics → **impact degree**

Studying the Perturbations of a Biological Model

Biological models are well-known for being resilient

- Alternative pathways
- Restoration of oscillations

Observe or measure this in qualitative models:

- Running the models → slow and inefficient
- Model checking → requires powerful methods
- Resilience times → requires timing data
- Observation of specific characteristics → **impact degree**

Refine this analysis with new model checking methods:

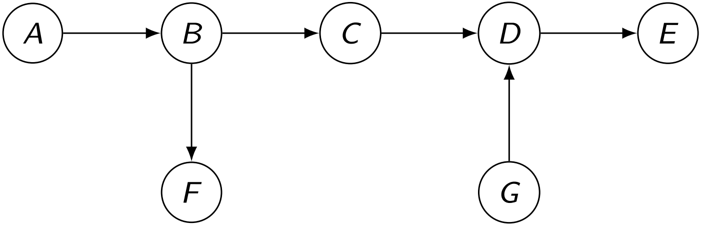
- The Process Hitting framework
- Efficient **reachability analysis**
- Finer study of the perturbations

Impact Degree

[Jiang, Tamura, Ching, Akutsu in *Communications and Computer Sciences*, 2013]

Reaction networks = set of species consumed and produced by reactions

- Reaches an equilibrium state

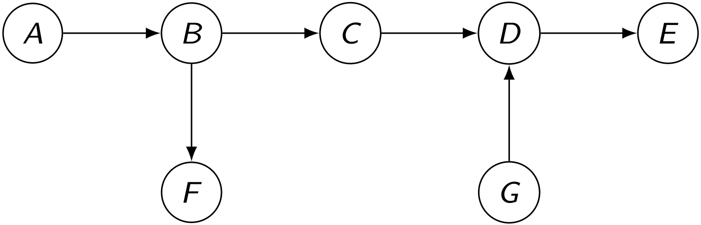


Impact Degree

[Jiang, Tamura, Ching, Akutsu in *Communications and Computer Sciences*, 2013]

Reaction networks = set of species consumed and produced by reactions

- Reaches an equilibrium state



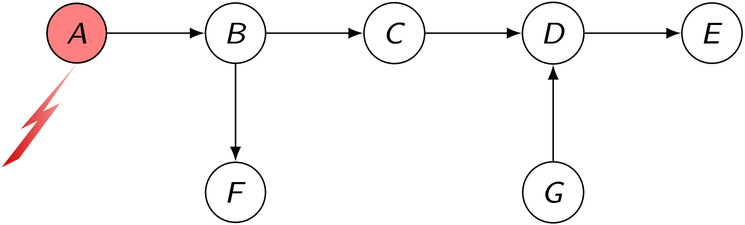
Impact degree of A = number of nodes impacted by a knockout

Impact Degree

[Jiang, Tamura, Ching, Akutsu in *Communications and Computer Sciences*, 2013]

Reaction networks = set of species consumed and produced by reactions

- Reaches an equilibrium state



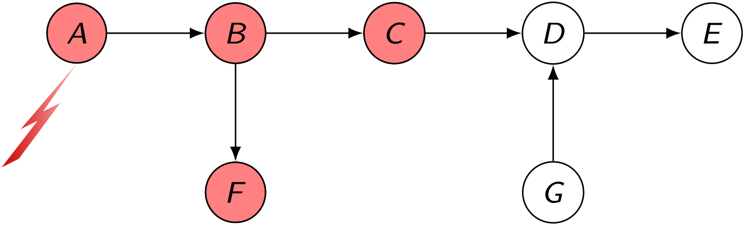
Impact degree of A = number of nodes impacted by a knockout

Impact Degree

[Jiang, Tamura, Ching, Akutsu in *Communications and Computer Sciences*, 2013]

Reaction networks = set of species consumed and produced by reactions

- Reaches an equilibrium state



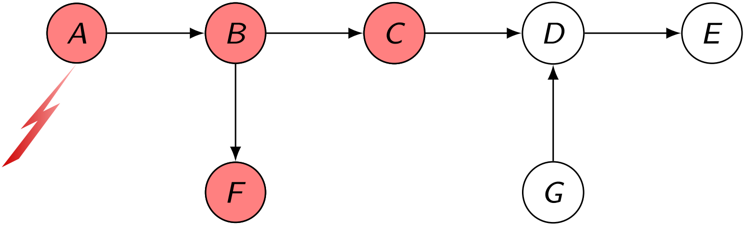
Impact degree of A = number of nodes impacted by a knockout → For A: 4

Impact Degree

[Jiang, Tamura, Ching, Akutsu in *Communications and Computer Sciences*, 2013]

Reaction networks = set of species consumed and produced by reactions

- Reaches an equilibrium state



Impact degree of A = number of nodes impacted by a knockout → For A: 4

- Notion of importance/criticality of a node
- Highlights the resilience of biological systems (alternative paths)

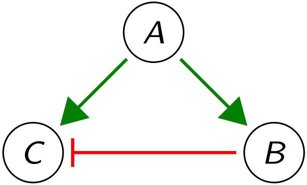
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



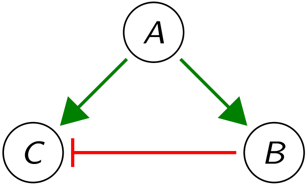
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

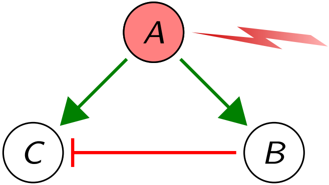
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

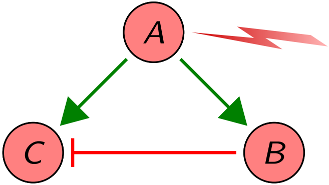
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

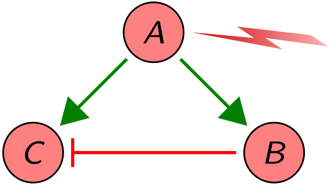
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

- Number of species that are completely turned off → For A: 3

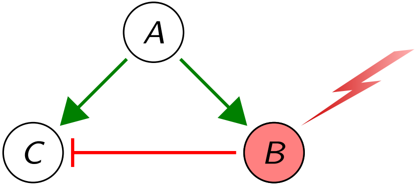
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

- Number of species that are completely turned off → For A: 3

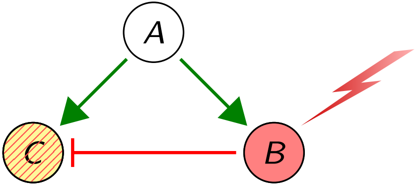
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

- Number of species that are completely turned off → For A: 3

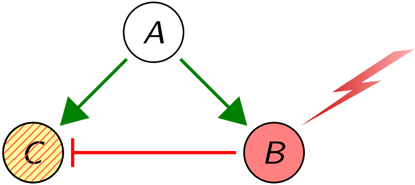
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state



New notion of **impact degree**

- Number of species that are completely turned off → For A: 3
- Number of species whose behavior is modified → For B: $1 + 1 = 2$

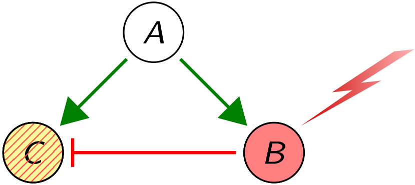
Application to regulation networks

Model from [Comet, Bernot in *Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, 2010]

Regulation networks = set of species regulated by other species

→ The regulating species are not consumed

→ Negative regulations → Not always a steady state

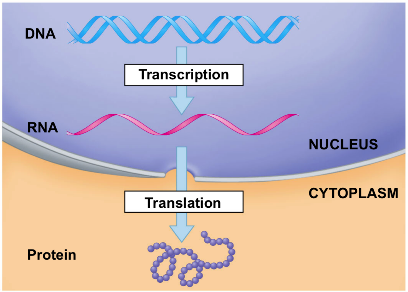


New notion of **impact degree**

- Number of species that are completely turned off → For A: 3
- Number of species whose behavior is modified → For B: $1 + 1 = 2$

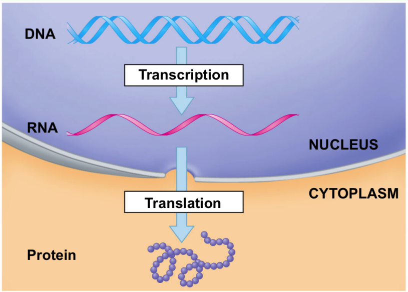
→ Requires a more precise study of the behavior

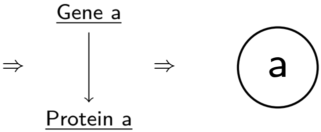
Abstractions of the Representation



© 2012 Pearson Education, Inc.

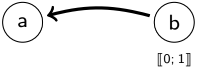
Abstractions of the Representation

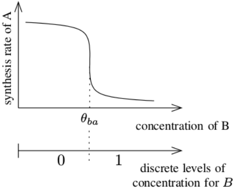




Discretization and Asynchronism

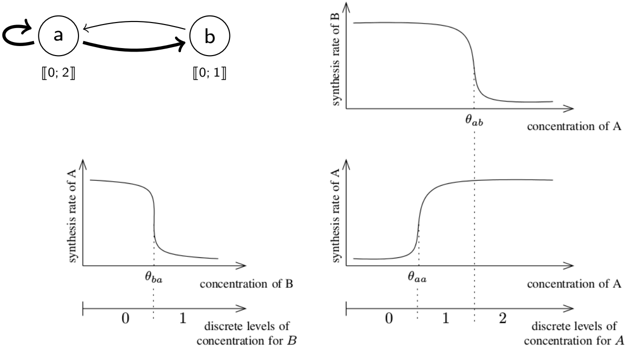
[Richard, Comet, Bernot (tutorial), 2008]





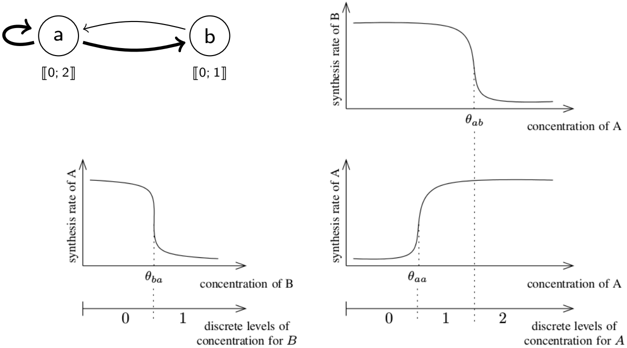
Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]



Discretization and Asynchronism

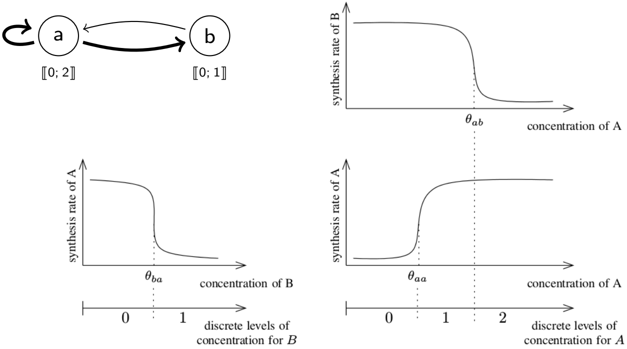
[Richard, Comet, Bernot (tutorial), 2008]



- Unknown real values of concentrations or continuous activity levels
  → Abstracted as thresholds or **discrete levels**

