Towards the analysis and inference of large biological models

Maxime Folschette, Morgan Magnin
maxime.folschette | morgan.magnin @irccyn.ec-nantes.fr
Joint work with: K.Inoue, L. Paulevé, O. Roux

École Centrale de Nantes - IRCCyN - *MeForBio team*

5th JFLI-LRI-NII workshop - 2013/10/10

Overview

MeForBio (IRCCyN team, ECN): Formal Methods for Bioinformatics

Research axes

- Models: automata, Petri nets, boolean networks, process algebra ($\rightarrow$ process hitting)
- Extended with: time (**chronometry** vs **chronology**) and/or parameters
- Analysis techniques: **model-checking**, **control**, **abstraction**, **parameters inference**
- Applied (and/or designed) to biology, e.g. biological regulatory networks

MeForBio (IRCCyN team, ECN)

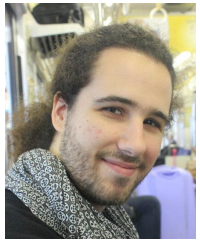
MeForBio (IRCCyN team, ECN)

Olivier ROUX
Professor & team leader

Morgan MAGNIN
Associate professor

Carito GUZIOLOWSKI
Associate professor

Julien GRAS
Research engineer



Maxime FOLSCHETTE
3rd year PhD student

Courtney CHANCELLOR
2nd year PhD student

Today's issue

Tricky question

How can we study complex dynamical biological systems, **involving up to 1.000 interacting components**?

Observation

- Classical model-checking approaches suffer from state space explosion
- Leads:
  - Taking profit for Process Algebra structure, based on a **compact representation of the interactions**
  - Develop **static analysis approaches** to verify some crucial properties, e.g. stable states, reachability, key processes, ...

Contribution

Scientific challenge

How can we cope with the analysis of **large-scale systems**, involving up to 1.000 interacting components?

Objectives of this talk (and Loïc's one)

- Introduce a Process Algebra inspired framework based on a compact representation of the interactions
- Develop efficient **static analysis approaches** to answer most common problems
- Apply the methodology to large-scale biological regulatory networks

Overview

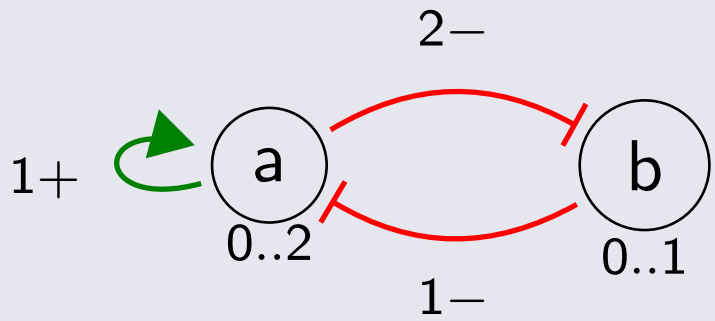
Short introduction to Biological Regulatory Networks

Principle of R. Thomas' discrete modeling [TGL76]

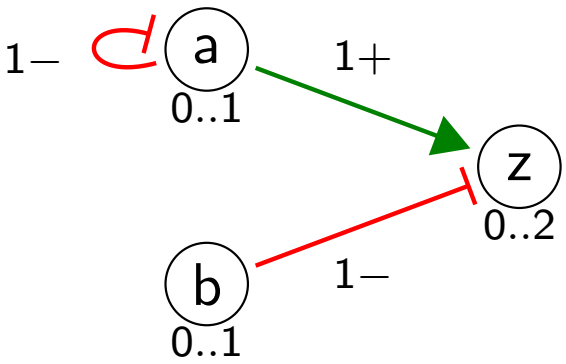
- **Activations** and **inhibitions** between genes
- Gene/protein couples
- Genes expression is associated to a set of discrete logic levels
- **Effective control** beyond a given threshold; opposite effect below.

Interaction graph

- Nodes = **Genes**
- Directed edges = **Interactions**
- But what is the **evolutionary tendency** of a when a is at level 1 and b at level 1? $\Rightarrow$ Need for **parametrization**



Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
$-$	$+$	1
$-$	$-$	0
$+$	$+$	2
$+$	$-$	1

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

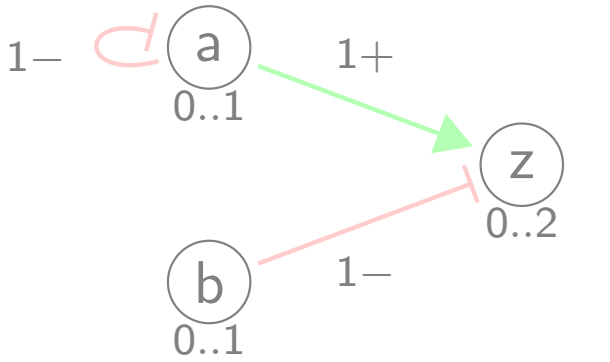
$k_{b,\omega}$
1

Proposed by René Thomas in 1973, several extensions since then

Historical bio-informatics model for studying genes interactions

Widely used and well-adapted to represent dynamic gene systems

Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
$-$	$+$	1
$-$	$-$	0
$+$	$+$	2
$+$	$-$	1

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Interaction Graph: structure of the system (genes & interactions)

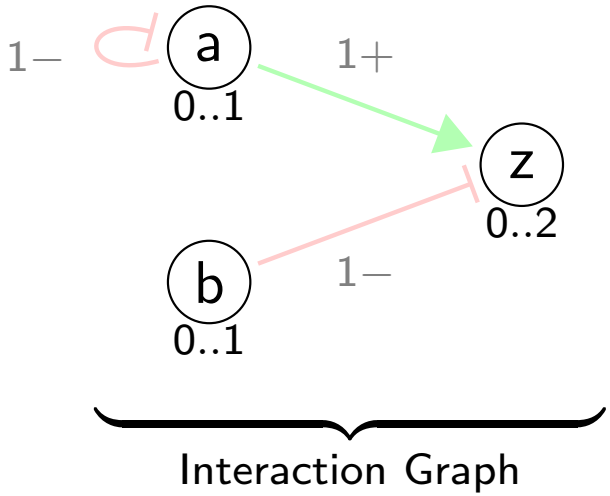
Nodes: genes

- Name a, b, z
- Possible values (levels of expression) 0..1, 0..2

Edges: interactions

- Threshold 1
- Type (activation or inhibition) $+$ / $-$

Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
$-$	$+$	1
$-$	$-$	0
$+$	$+$	2
$+$	$-$	1

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Interaction Graph: structure of the system (genes & interactions)

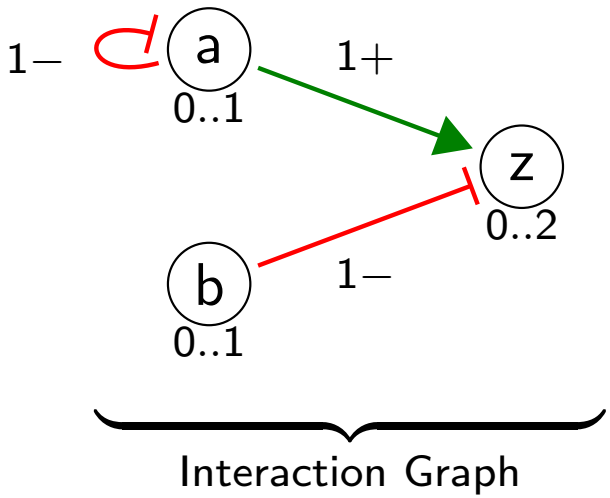
Nodes: genes

- Name a, b, z
- Possible values (levels of expression) 0..1, 0..2

Edges: interactions

- Threshold 1
- Type (activation or inhibition) $+$ / $-$

Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
−	+	1
−	−	0
+	+	2
+	−	1

ω	$k_{a,\omega}$
a	
+	1
−	0

$k_{b,\omega}$
1

Interaction Graph: structure of the system (genes & interactions)

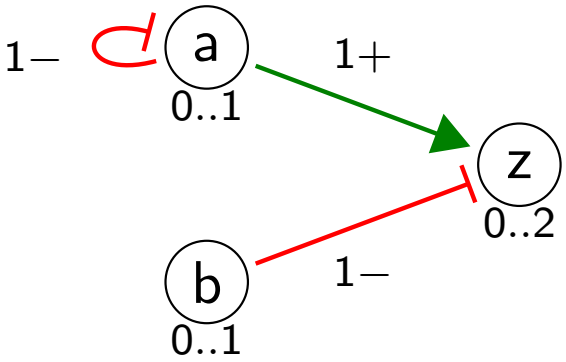
Nodes: genes

- Name a, b, z
- Possible values (levels of expression) 0..1, 0..2

Edges: interactions

- Threshold 1
- Type (activation or inhibition) $+$ / $-$

Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
$-$	$+$	1
$-$	$-$	0
$+$	$+$	2
$+$	$-$	1

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Parametrization

Parametrization: strength of the influences (cooperations)

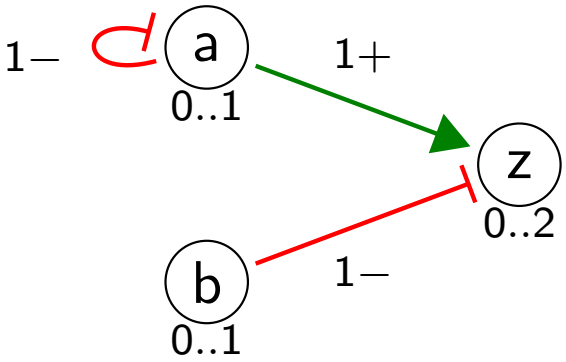
Maps of tendencies for each gene

→ To any **influences of predecessors** ω

→ Corresponds a **parameter** $k_{x,\omega}$

$k_{z,\{a^+,b^+\}} = 2$ means: z tends to 2 when $a \geq 1$ and $b < 1$

Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
$-$	$+$	1
$-$	$-$	0
$+$	$+$	2
$+$	$-$	1