Discretization and Asynchronism

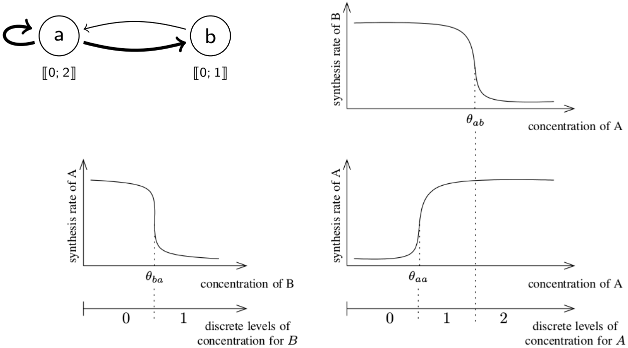
[Richard, Comet, Bernot (tutorial), 2008]



- Unknown real values of concentrations or continuous activity levels
  → Abstracted as thresholds or **discrete levels**
- Continuous variations of the real values
  → **Unitary** dynamics

Discretization and Asynchronism

[Richard, Comet, Bernot (tutorial), 2008]

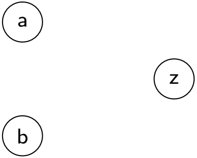


- Unknown real values of concentrations or continuous activity levels
  $\rightarrow$ Abstracted as thresholds or **discrete levels**
- Continuous variations of the real values
  $\rightarrow$ **Unitary** dynamics
- Simultaneous crossings of two thresholds never occurs
  $\rightarrow$ **Asynchronous** dynamics

Discrete Networks / Thomas Modeling

[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

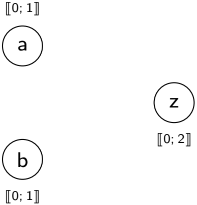
- A set of components $N = \{a, b, z\}$



Discrete Networks / Thomas Modeling

[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $z \in \mathbb{F}^z = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$



Discrete Networks / Thomas Modeling

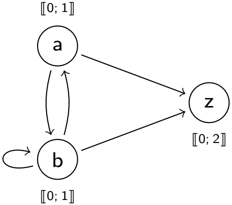
[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $z \in \mathbb{F}^z = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$
- An evolution function for each component $f^z : \mathbb{F} \to \mathbb{F}^z$

b	$f^a(b)$
0	**1**
1	**0**

a	b	$f^b(a, b)$
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

a	b	$f^z(a, b)$
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**2**



Discrete Networks / Thomas Modeling

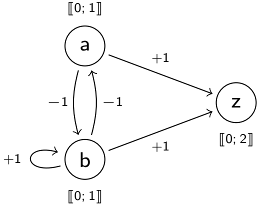
[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $z \in \mathbb{F}^z = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$
- An evolution function for each component $f^z : \mathbb{F} \to \mathbb{F}^z$
- Signs and thresholds on the edges $a \xrightarrow{+1} z$

b	$f^a(b)$
0	**1**
1	**0**

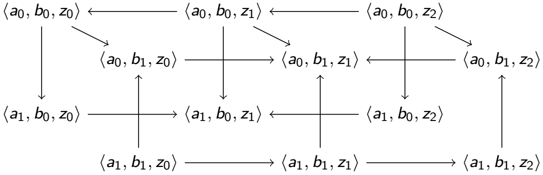
a	b	$f^b(a, b)$
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

a	b	$f^z(a, b)$
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**2**



Analysis of Thomas Modeling

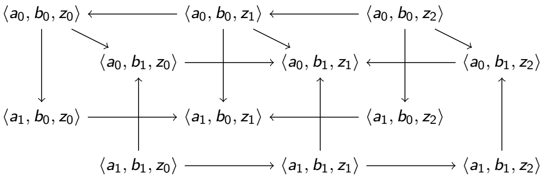
The State graph is computed in a unitary and asynchronous fashion


$\langle a_0, b_0, z_0 \rangle$
$\langle a_0, b_0, z_1 \rangle$
$\langle a_0, b_0, z_2 \rangle$
$\langle a_0, b_1, z_0 \rangle$
$\langle a_0, b_1, z_1 \rangle$
$\langle a_0, b_1, z_2 \rangle$
$\langle a_1, b_0, z_0 \rangle$
$\langle a_1, b_0, z_1 \rangle$
$\langle a_1, b_0, z_2 \rangle$
$\langle a_1, b_1, z_0 \rangle$
$\langle a_1, b_1, z_1 \rangle$
$\langle a_1, b_1, z_2 \rangle$

→ **Exponential** size in the number of components

Analysis of Thomas Modeling

The State graph is computed in a unitary and asynchronous fashion


$\langle a_0, b_0, z_0 \rangle$
$\langle a_0, b_0, z_1 \rangle$
$\langle a_0, b_0, z_2 \rangle$
$\langle a_0, b_1, z_0 \rangle$
$\langle a_0, b_1, z_1 \rangle$
$\langle a_0, b_1, z_2 \rangle$
$\langle a_1, b_0, z_0 \rangle$
$\langle a_1, b_0, z_1 \rangle$
$\langle a_1, b_0, z_2 \rangle$
$\langle a_1, b_1, z_0 \rangle$
$\langle a_1, b_1, z_1 \rangle$
$\langle a_1, b_1, z_2 \rangle$

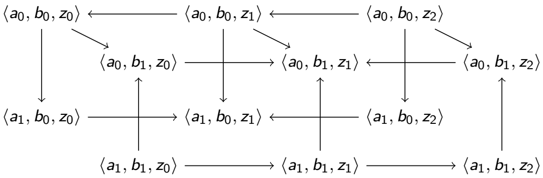
→ **Exponential** size in the number of components

Some works link the structure of the model to some dynamic properties:

- **Thomas' conjectures** (conditions for multi-stationarity or sustained oscillations)
    - Boolean case: [Remy, Ruet, Thieffry in *Advances in Applied Mathematics*, 2008]
    - Multivalued case: [Richard, Comet in *Discrete Applied Mathematics*, 2007]

Analysis of Thomas Modeling

The State graph is computed in a unitary and asynchronous fashion


$\langle a_0, b_0, z_0 \rangle$
$\langle a_0, b_0, z_1 \rangle$
$\langle a_0, b_0, z_2 \rangle$
$\langle a_0, b_1, z_0 \rangle$
$\langle a_0, b_1, z_1 \rangle$
$\langle a_0, b_1, z_2 \rangle$
$\langle a_1, b_0, z_0 \rangle$
$\langle a_1, b_0, z_1 \rangle$
$\langle a_1, b_0, z_2 \rangle$
$\langle a_1, b_1, z_0 \rangle$
$\langle a_1, b_1, z_1 \rangle$
$\langle a_1, b_1, z_2 \rangle$

$\rightarrow$ **Exponential** size in the number of components

Some works link the structure of the model to some dynamic properties:

- **Thomas' conjectures** (conditions for multi-stationarity or sustained oscillations)
  - Boolean case: [Remy, Ruet, Thieffry in *Advances in Applied Mathematics*, 2008]
  - Multivalued case: [Richard, Comet in *Discrete Applied Mathematics*, 2007]

But reachability properties require to compute the whole state graph:
Example: From the initial state $(a, b, z) = (0, 0, 0)$, is it possible to reach $z = 2$?

- **Temporal logics**
  - CTL: [Bernot, Comet, Richard, Guespin in *Journal of Theoretical Biology*, 2004]
  - LTL: [Ito, Izumi, Hagihara, Yonezaki in *BioInformatics and BioEngineering*, 2010]

Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

The **Process Hitting** is:

- A recent formalism well-adapted to the modeling of BRNs
- An **atomistic, qualitative and asynchronous** modeling (explicit & discrete expression levels)
- **Simple but powerful** dynamics (constraints on the form of actions)

Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

The **Process Hitting** is:

- A recent formalism well-adapted to the modeling of BRNs
- An **atomistic, qualitative and asynchronous** modeling (explicit & discrete expression levels)
- **Simple but powerful** dynamics (constraints on the form of actions)

Previously developed tools:

- **Reachability analysis** by abstract interpretation
- Fixed points enumeration
- Stochastic parameters

Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

The **Process Hitting** is:

- A recent formalism well-adapted to the modeling of BRNs
- An **atomistic, qualitative and asynchronous** modeling (explicit & discrete expression levels)
- **Simple but powerful** dynamics (constraints on the form of actions)

Previously developed tools:

- **Reachability analysis** by abstract interpretation
- Fixed points enumeration
- Stochastic parameters

→ The **reachability analysis** is very efficient (polynomial time)
→ The Process Hitting is also well-adapted to study **large BRNs**

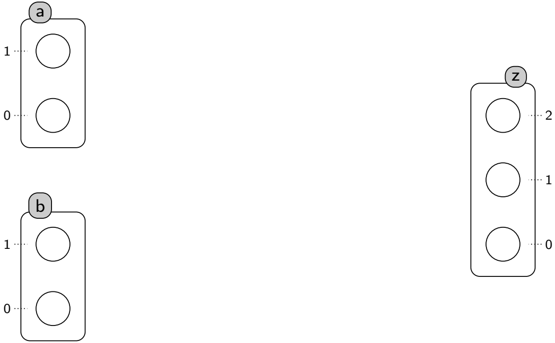
Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Sorts: components a, b, z

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

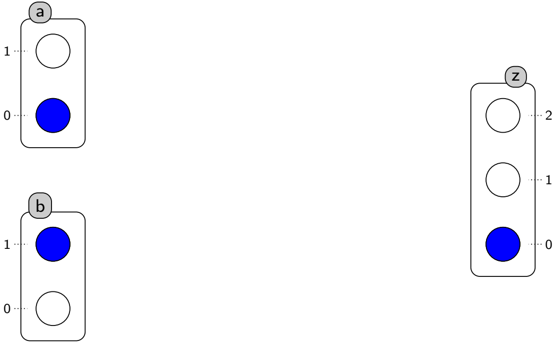


Sorts: components a, b, z

Processes: local states / discrete expression levels z_0, z_1, z_2

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



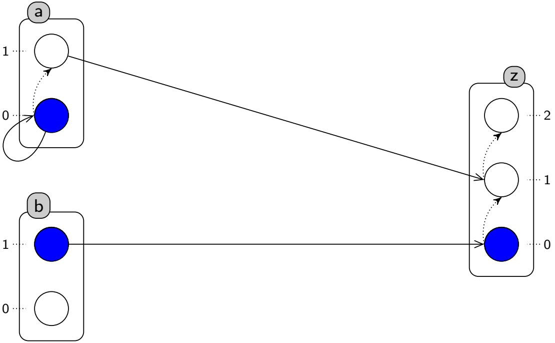
Sorts: components a, b, z

Processes: local states / discrete expression levels z_0, z_1, z_2

States: sets of active processes $\langle a_0, b_1, z_0 \rangle$

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Sorts: components $a, \ b, \ z$

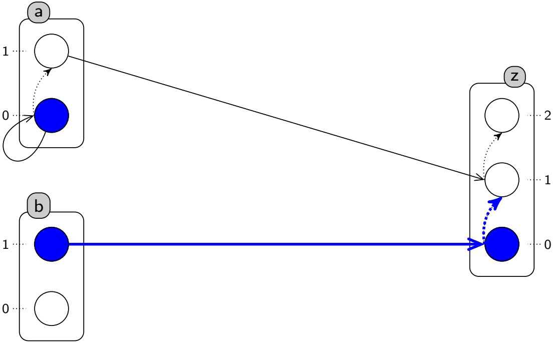
Processes: local states / discrete expression levels $z_0, \ z_1, \ z_2$

States: sets of active processes $\langle a_0, b_1, z_0 \rangle$

Actions: dynamics $b_1 \rightarrow z_0 \upharpoonright z_1, \ a_0 \rightarrow a_0 \upharpoonright a_1, \ a_1 \rightarrow z_1 \upharpoonright z_2$

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Sorts: components a, b, z

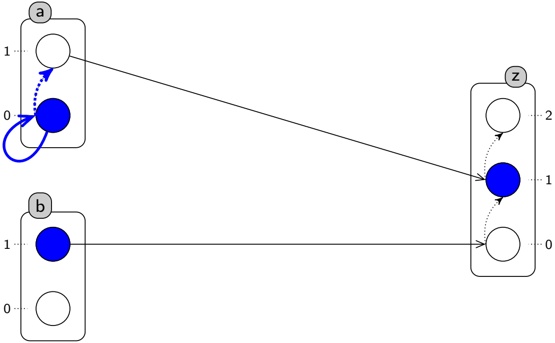
Processes: local states / discrete expression levels z_0, z_1, z_2

States: sets of active processes $\langle a_0, b_1, z_0 \rangle$

Actions: dynamics $\underline{b_1 \rightarrow z_0 \upharpoonright z_1}$, $a_0 \rightarrow a_0 \upharpoonright a_1$, $a_1 \rightarrow z_1 \upharpoonright z_2$

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Sorts: components a, b, z

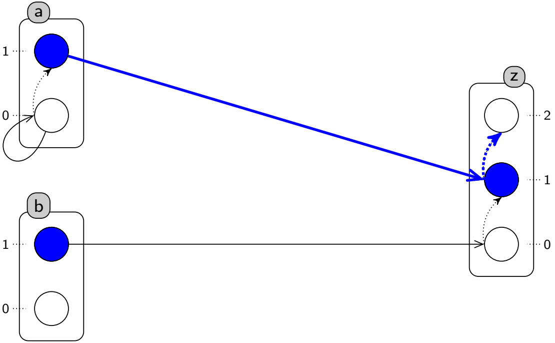
Processes: local states / discrete expression levels z_0, z_1, z_2

States: sets of active processes $\langle a_0, b_1, z_1 \rangle$

Actions: dynamics $b_1 \rightarrow z_0 \Rsh z_1$, $\underline{a_0 \rightarrow a_0 \Rsh a_1}$, $a_1 \rightarrow z_1 \Rsh z_2$

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Sorts: components a, b, z

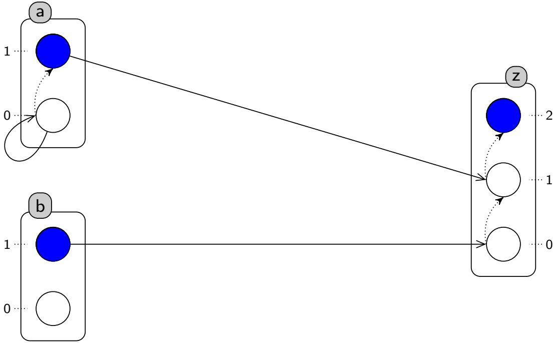
Processes: local states / discrete expression levels z_0, z_1, z_2

States: sets of active processes $\langle a_1, b_1, z_1 \rangle$

Actions: dynamics $b_1 \rightarrow z_0 \upharpoonright z_1$, $a_0 \rightarrow a_0 \upharpoonright a_1$, $\underline{a_1 \rightarrow z_1 \upharpoonright z_2}$

Standard Process Hitting

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Sorts: components a, b, z

Processes: local states / discrete expression levels z_0, z_1, z_2

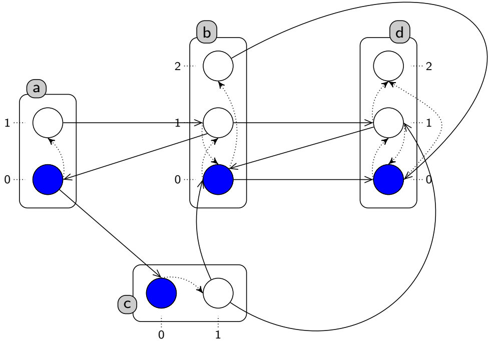
States: sets of active processes $\langle a_1, b_1, z_2 \rangle$

Actions: dynamics $b_1 \rightarrow z_0 \upharpoonright z_1$, $a_0 \rightarrow a_0 \upharpoonright a_1$, $a_1 \rightarrow z_1 \upharpoonright z_2$

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



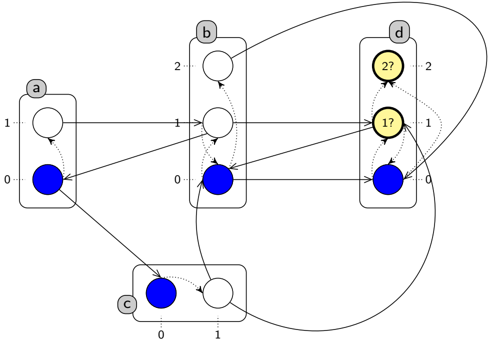
- Initial state

$\langle a_1, b_0, c_0, d_0 \rangle$

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$\langle a_1, b_0, c_0, d_0 \rangle$

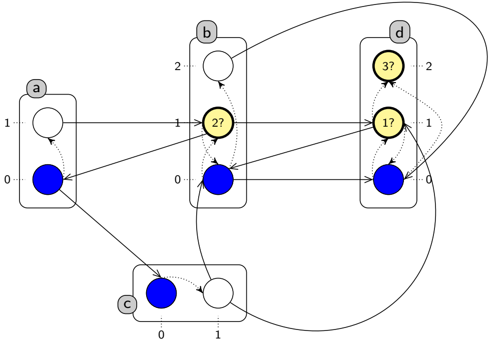
- Objectives

$[\ \Rsh d_1 \ :: \Rsh d_2\ ]$

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$$\langle a_1, b_0, c_0, d_0 \rangle$$

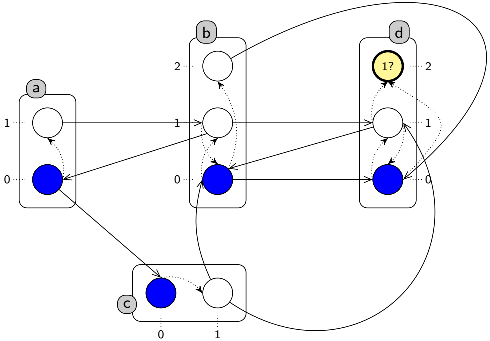
- Objectives

$$[\ \Rsh d_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$\langle a_1, b_0, c_0, d_0 \rangle$

- Objectives

$[\ \Rsh d_1 :: \Rsh d_2\ ]$

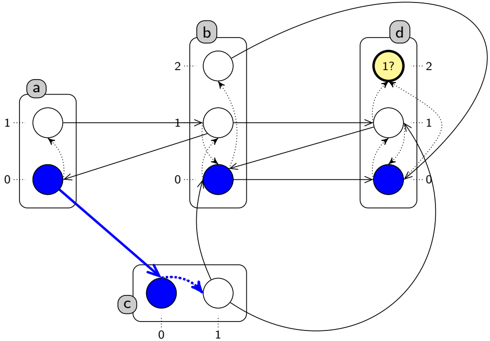
$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$

$[\ \Rsh d_2\ ]$

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$$\langle a_1, b_0, c_0, d_0 \rangle$$

- Objectives

$$[\ \Rsh d_1\ ::\ \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_2\ ]$$

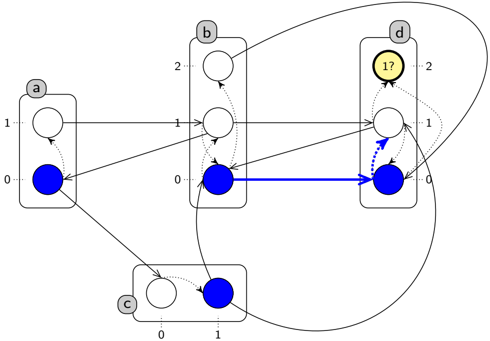
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$\langle a_1, b_0, c_0, d_0 \rangle$

- Objectives

$[\ \Rsh d_1 :: \Rsh d_2\ ]$

$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$

$[\ \Rsh d_2\ ]$

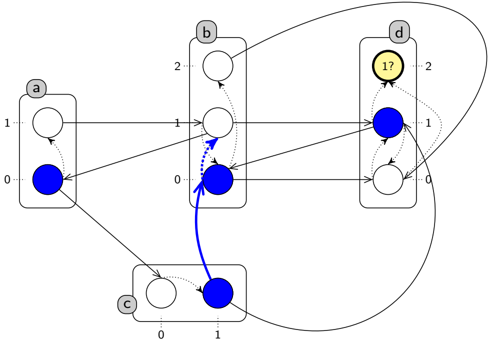
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$\langle a_1, b_0, c_0, d_0 \rangle$

- Objectives

$[\ \Rsh d_1 :: \Rsh d_2\ ]$

$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$

$[\ \Rsh d_2\ ]$

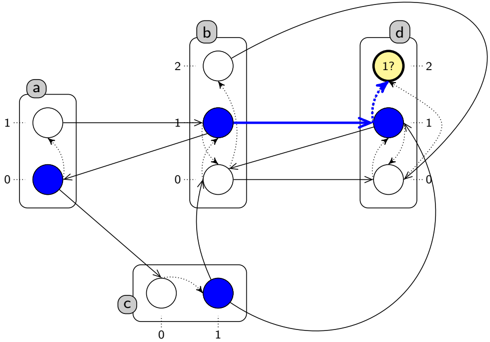
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$$\langle a_1, b_0, c_0, d_0 \rangle$$

- Objectives

$$[\ \Rsh d_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_2\ ]$$

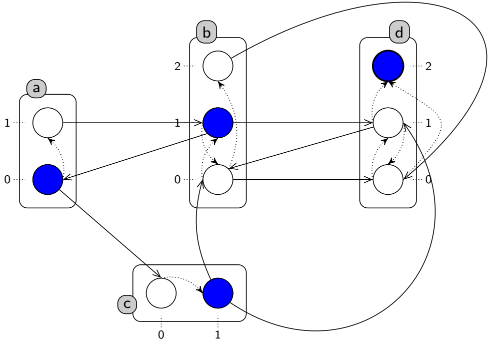
→ Concretization of the objective = scenario

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

Static analysis: successive reachability

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Successive reachability of processes:



- Initial state

$\langle a_1, b_0, c_0, d_0 \rangle$

- Objectives

$[\ {\Rsh}\, d_1 :: {\Rsh}\, d_2\ ]$

$[\ {\Rsh}\, d_1 :: {\Rsh}\, b_1 :: {\Rsh}\, d_2\ ]$

$[\ {\Rsh}\, d_2\ ]$

$\rightarrow$ Concretization of the objective = scenario

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

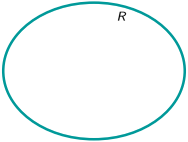
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

« From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



Exact solution

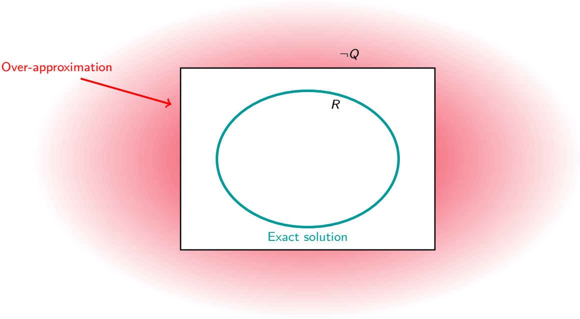
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

« From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



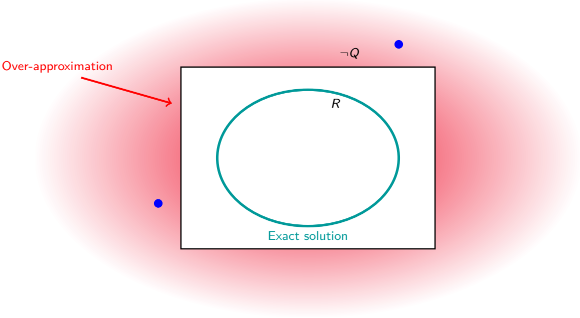
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