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Parametrization

Parametrization: strength of the influences (cooperations)

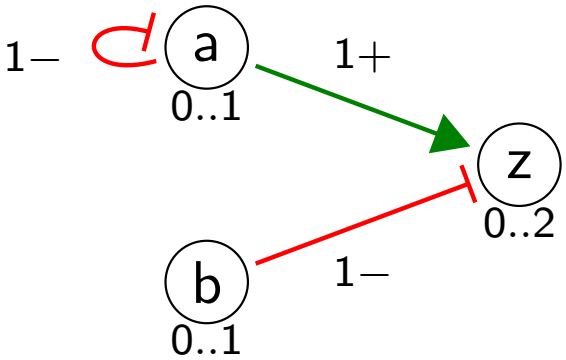
Maps of tendencies for each gene

→ To any **influences of predecessors** ω

→ Corresponds a **parameter** $k_{x,\omega}$

$k_{z,\{a^+,b^+\}} = 2$ means: z tends to 2 when $a \geq 1$ and $b < 1$

Biological Regulatory Network (Thomas' modeling)



ω		$k_{z,\omega}$
a	b	
−	+	1
−	−	0
+	+	2
+	−	1

ω	$k_{a,\omega}$
a	
+	1
−	0

$k_{b,\omega}$
1

Biological Regulatory Network

→ All needed information to run the model or study its dynamics:
- Build the State Graph
- Find reachability properties, fixed points, attractors
- Other properties...

→ **Strengths**: well adapted for the study of biological systems

→ **Drawbacks**: inherent complexity; needs the full specification of cooperations

Overview

Contribution

Scientific challenge

How can we cope with the analysis of **large-scale systems**, involving up to 1.000 interacting components?

Objectives of this part

- Introduce a Process Algebra inspired framework based on a compact representation of the interactions
- Develop efficient **static analysis approaches** to answer most common problems
- Apply the methodology to large-scale biological regulatory networks

Joint work with

- L. Paulevé (ETH Zurich), M. Folschette, O. Roux (IRCCyN)
- K. Inoue (NII)

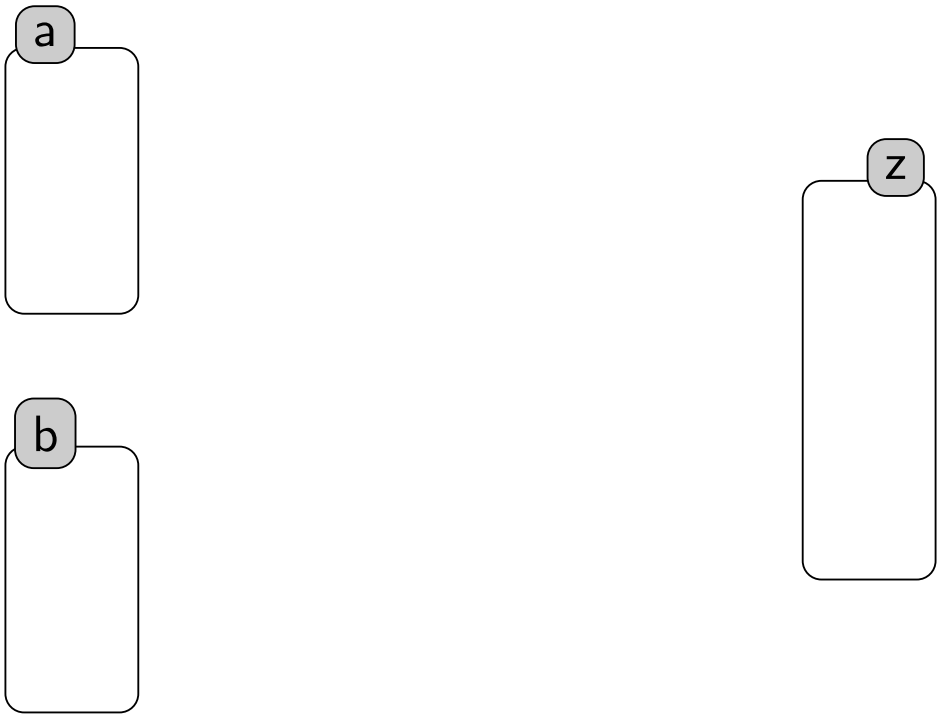
Intuitive principle of the Process Hitting framework

Process	=	component a at level i
Interaction	=	a at level i makes b at level j increase or decrease to level k
denoted		$a_i \to b_j \Uparrow b_k$ (hit and bounce)

Definition (Interaction and Retroaction)

Interaction $(a_i \to b_j \Uparrow b_k)$, where a_i is the level of a process a and $b_j \neq b_k$,
Retroaction $(a_i \to a_i \Uparrow a_k)$: when $a_i = b_j$.

The Process Hitting modeling



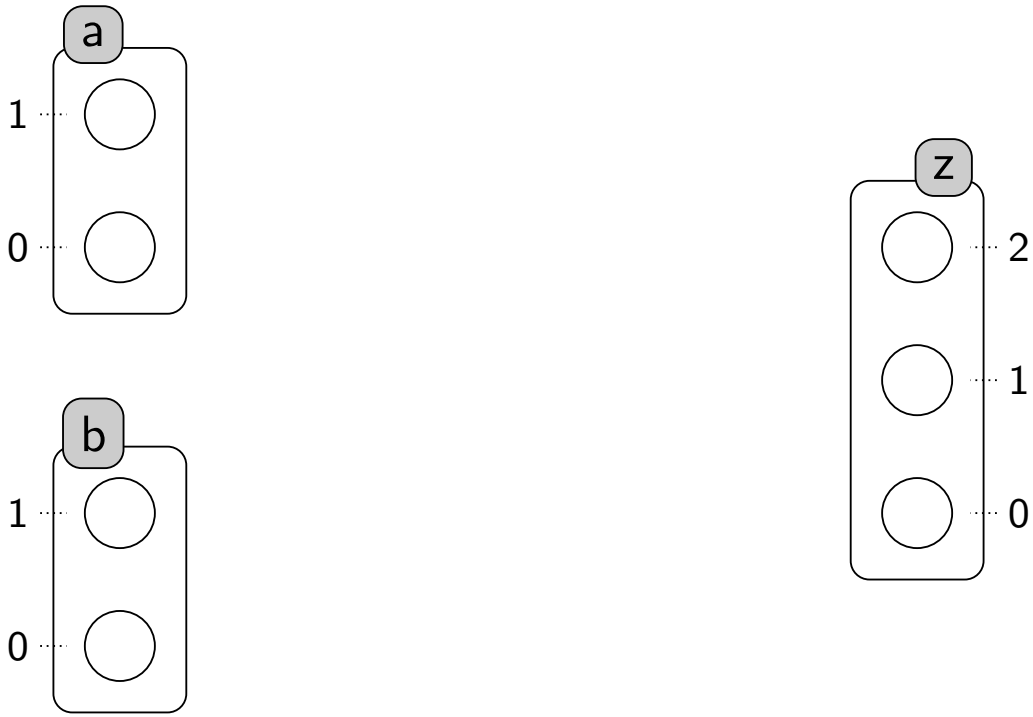
Sorts: components a, b, z

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

States: sets of active processes

Actions: dynamics $b_1 \to z_0 \restriction z_1$, $a_0 \to a_0 \restriction a_1$, $a_1 \to z_1 \restriction z_2$

The Process Hitting modeling



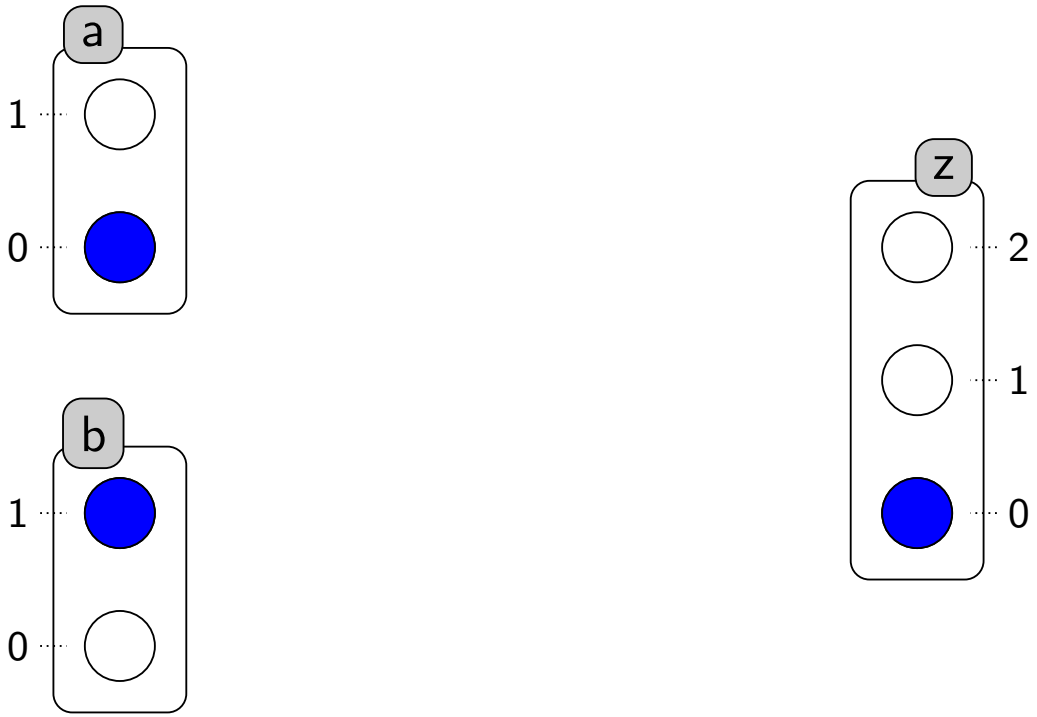
Sorts: components a, b, z

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

States: sets of active processes

Actions: dynamics $b_1 \to z_0 \Rsh z_1$, $a_0 \to a_0 \Rsh a_1$, $a_1 \to z_1 \Rsh z_2$