> « From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



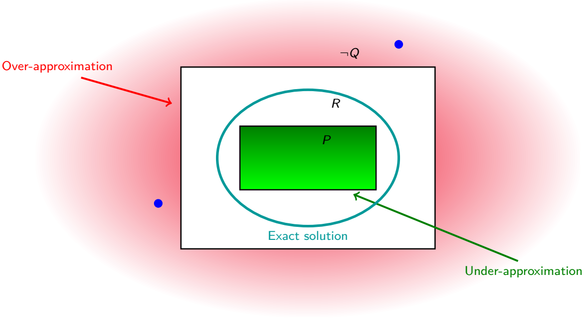
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

« From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



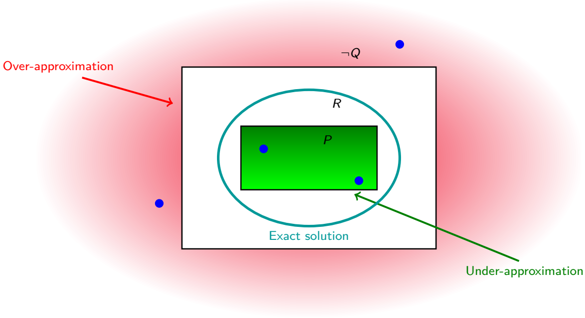
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

« From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



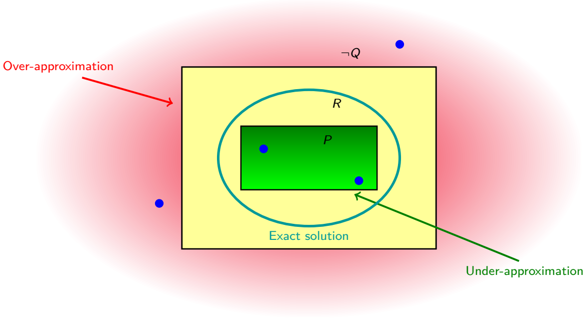
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

> « From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



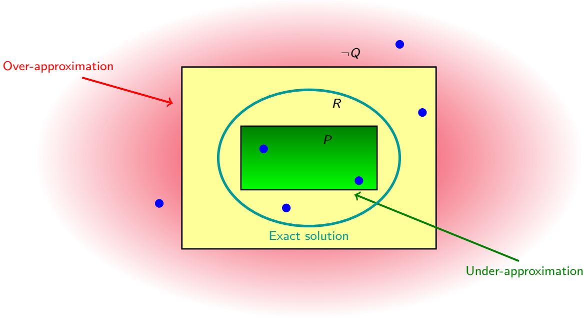
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

> « From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



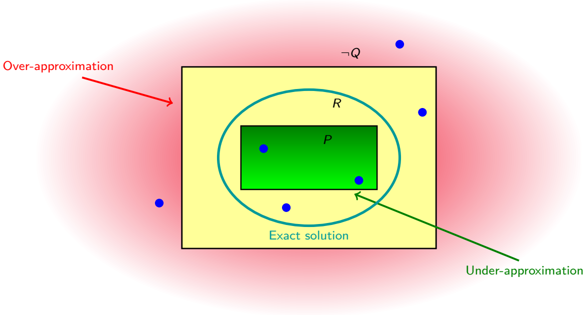
Approximations for the Reachability Analysis

[Paulevé *et al.* in *Mathematical Structures in Computer Science*, 2012]

Check reachability properties:

> « From an initial state s_0, is it possible to reach a state s_n where a_i is active? »

Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



Polynomial complexity in the number of sorts
Exponential complexity in the number of processes in each sort

→ Efficient for big models with few expression levels

Implementation & Execution times

PINT**: Existing free OCaml library**

→ Compiler + tools for Process Hitting models
→ Documentation & examples: `https://github.com/pauleve/pint`

Implementation & Execution times

PINT: **Existing free OCaml library**

→ Compiler + tools for Process Hitting models
→ Documentation & examples: https://github.com/pauleve/pint

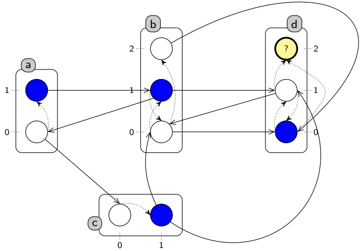
Computation time for various reachability analyses:

Model	Sorts	Procs	Actions	States	Biocham[1]	libddd[2]	PINT
egfr20	35	196	670	2^{64}	[3s–∞]	[1s–150s]	**0.007s**
tcrsig40	54	156	301	2^{73}	[1s–∞]	[0.6s–∞]	**0.004s**
tcrsig94	133	448	1124	2^{194}	∞	∞	**0.030s**
egfr104	193	748	2356	2^{320}	∞	∞	**0.050s**

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

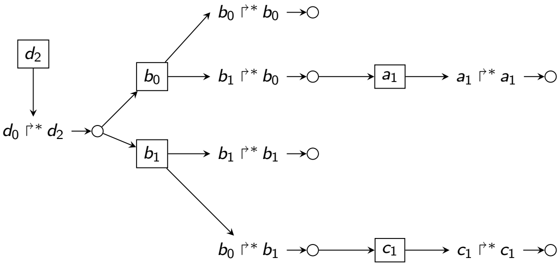
egfr20 : Epithelial Growth Factor Receptor (20 components) [Sahin *et al.*, 2009]
egfr104 : Epithelial Growth Factor Receptor (104 components) [Samaga *et al.*, 2009]
tcrsig40 : T-Cell Receptor (40 composants) [Klamt *et al.*, 2006]
tcrsig94 : T-Cell Receptor (94 composants) [Saez-Rodriguez *et al.*, 2007]

Under-approximation



Sufficient condition:

- no cycle
- each objective has a solution



$\boxed{d_2}$

$d_0 \Rsh^* d_2$ → ○

$\boxed{b_0}$

$b_0 \Rsh^* b_0$ → ○

$b_1 \Rsh^* b_0$ → ○ → $\boxed{a_1}$ → $a_1 \Rsh^* a_1$ → ○

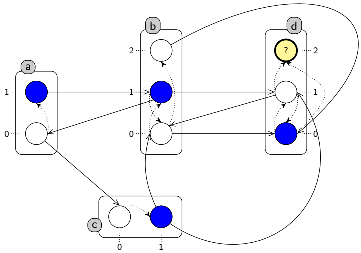
$\boxed{b_1}$

$b_1 \Rsh^* b_1$ → ○

$b_0 \Rsh^* b_1$ → ○ → $\boxed{c_1}$ → $c_1 \Rsh^* c_1$ → ○

$\boxed{d_2}$	Required process
$d_0 \Rsh^* d_2$	Objective
○	Solution to an objective

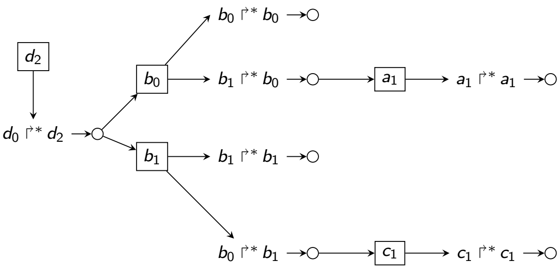
Under-approximation



Sufficient condition:

- no cycle
- each objective has a solution

R is **true**



$\boxed{d_2}$

$d_0 \Rsh^* d_2 \rightarrow$ ○

$\boxed{b_0}$

$b_0 \Rsh^* b_0 \rightarrow$ ○

$b_1 \Rsh^* b_0 \rightarrow$ ○ $\rightarrow \boxed{a_1} \rightarrow a_1 \Rsh^* a_1 \rightarrow$ ○

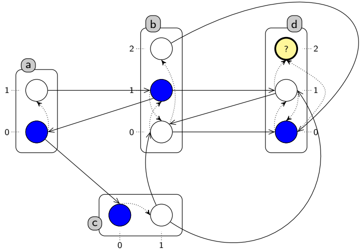
$\boxed{b_1}$

$b_1 \Rsh^* b_1 \rightarrow$ ○

$b_0 \Rsh^* b_1 \rightarrow$ ○ $\rightarrow \boxed{c_1} \rightarrow c_1 \Rsh^* c_1 \rightarrow$ ○

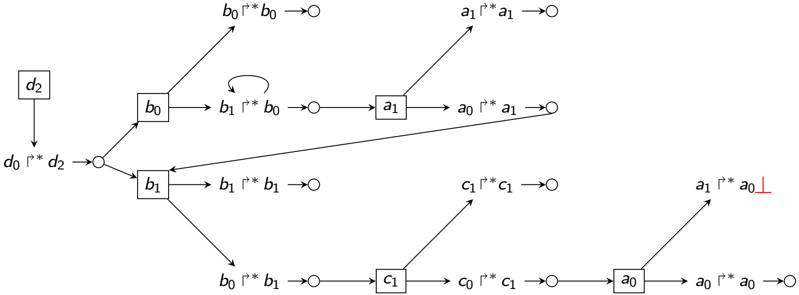
$\boxed{d_2}$	Required process
$d_0 \Rsh^* d_2$	Objective
○	Solution to an objective

Under-approximation

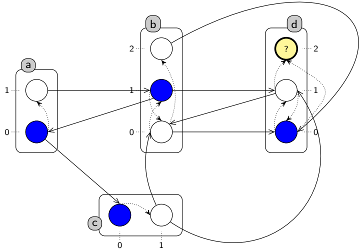


Sufficient condition:

- no cycle
- ~~each objective has a solution~~



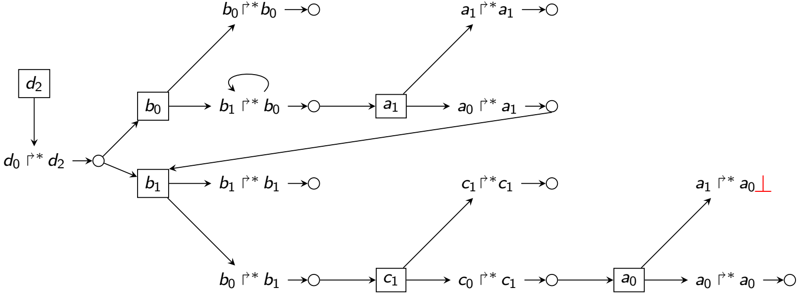
Under-approximation



Sufficient condition:

- no cycle
- ~~each objective has a solution~~

Inconclusive



$\boxed{d_2}$

$d_0 \Rsh^{*} d_2$

$\boxed{b_0}$

$b_0 \Rsh^{*} b_0$

$b_1 \Rsh^{*} b_0$

$\boxed{a_1}$

$a_1 \Rsh^{*} a_1$

$a_0 \Rsh^{*} a_1$

$\boxed{b_1}$

$b_1 \Rsh^{*} b_1$

$b_0 \Rsh^{*} b_1$

$\boxed{c_1}$

$c_1 \Rsh^{*} c_1$

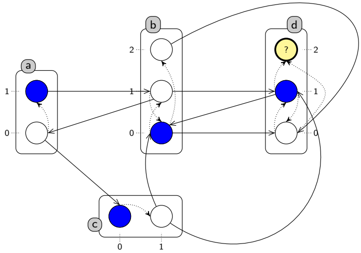
$c_0 \Rsh^{*} c_1$

$\boxed{a_0}$

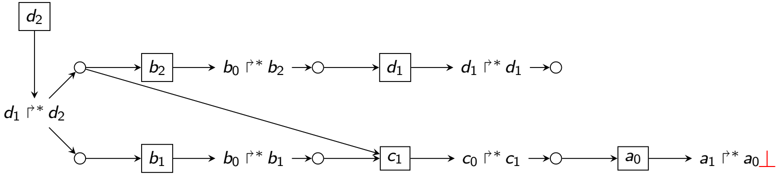
$a_1 \Rsh^{*} a_0 \bot$

$a_0 \Rsh^{*} a_0$

Over-approximation



Necessary condition:

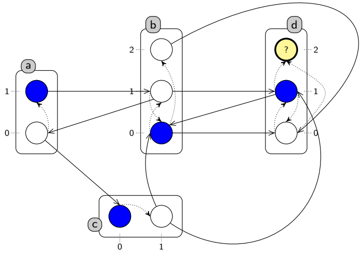

$\boxed{d_2}$

$d_1 \upharpoonright^* d_2$

$\boxed{b_2}$ $b_0 \upharpoonright^* b_2$ $\boxed{d_1}$ $d_1 \upharpoonright^* d_1$

$\boxed{b_1}$ $b_0 \upharpoonright^* b_1$ $\boxed{c_1}$ $c_0 \upharpoonright^* c_1$ $\boxed{a_0}$ $a_1 \upharpoonright^* a_0 \perp$

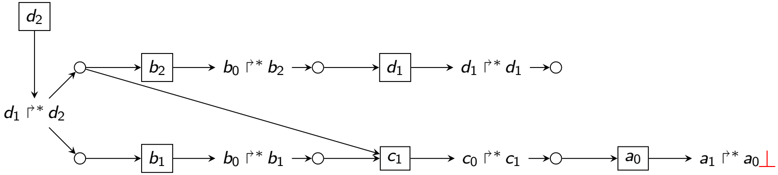
Over-approximation



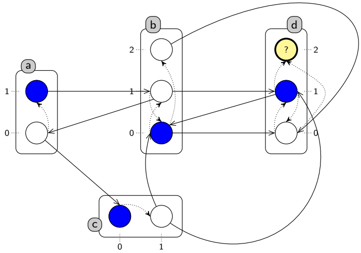
Necessary condition:

There exists a traversal with no cycle

- objective → follow one solution
- solution → follow all processes
- process → follow all objectives



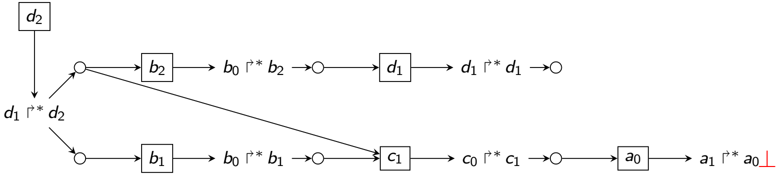
Over-approximation



Necessary condition:

~~There exists a traversal~~ with no cycle

- ~~objective → follow one solution~~
- solution → follow all processes
- process → follow all objectives



d_2

$d_1 \Rsh^{*} d_2$

b_2

$b_0 \Rsh^{*} b_2$

d_1

$d_1 \Rsh^{*} d_1$

b_1

$b_0 \Rsh^{*} b_1$

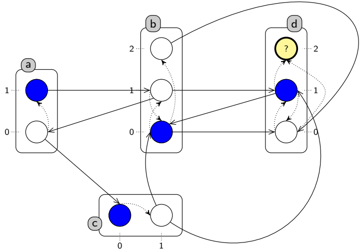
c_1

$c_0 \Rsh^{*} c_1$

a_0

$a_1 \Rsh^{*} a_0 \bot$

Over-approximation

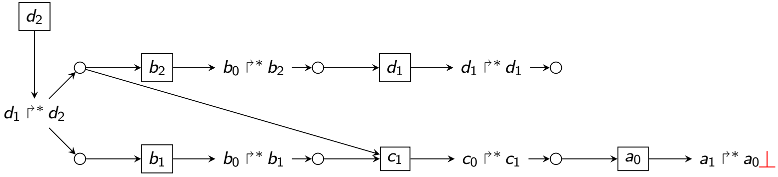


Necessary condition:

~~There exists a traversal~~ with no cycle

- ~~objective → follow one solution~~
- solution → follow all processes
- process → follow all objectives

R is **false**



d_2

$d_1 \Rsh^* d_2$

b_2

$b_0 \Rsh^* b_2$

d_1

$d_1 \Rsh^* d_1$

b_1

$b_0 \Rsh^* b_1$

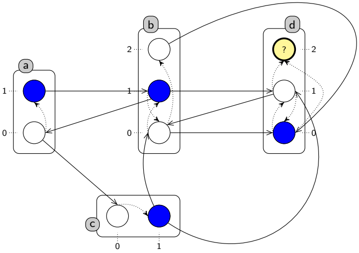
c_1

$c_0 \Rsh^* c_1$

a_0

$a_1 \Rsh^* a_0 \perp$

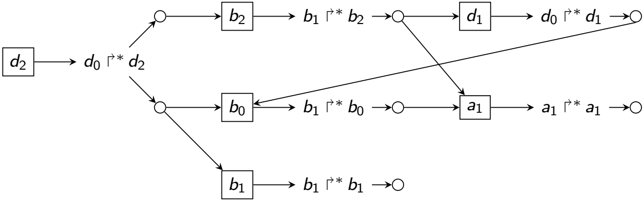
Over-approximation



Necessary condition:

There exists a traversal with no cycle

- objective → follow one solution
- solution → follow all processes
- process → follow all objectives



$\boxed{d_2}$

$d_0 \Rsh^* d_2$

$\boxed{b_2}$

$b_1 \Rsh^* b_2$

$\boxed{d_1}$

$d_0 \Rsh^* d_1$

$\boxed{b_0}$

$b_1 \Rsh^* b_0$

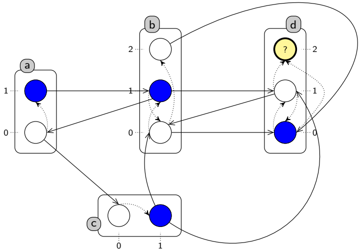
$\boxed{a_1}$

$a_1 \Rsh^* a_1$

$\boxed{b_1}$

$b_1 \Rsh^* b_1$

Over-approximation

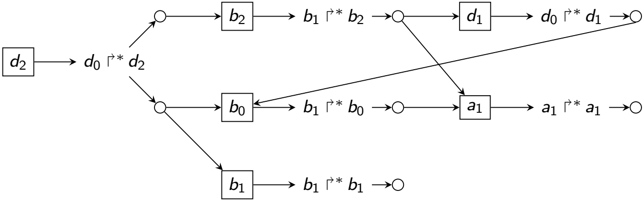


Necessary condition:

There exists a traversal with no cycle

- objective → follow one solution
- solution → follow all processes
- process → follow all objectives

Inconclusive



d_2 $d_0 \Rsh^{*} d_2$

b_2 $b_1 \Rsh^{*} b_2$ d_1 $d_0 \Rsh^{*} d_1$

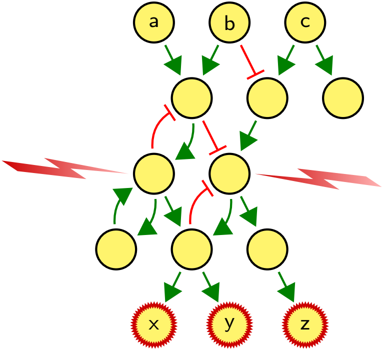
b_0 $b_1 \Rsh^{*} b_0$ a_1 $a_1 \Rsh^{*} a_1$

b_1 $b_1 \Rsh^{*} b_1$

Cut sets

[Paulevé, Andrieux, Koeppl in *Computer Aided Verification*, 2013.]

Cut set = set of nodes whose knockout is sufficient to turn off some outputs

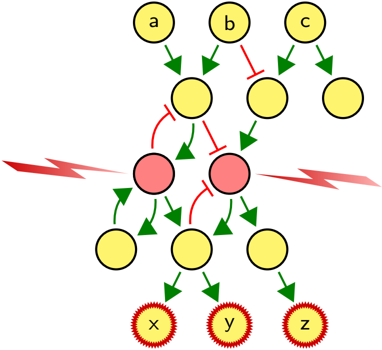


- "Absolute" vision of possible perturbations
- Need for an intermediate point of view → Finer analysis

Cut sets

[Paulevé, Andrieux, Koeppl in *Computer Aided Verification*, 2013.]

Cut set = set of nodes whose knockout is sufficient to turn off some outputs

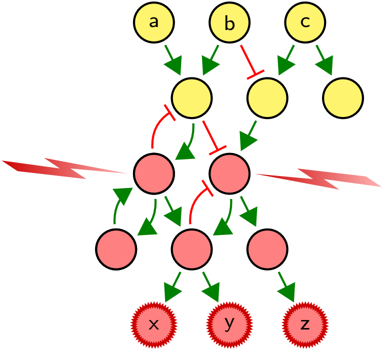


- "Absolute" vision of possible perturbations
- Need for an intermediate point of view → Finer analysis

Cut sets

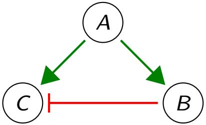
[Paulevé, Andrieux, Koeppl in *Computer Aided Verification*, 2013.]

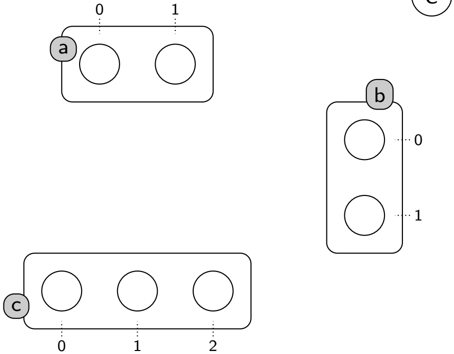
Cut set = set of nodes whose knockout is sufficient to turn off some outputs



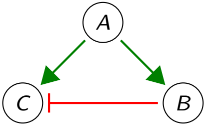
- "Absolute" vision of possible perturbations
- Need for an intermediate point of view → Finer analysis