The Process Hitting modeling



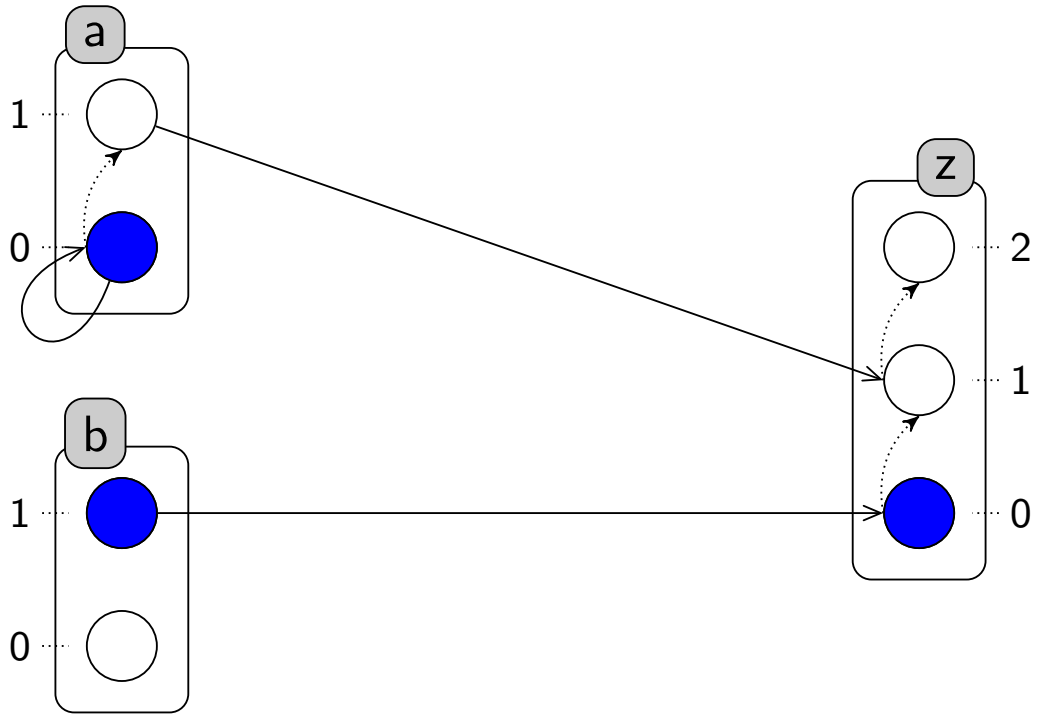
Sorts: components a, b, z

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

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

Actions: dynamics $b_1 \to z_0 \Uparrow z_1$, $a_0 \to a_0 \Uparrow a_1$, $a_1 \to z_1 \Uparrow z_2$

The Process Hitting modeling



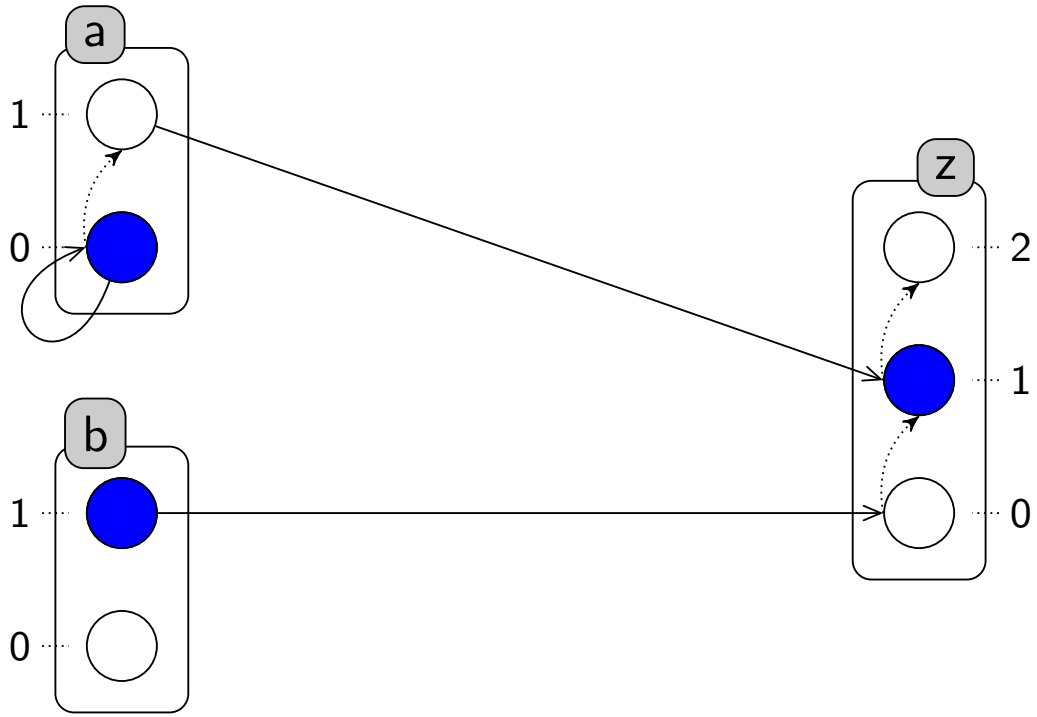
Sorts: components a, b, z

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

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

Actions: dynamics $\underline{b_1 \to z_0 \uparrow z_1}$, $\underline{a_0 \to a_0 \uparrow a_1}$, $a_1 \to z_1 \uparrow z_2$

The Process Hitting modeling



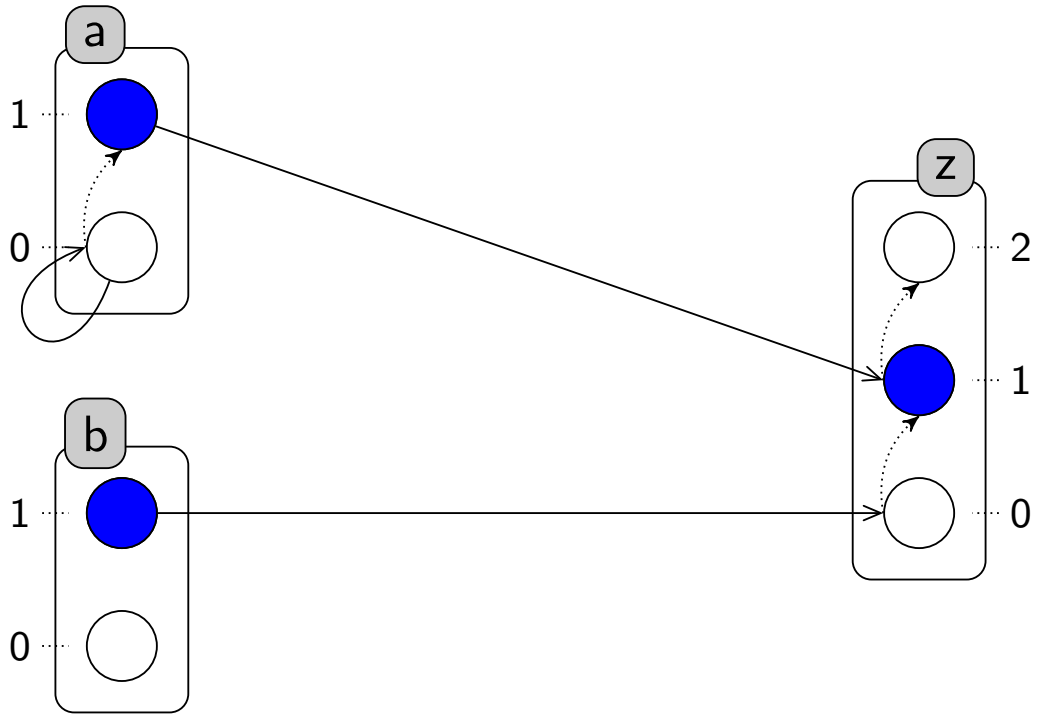
Sorts: components a, b, z

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

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

Actions: dynamics $b_1 \to z_0 \Rsh z_1$, $a_0 \to a_0 \Rsh a_1$, $a_1 \to z_1 \Rsh z_2$