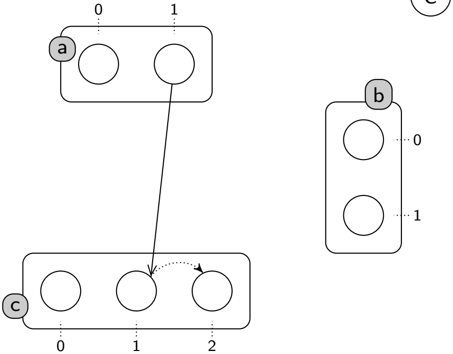
Translation in Process Hitting



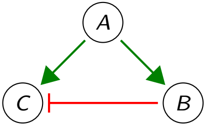


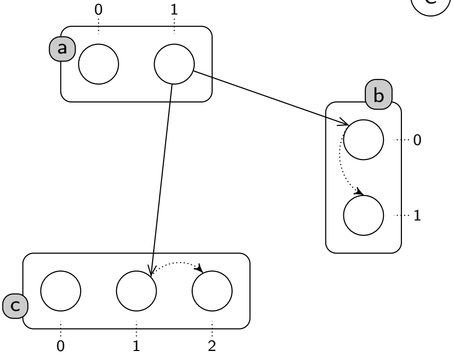
Translation in Process Hitting



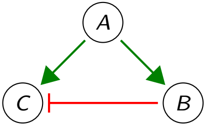


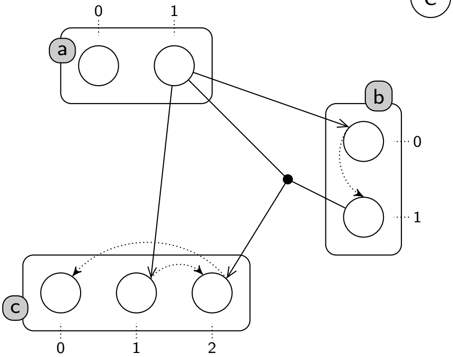
Translation in Process Hitting



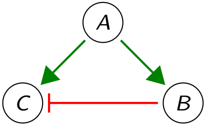


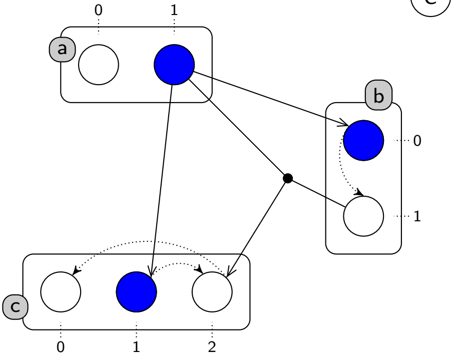
Translation in Process Hitting



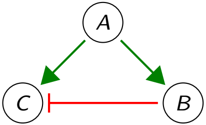


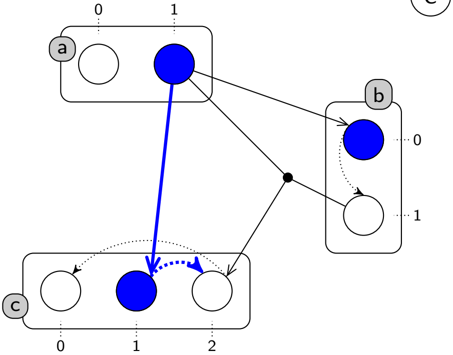
Translation in Process Hitting



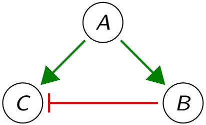


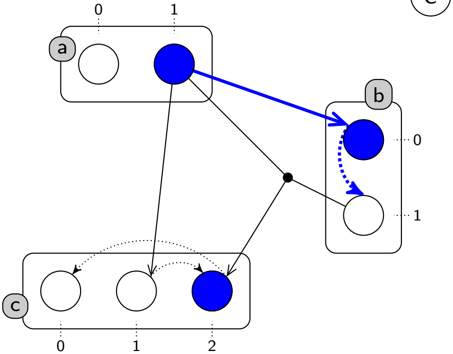
Translation in Process Hitting



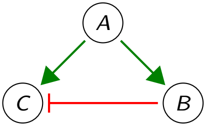


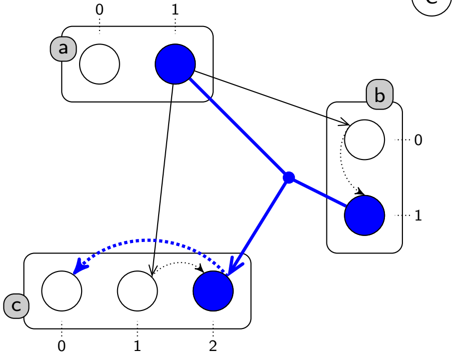
Translation in Process Hitting



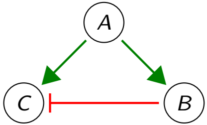


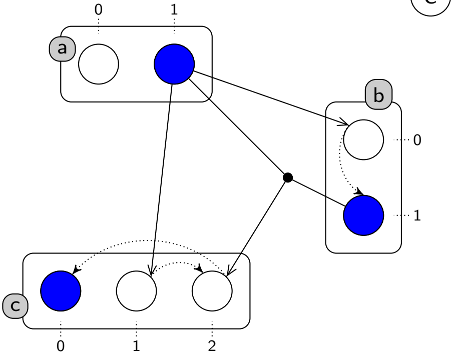
Translation in Process Hitting



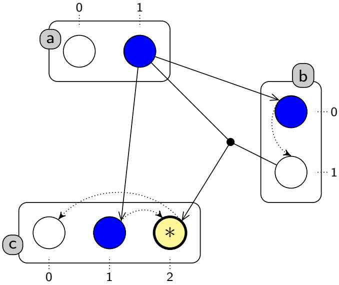


Translation in Process Hitting





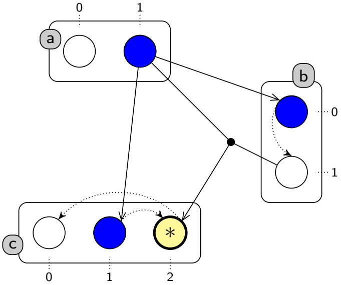
Detailed application of the Static Analysis



Graph of local causality:

$\boxed{c_0}$	Required process
$c_1 \upharpoonright^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

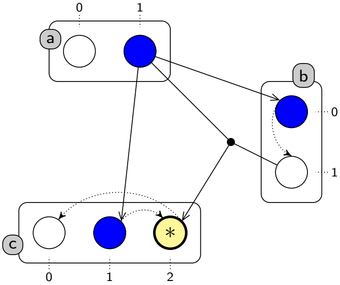


Graph of local causality:

$\boxed{c_2}$

$\boxed{c_0}$	Required process
$c_1 \Rsh^{*} c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis



Graph of local causality:

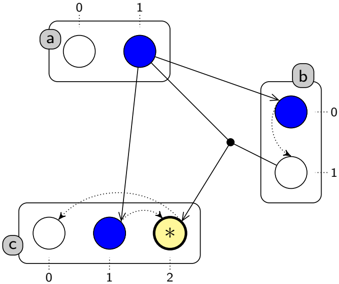
$\boxed{c_2} \longrightarrow c_1 \Rsh^* c_2$

$\boxed{c_0}$ Required process

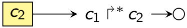
$c_1 \Rsh^* c_0$ Objective

○ Solution to an objective

Detailed application of the Static Analysis



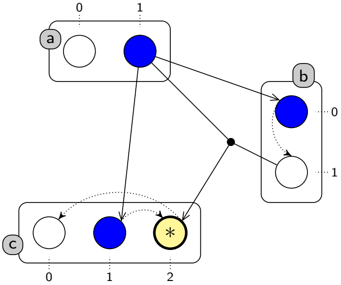
Graph of local causality:



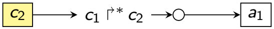
$\boxed{c_2} \longrightarrow c_1 \Rsh^{*} c_2 \longrightarrow \bigcirc$

$\boxed{c_0}$	Required process
$c_1 \Rsh^{*} c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

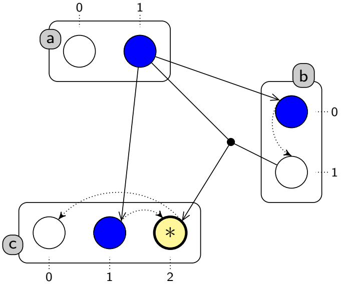


Graph of local causality:

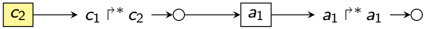


$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

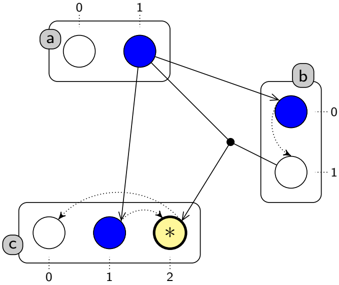


Graph of local causality:



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

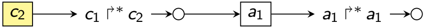
Detailed application of the Static Analysis



Sufficient condition:

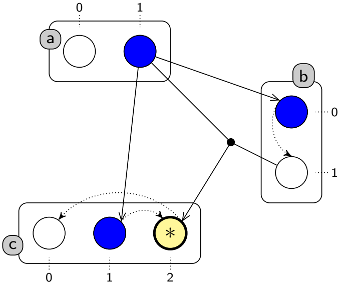
- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis



Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

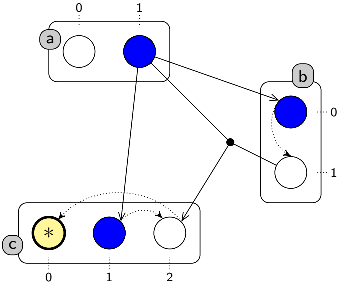
Graph of local causality:

OK

$$\boxed{c_2} \longrightarrow c_1 \Rsh^* c_2 \longrightarrow \bigcirc \longrightarrow \boxed{a_1} \longrightarrow a_1 \Rsh^* a_1 \longrightarrow \bigcirc$$

$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis



Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

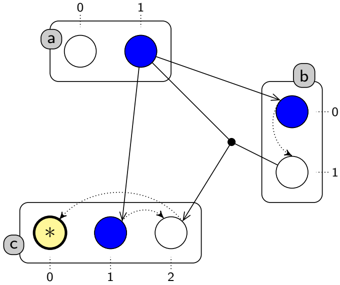
OK

$$\boxed{c_2} \longrightarrow c_1 \Rsh^* c_2 \longrightarrow \circ \longrightarrow \boxed{a_1} \longrightarrow a_1 \Rsh^* a_1 \longrightarrow \circ$$

$$\boxed{c_0}$$

$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

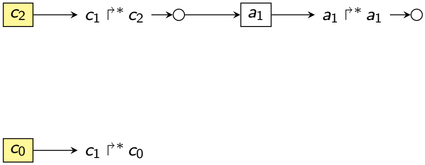


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

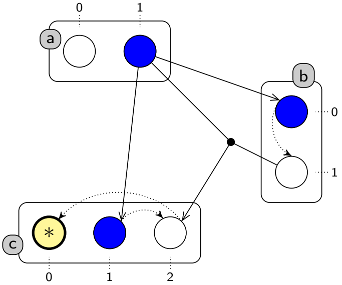
OK


$$\boxed{c_2} \longrightarrow c_1 \upharpoonright^* c_2 \longrightarrow ○ \longrightarrow \boxed{a_1} \longrightarrow a_1 \upharpoonright^* a_1 \longrightarrow ○$$

$$\boxed{c_0} \longrightarrow c_1 \upharpoonright^* c_0$$

$\boxed{c_0}$	Required process
$c_1 \upharpoonright^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

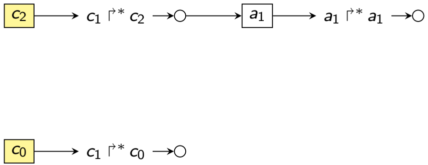


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

OK

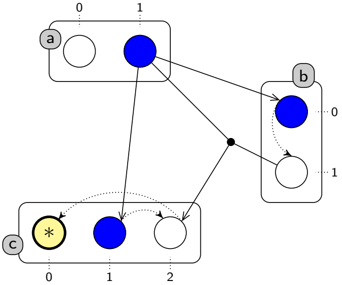


$\boxed{c_2} \longrightarrow c_1 \Rsh^* c_2 \longrightarrow \bigcirc \longrightarrow \boxed{a_1} \longrightarrow a_1 \Rsh^* a_1 \longrightarrow \bigcirc$

$\boxed{c_0} \longrightarrow c_1 \Rsh^* c_0 \longrightarrow \bigcirc$

$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

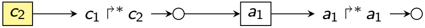


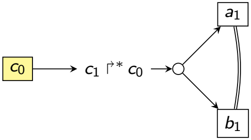
Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

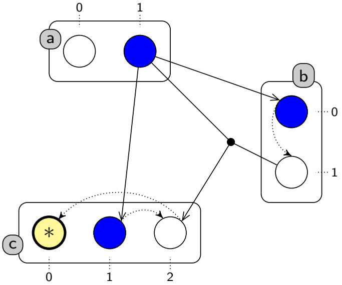
OK





$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

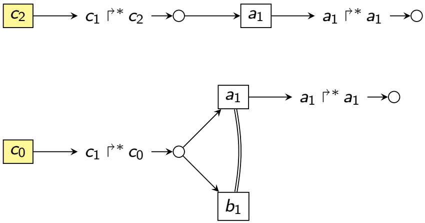


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

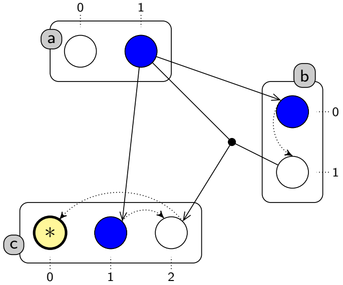
Graph of local causality:

OK



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

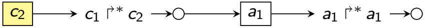


Sufficient condition:

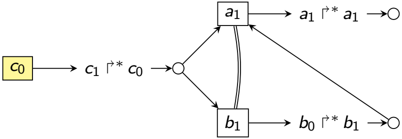
- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

OK



$\boxed{c_2}$ → $c_1 \Rsh^* c_2$ → ○ → $\boxed{a_1}$ → $a_1 \Rsh^* a_1$ → ○



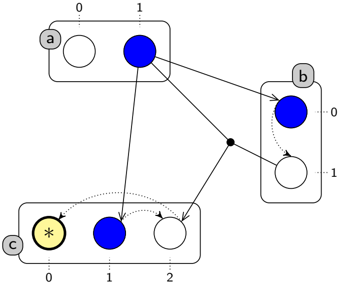
$\boxed{c_0}$ → $c_1 \Rsh^* c_0$ → ○

$\boxed{a_1}$ → $a_1 \Rsh^* a_1$ → ○

$\boxed{b_1}$ → $b_0 \Rsh^* b_1$ → ○

$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

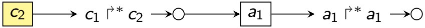


Sufficient condition:

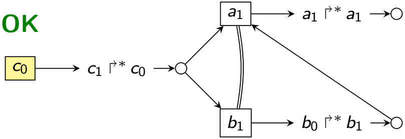
- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

OK

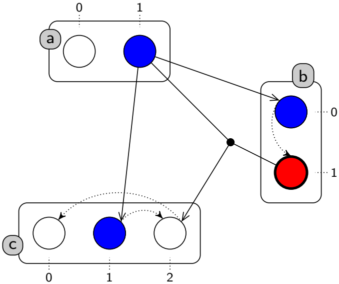


OK



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis



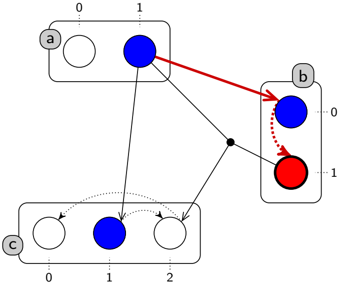
Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

$\boxed{c_0}$	Required process
$c_1 \upharpoonright^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis



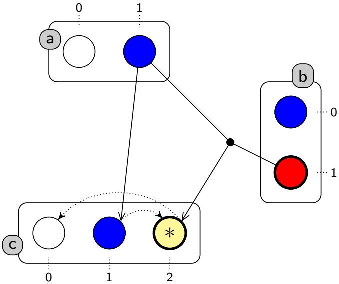
Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

$\boxed{c_0}$	Required process
$c_1 \upharpoonright^{*} c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

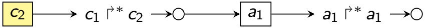


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

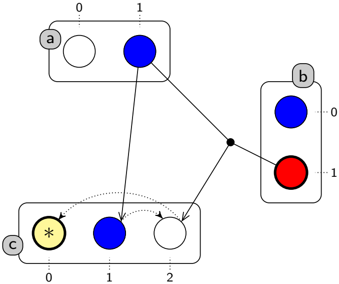
Graph of local causality:

OK



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

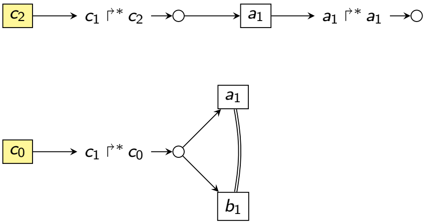


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

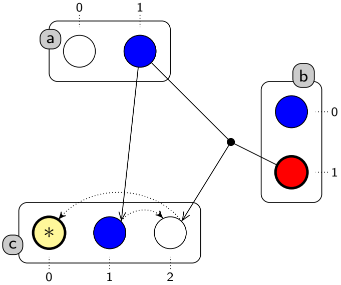
Graph of local causality:

OK



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

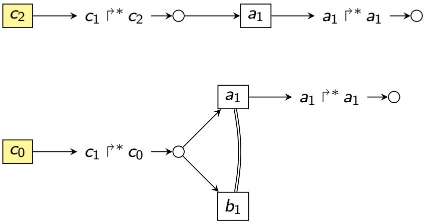


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

OK



$\boxed{c_2} \rightarrow c_1 \Rsh^* c_2 \rightarrow ○ \rightarrow \boxed{a_1} \rightarrow a_1 \Rsh^* a_1 \rightarrow ○$

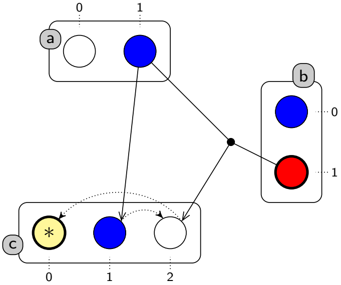
$\boxed{c_0} \rightarrow c_1 \Rsh^* c_0 \rightarrow ○$

$\boxed{a_1} \rightarrow a_1 \Rsh^* a_1 \rightarrow ○$

$\boxed{b_1}$

$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

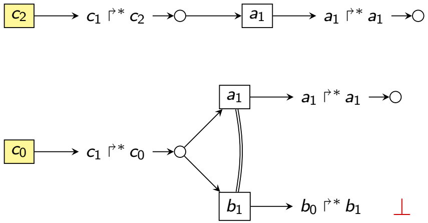


Sufficient condition:

- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

OK



$\boxed{c_2} \longrightarrow c_1 \Rsh^* c_2 \longrightarrow \bigcirc \longrightarrow \boxed{a_1} \longrightarrow a_1 \Rsh^* a_1 \longrightarrow \bigcirc$

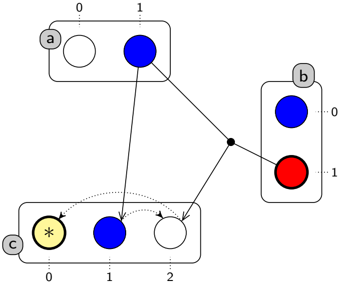
$\boxed{c_0} \longrightarrow c_1 \Rsh^* c_0 \longrightarrow \bigcirc$

$\boxed{a_1} \longrightarrow a_1 \Rsh^* a_1 \longrightarrow \bigcirc$

$\boxed{b_1} \longrightarrow b_0 \Rsh^* b_1$ $\perp$

$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Detailed application of the Static Analysis

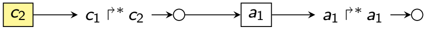


Sufficient condition:

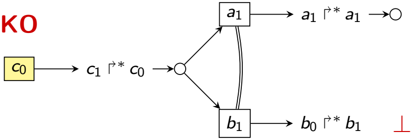
- no cycle
- each objective has a solution
- no contradiction between synchronous requirements

Graph of local causality:

OK



KO



$\boxed{c_0}$	Required process
$c_1 \Rsh^* c_0$	Objective
○	Solution to an objective

Alternative Paths

Provided that the computed path is minimal, new properties emerge:

Alternative Paths

Provided that the computed path is minimal, new properties emerge:

→ Some components have no impact

- They do not appear in the graph of local causality
- Initial states do not depend on them
- Simplifies the research

Alternative Paths

Provided that the computed path is minimal, new properties emerge:

→ Some components have no impact

- They do not appear in the graph of local causality
- Initial states do not depend on them
- Simplifies the research

→ Knocking out a component in a path may reveal an alternative path

- Resilience ⇒ existence of alternative paths (cf. cut sets)
- New path ⇒ New costs or new delays

Conclusion

The Process Hitting allows to represent biological regulatory networks:

- Qualitative and atomistic modeling
- Existing efficient **reachability analysis**
- Links with other formalisms → especially from Thomas' modeling

Conclusion

The Process Hitting allows to represent biological regulatory networks:

- Qualitative and atomistic modeling
- Existing efficient **reachability analysis**
- Links with other formalisms → especially from Thomas' modeling

The **impact degree**:

- Quantification of the importance of a component
- Highlights possible recovery paths
- But limited to the presence/absence of a component

Conclusion

The Process Hitting allows to represent biological regulatory networks:

- Qualitative and atomistic modeling
- Existing efficient **reachability analysis**
- Links with other formalisms → especially from Thomas' modeling

The **impact degree**:

- Quantification of the importance of a component
- Highlights possible recovery paths
- But limited to the presence/absence of a component

Quantification of the perturbation using Process Hitting:

- Adapted notion of **impact degree** (multiple values)
- Thanks to the powerful **reachability analysis**
- Additional properties with the graph of local causality

Thank you!

Do you have questions

or suggestions?

Bibliography

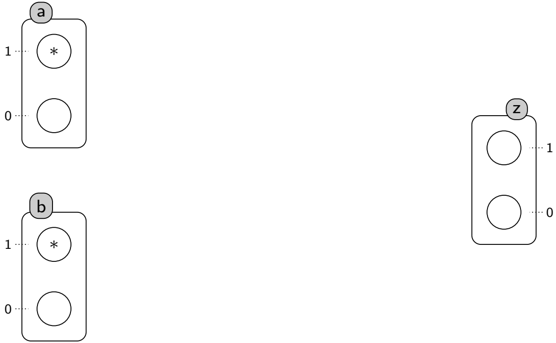
- Hao Jiang, Takeyuki Tamura, Wai-Ki Ching and Tatsuya Akutsu. On the Complexity of Inference and Completion of Boolean Networks from Given Singleton Attractors. In IEICE Transactions on Fundamentals of Electronics, *Communications and Computer Sciences* E96.A, n. 11, pages 2265–74, 2013.

- Jean-Paul Comet and Gilles Bernot. Introducing continuous time in discrete models of gene regulatory networks. In *Proceedings of the Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, EDP Sciences, 2010.

- Loïc Paulevé, Geoffroy Andrieux and Heinz Koeppl. Under-approximating cut sets for reachability in large scale automata networks. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification*, volume 8044 of Lecture Notes in Computer Science, pages 69–84. Springer Berlin Heidelberg, 2013.