The Process Hitting modeling



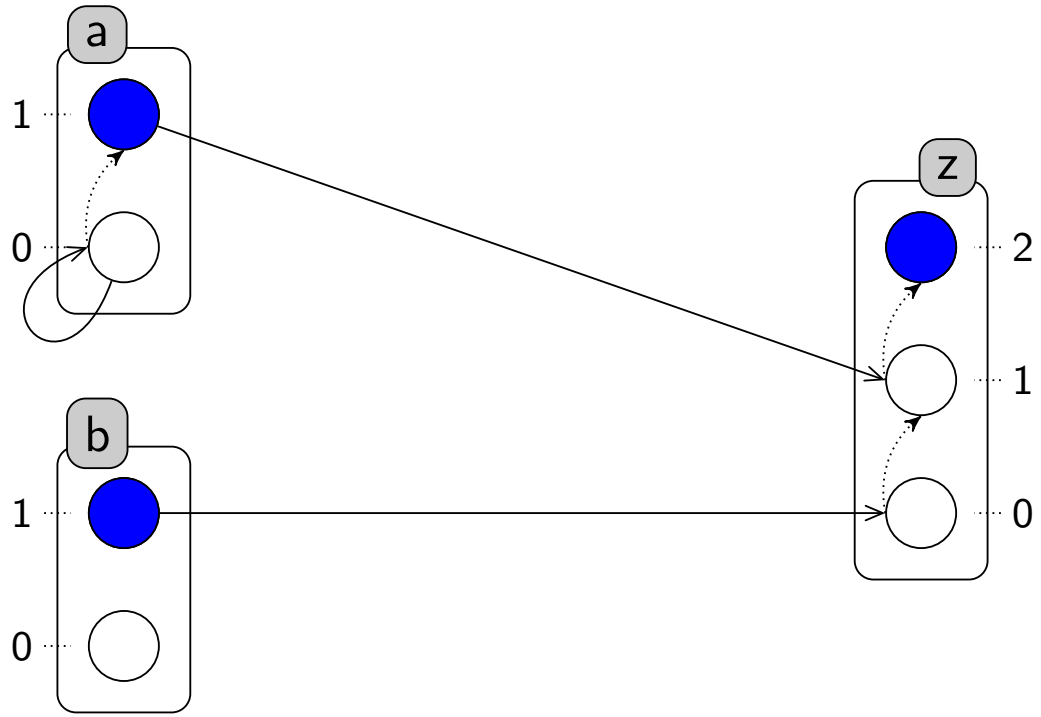
Sorts: components a, b, z

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

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

Actions: dynamics $b_1 \to z_0 \Rsh z_1$, $a_0 \to a_0 \Rsh a_1$, $a_1 \to z_1 \Rsh z_2$

The Process Hitting modeling



Sorts: components a, b, z

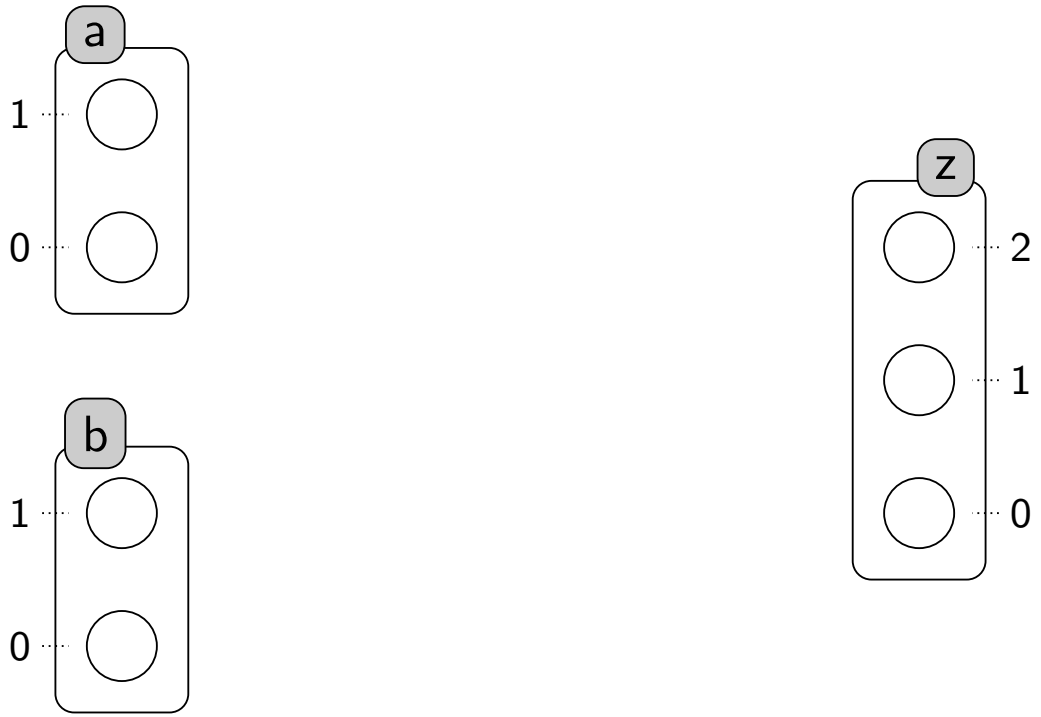
Processes: local states / levels of expression z_0, z_1, z_2

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

Actions: dynamics $b_1 \to z_0 \Rsh z_1$, $a_0 \to a_0 \Rsh a_1$, $a_1 \to z_1 \Rsh z_2$

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Rsh z_2$

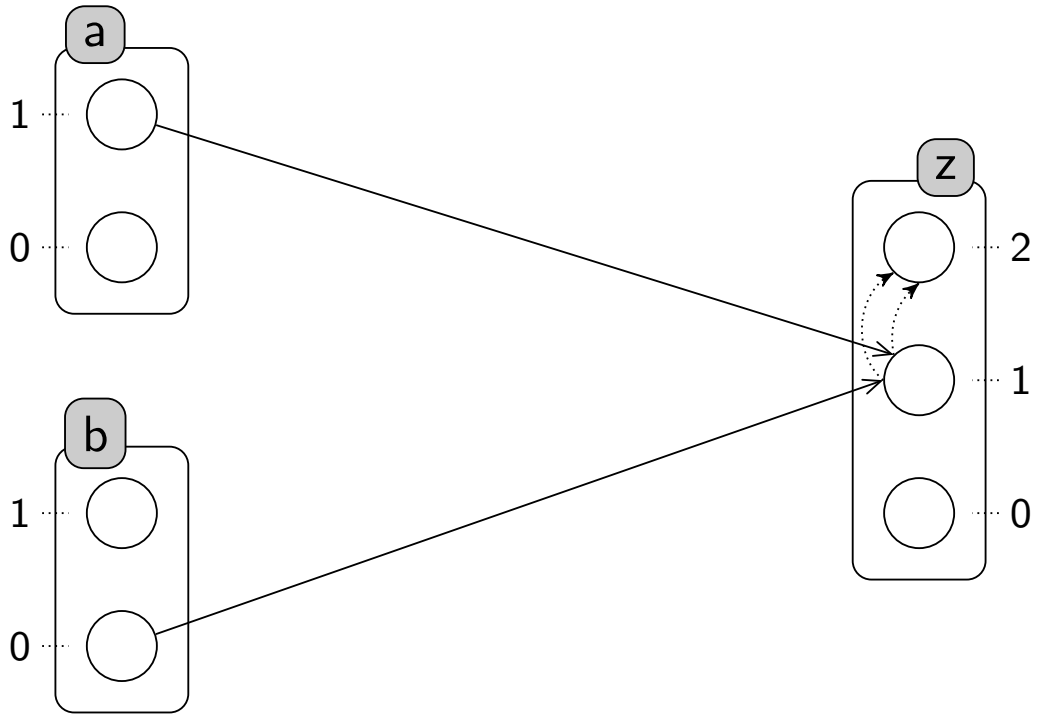
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Uparrow z_2$

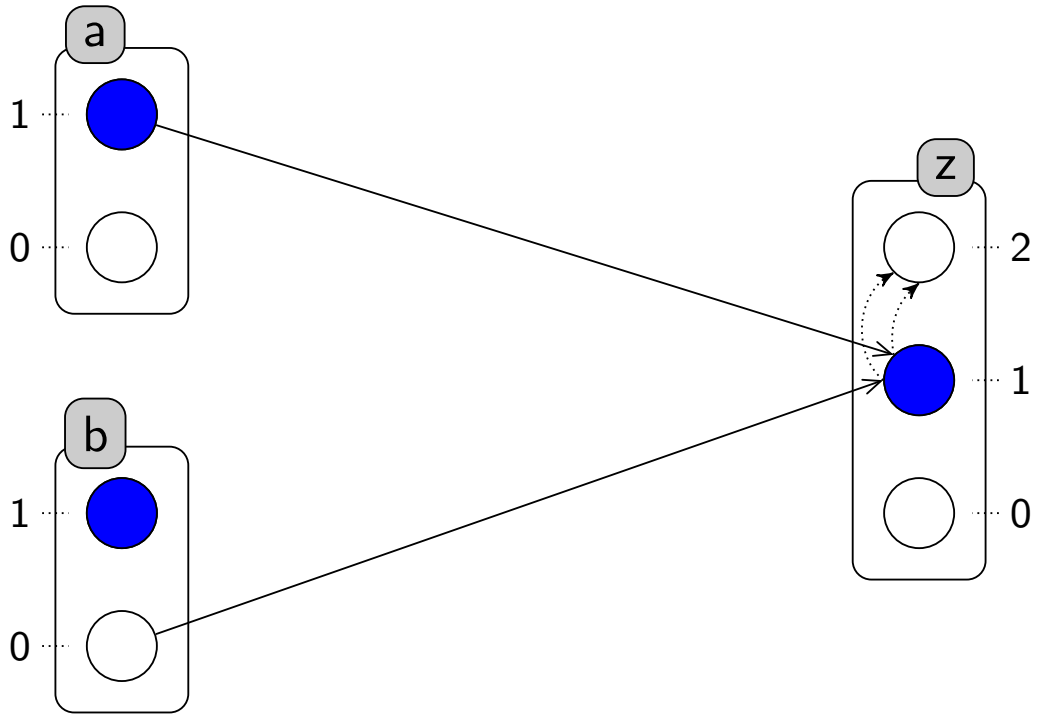
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \uparrow z_2$

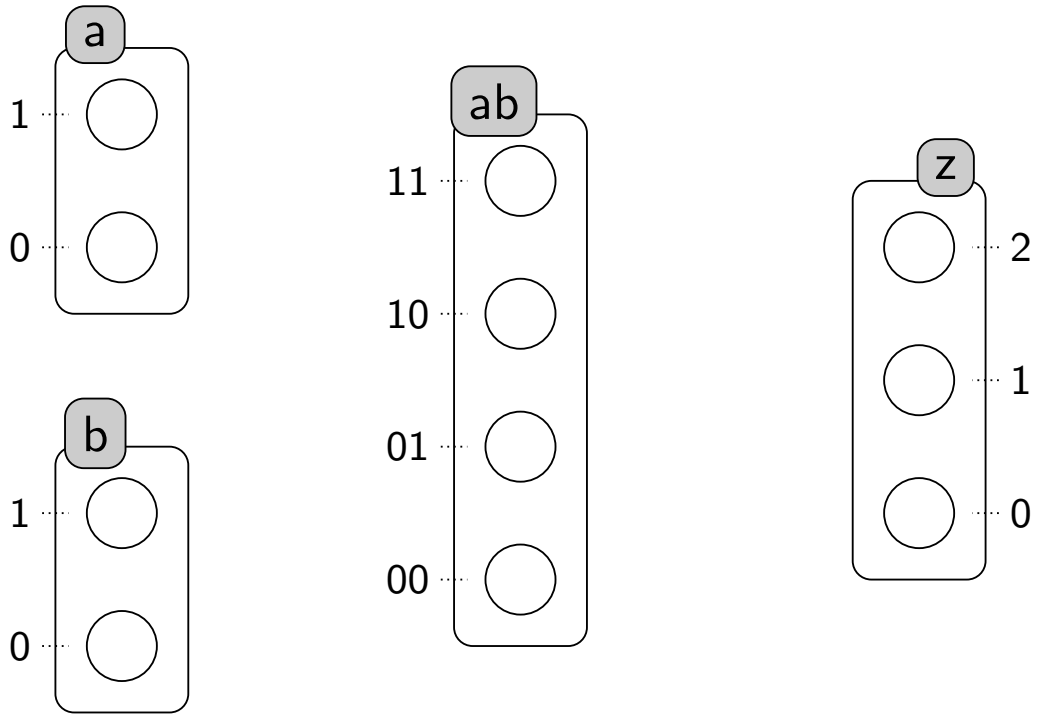
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Rsh z_2$

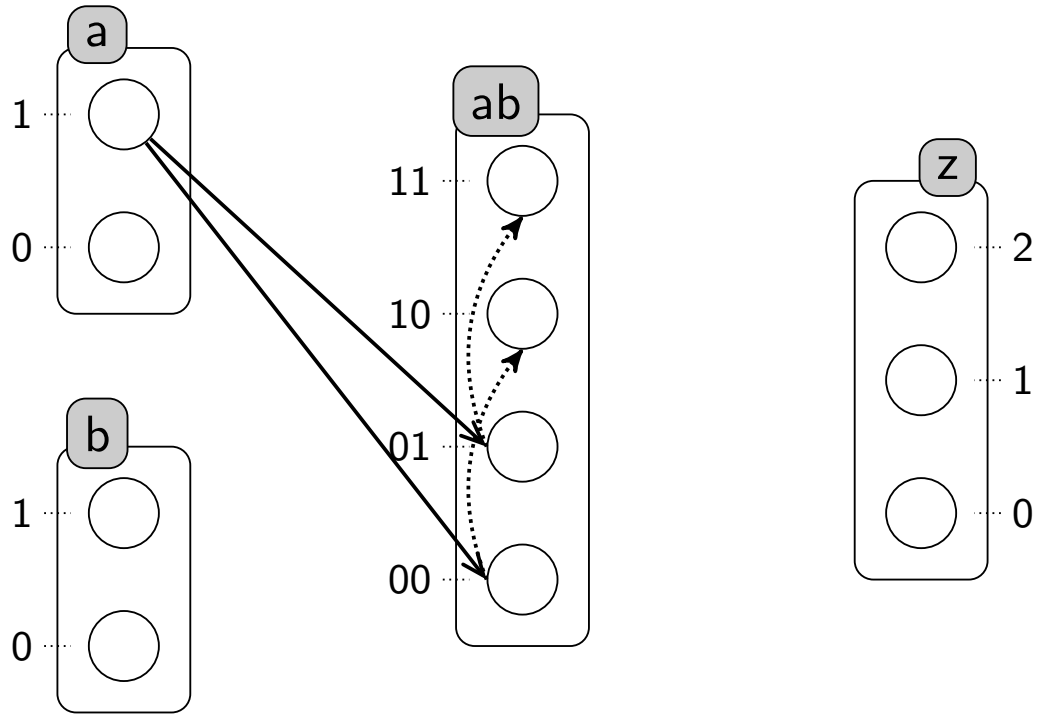
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\langle \underline{a_1, b_0} \rangle \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Rsh z_2$

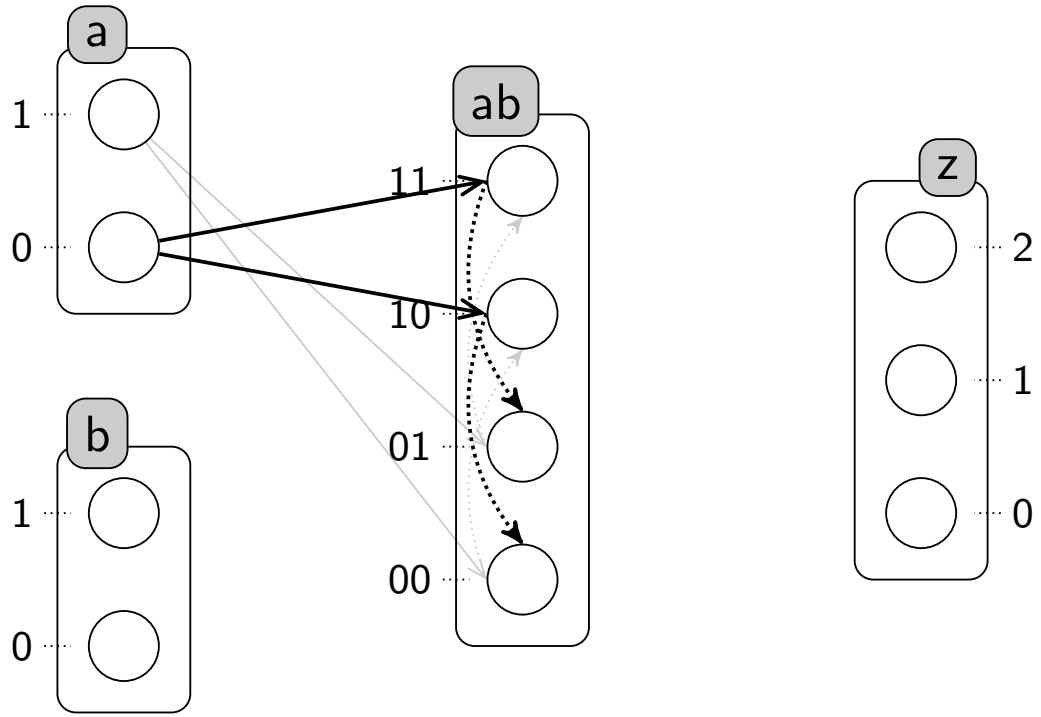
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\langle \underline{a_1, b_0} \rangle \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Uparrow z_2$

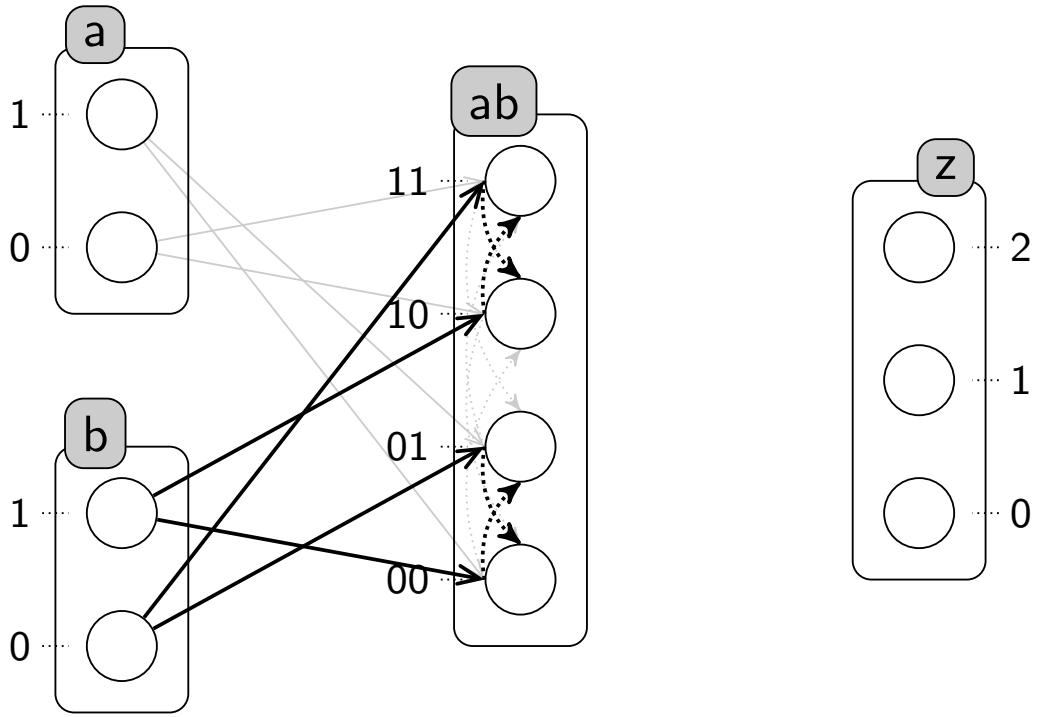
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\langle \underline{a_1, b_0} \rangle \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \uparrow z_2$