Cooperations

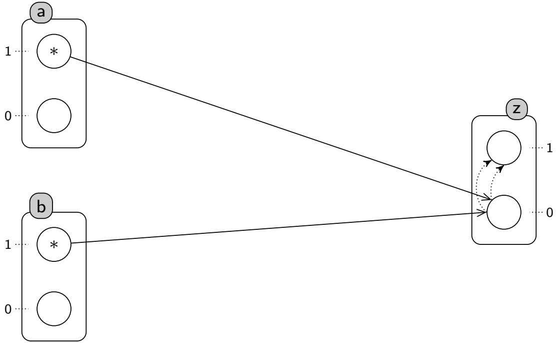
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonleft z_1$

Cooperations

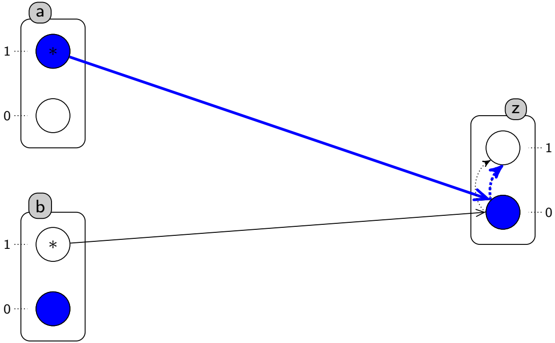
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Cooperations

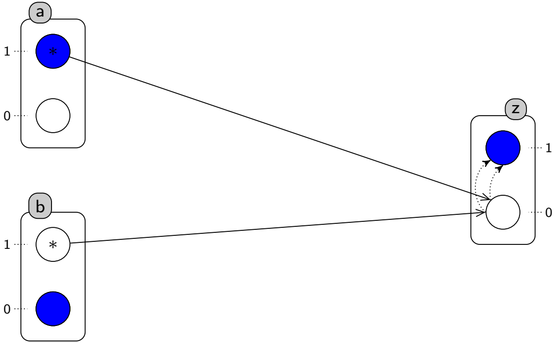
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Cooperations

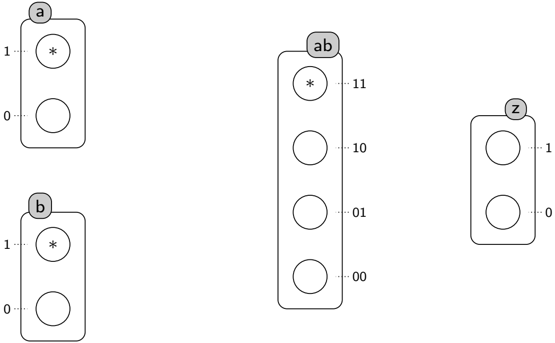
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

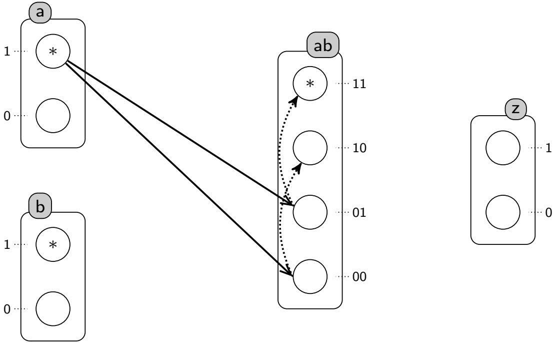


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

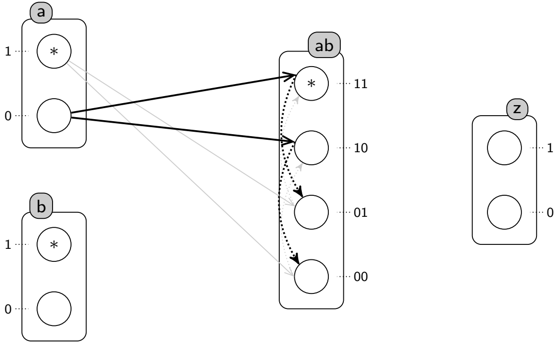


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

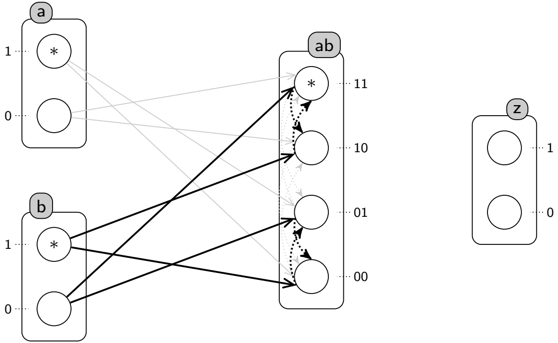


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

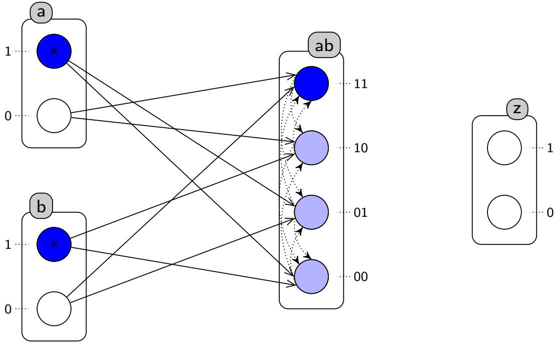


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



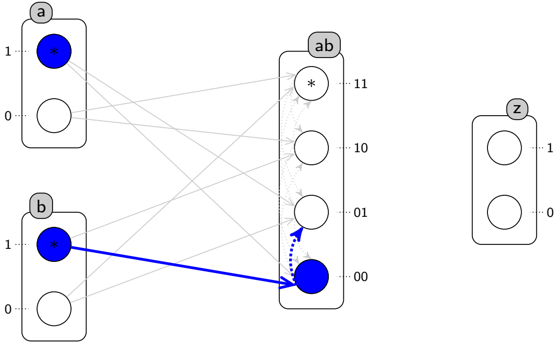
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Each configuration is represented by one process $\underline{a_1 \wedge b_1} \Rightarrow ab_{11}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



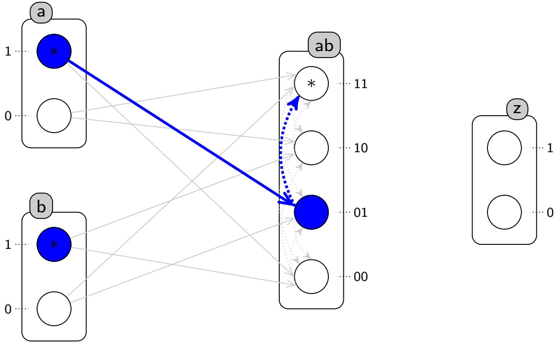
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Each configuration is represented by one process $\underline{a_1 \wedge b_1} \Rightarrow ab_{11}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



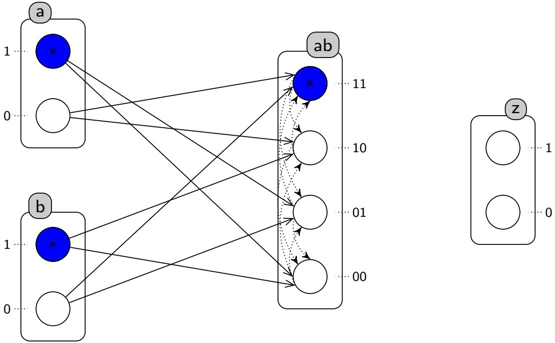
Cooperation between a_1 and b_1: $\underline{a_1 \land b_1} \to z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \land b_1}$

Each configuration is represented by one process $\underline{a_1 \land b_1} \Rightarrow ab_{11}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



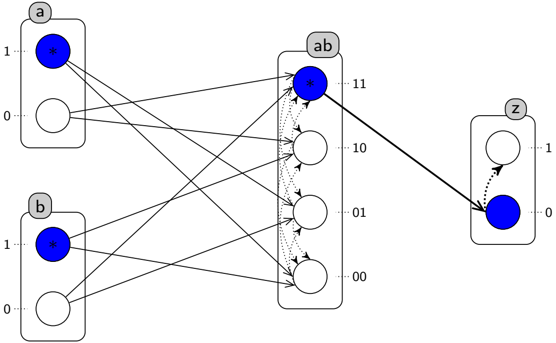
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Each configuration is represented by one process $\underline{a_1 \wedge b_1} \Rightarrow ab_{11}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



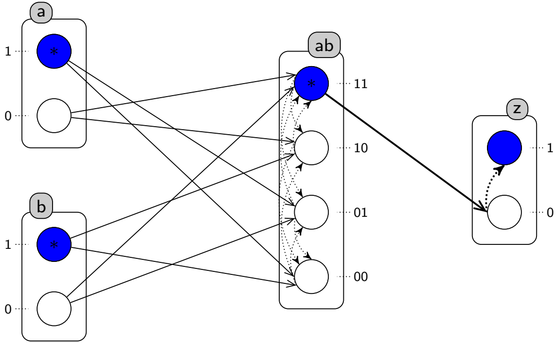
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \upharpoonright z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

Each configuration is represented by one process $\underline{a_1 \wedge b_1} \Rightarrow ab_{11}$

Cooperations

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Cooperation between a_1 and b_1: $\underline{a_1 \land b_1} \rightarrow z_0 \Rsh z_1$

Solution: a **cooperative sort** ab to express $\underline{a_1 \land b_1}$

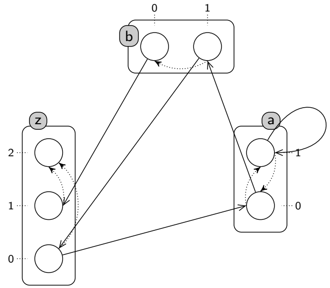
Each configuration is represented by one process $\underline{a_1 \land b_1} \Rightarrow ab_{11}$

Static Analysis: Fixed Points

[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action



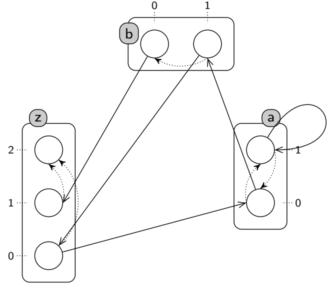
Static Analysis: Fixed Points

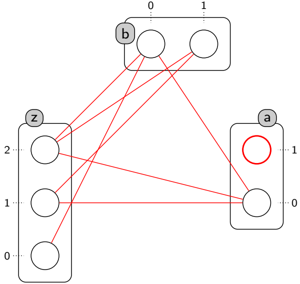
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph





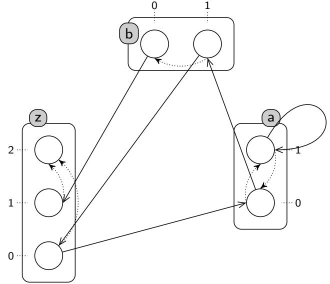
Static Analysis: Fixed Points

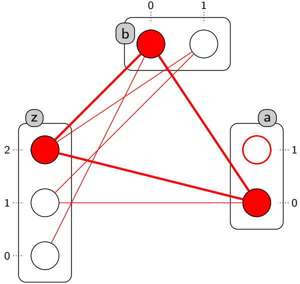
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points





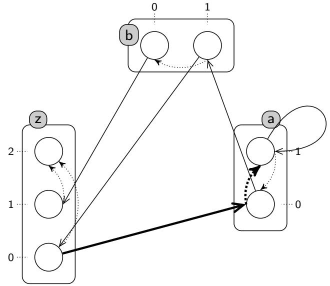
Static Analysis: Fixed Points

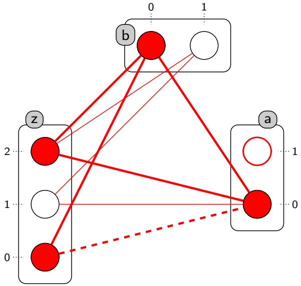
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points





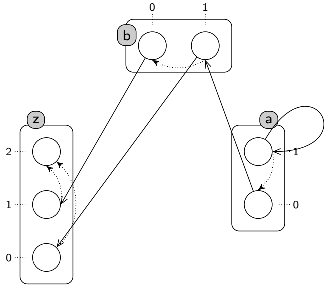
Static Analysis: Fixed Points

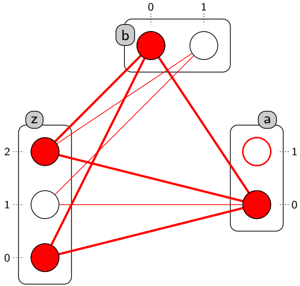
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points





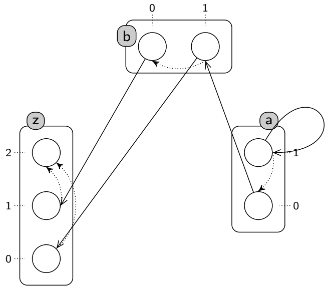
Static Analysis: Fixed Points

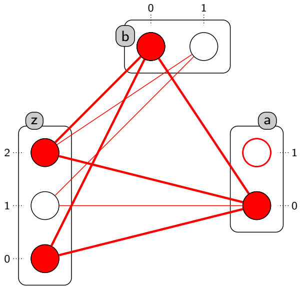
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph → **n-cliques** = fixed points





Exponential complexity w.r.t. the number of sorts