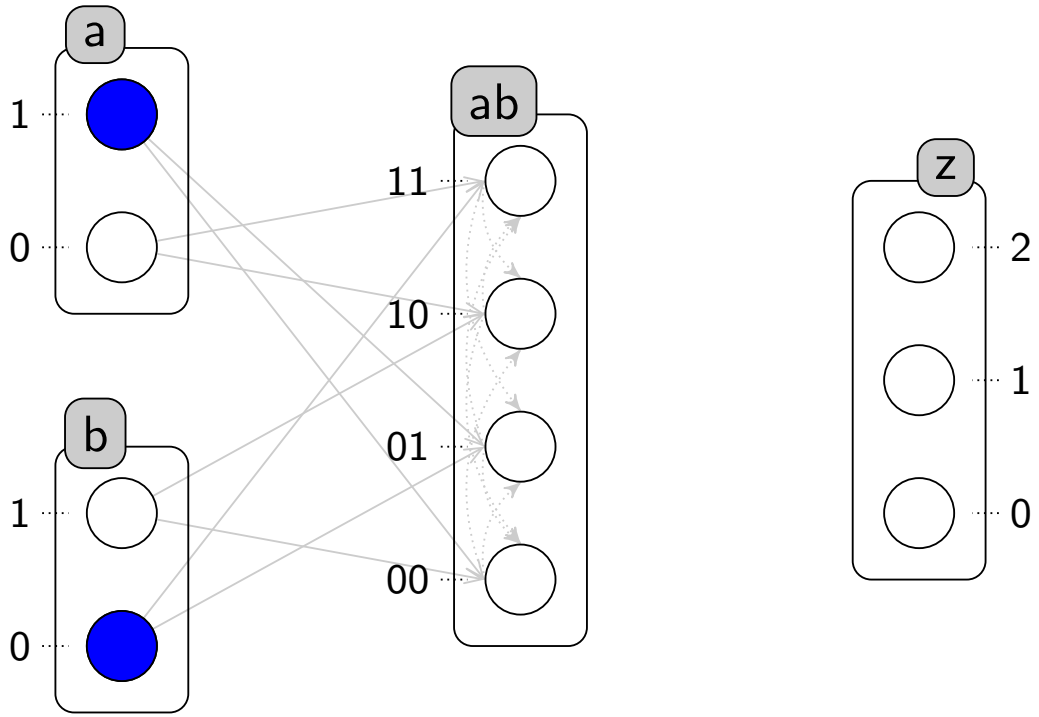
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \uparrow z_2$

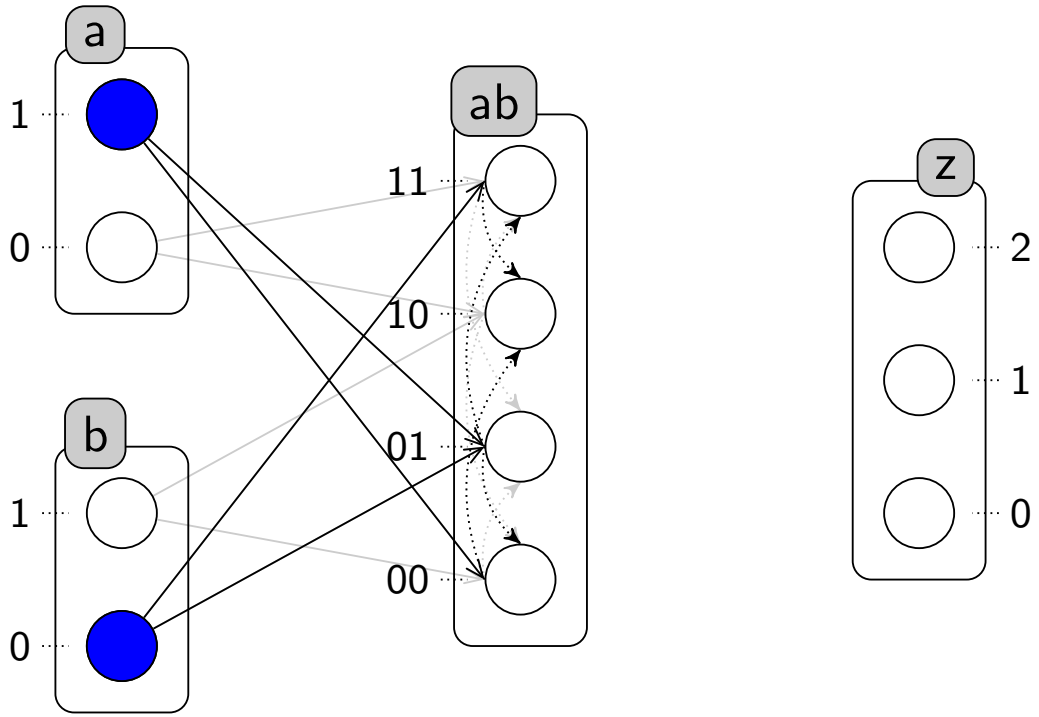
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Uparrow z_2$

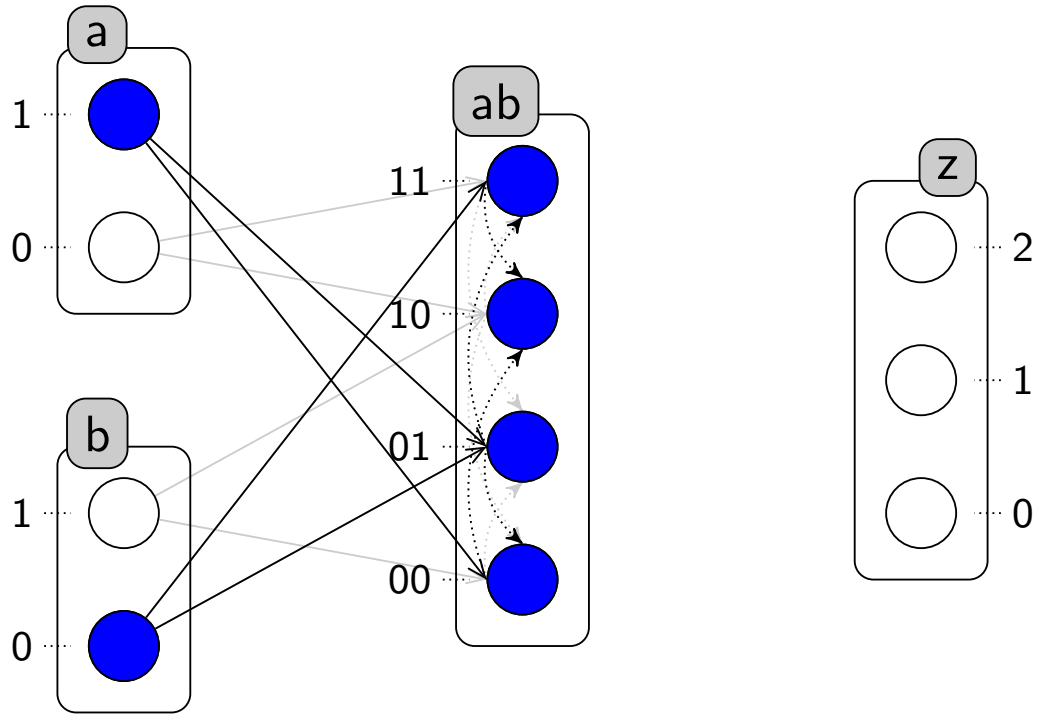
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \uparrow z_2$

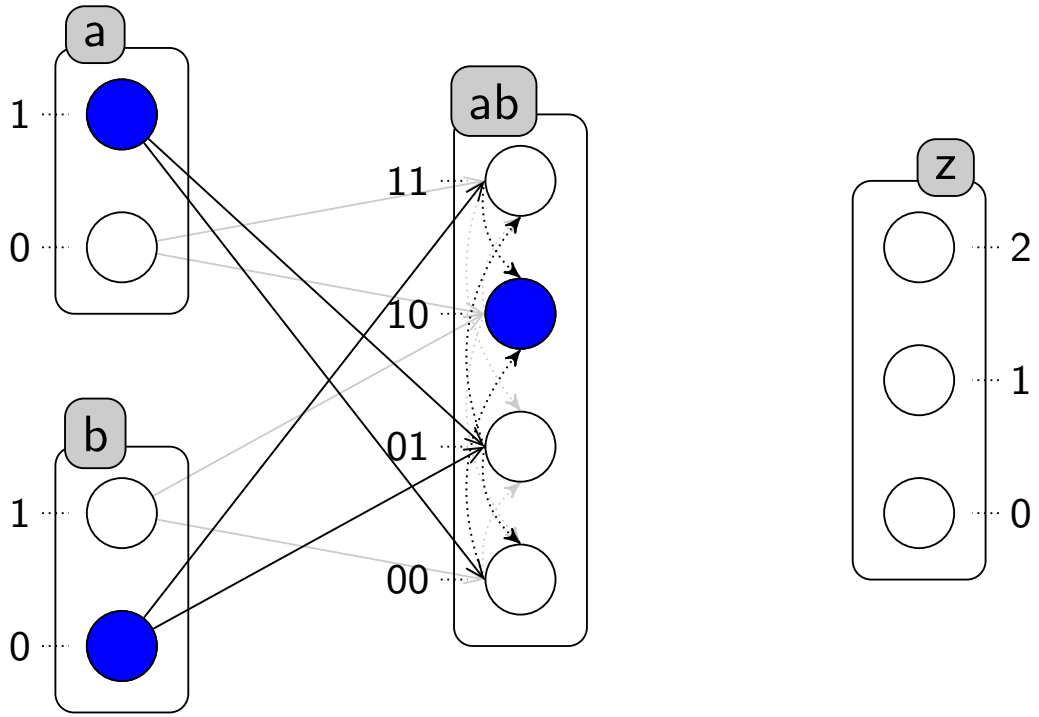
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \uparrow z_2$

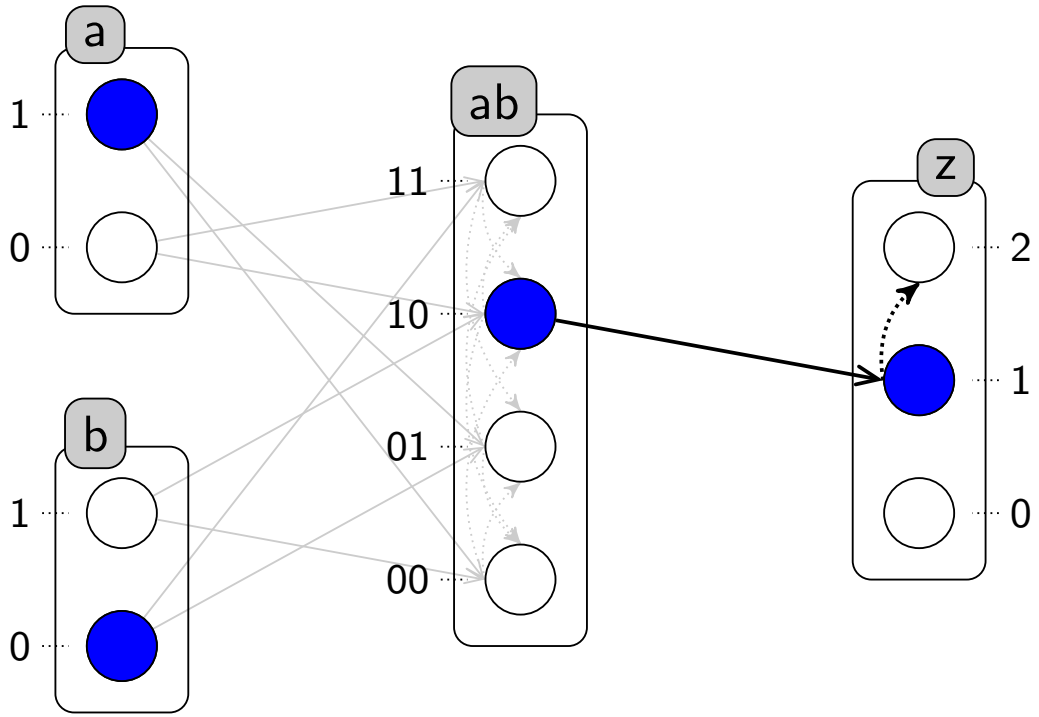
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Uparrow z_2$

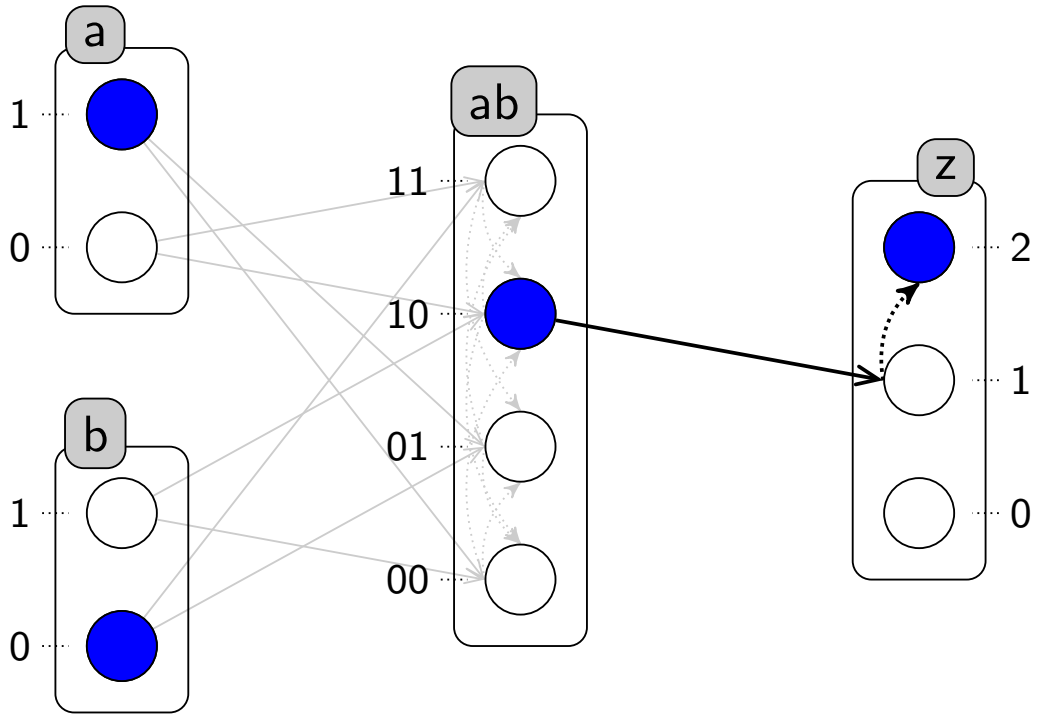
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \Uparrow z_2$

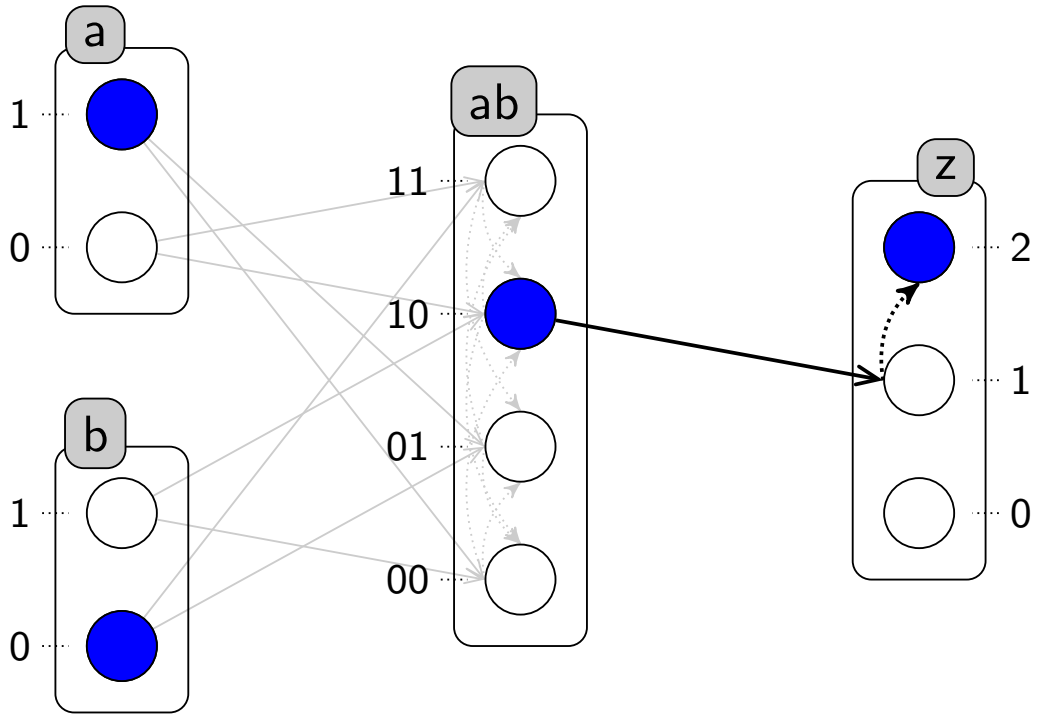
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Adding cooperations

[PMR12]



How to introduce some **cooperation** between sorts? $\underline{a_1 \wedge b_0} \rightarrow z_1 \uparrow z_2$

Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_0}$

Constraint: represent each configuration $\underline{\langle a_1, b_0 \rangle} \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

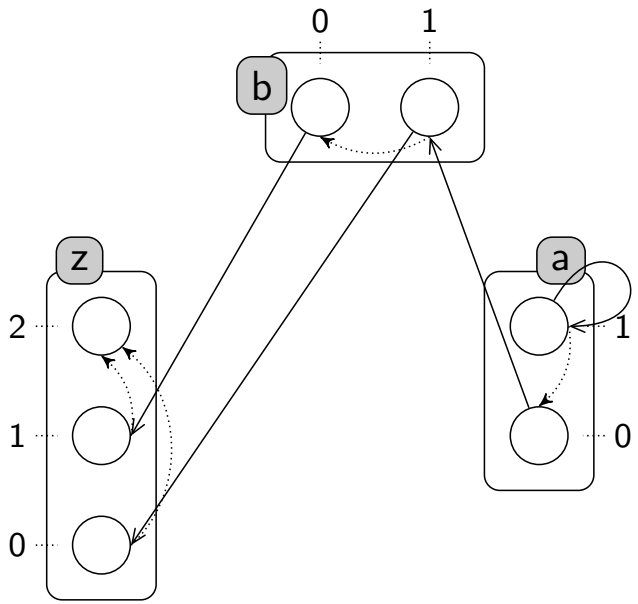
Static Analysis: Fixed Points

[PMR11]

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

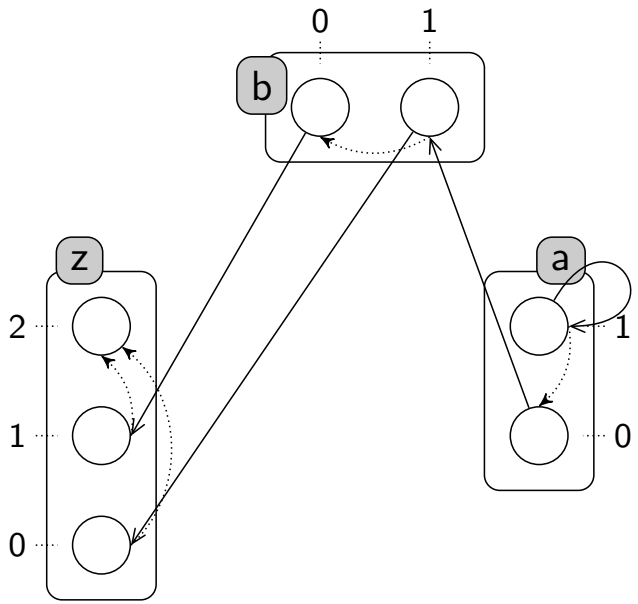
Static Analysis: Fixed Points

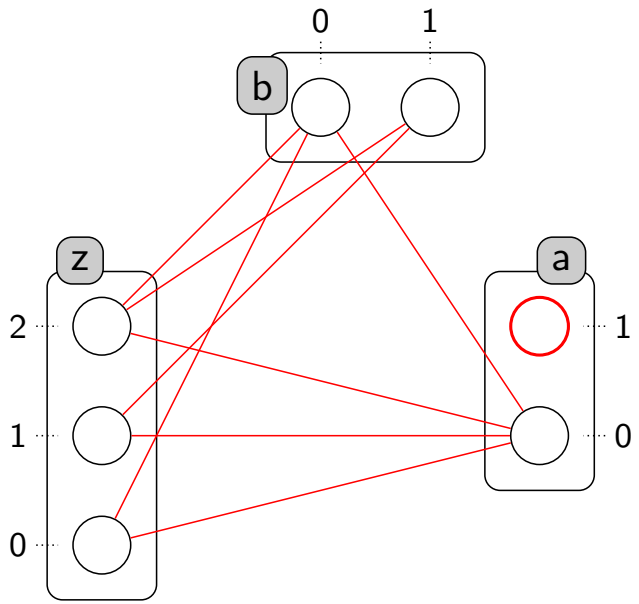
[PMR11]

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

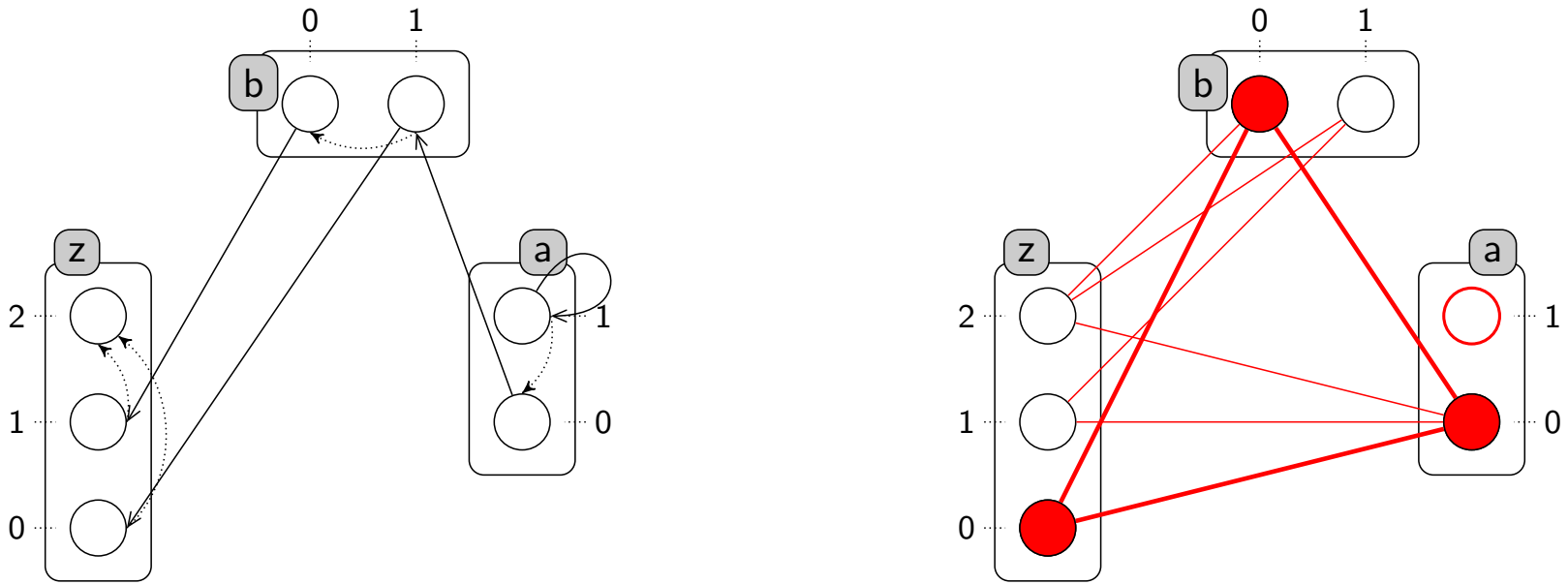
Static Analysis: Fixed Points

[PMR11]

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

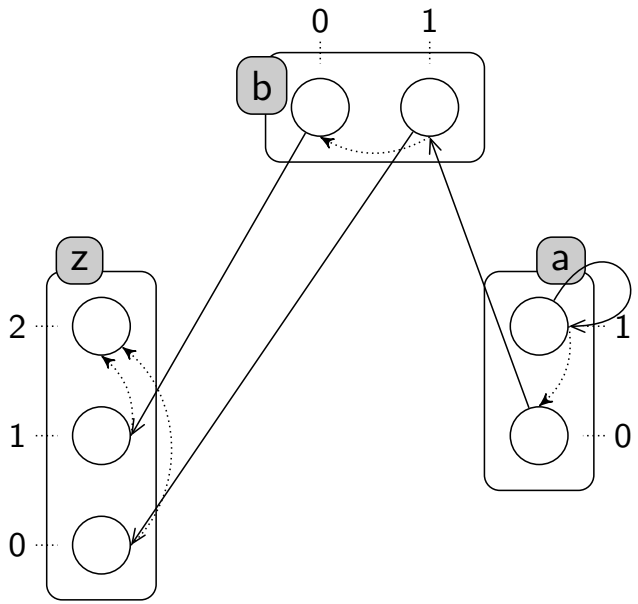
Static Analysis: Fixed Points

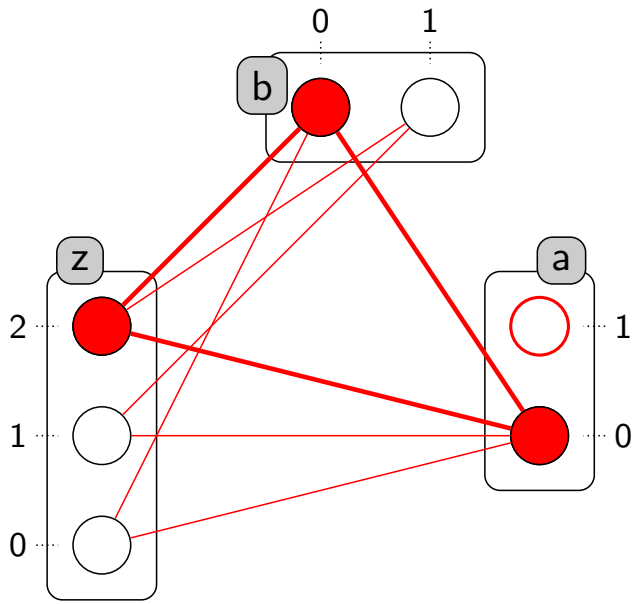
[PMR11]

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

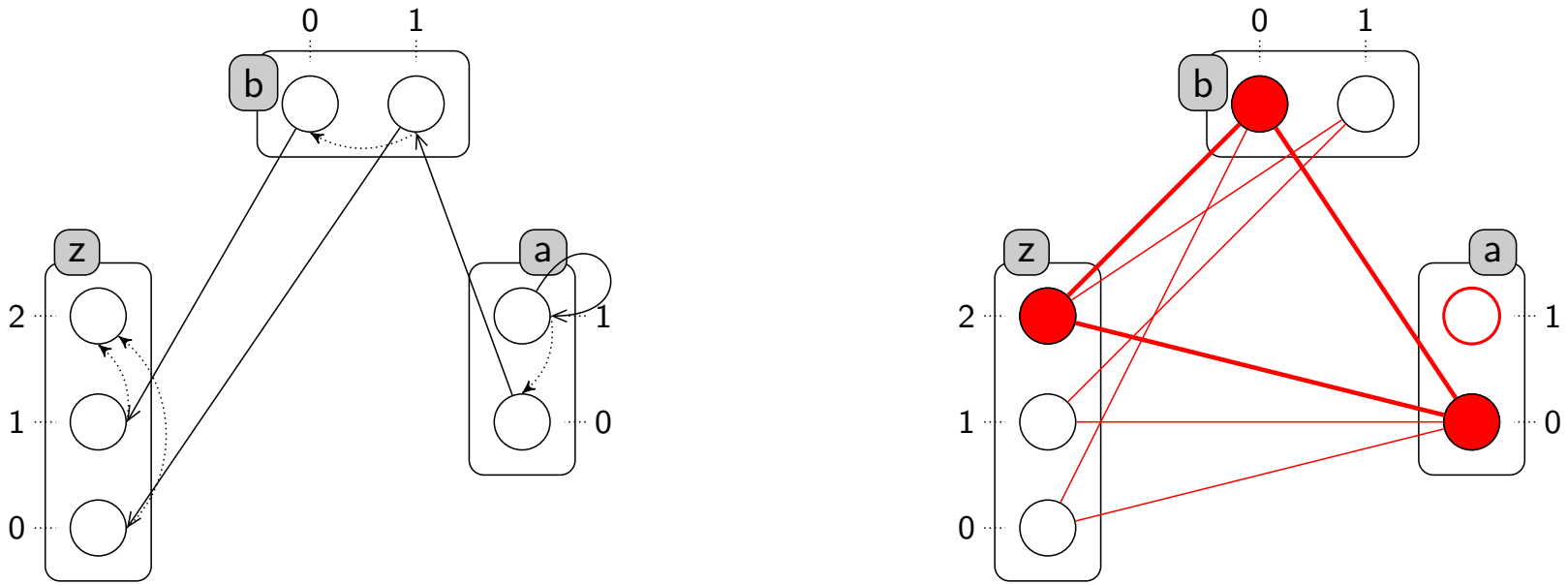
Static Analysis: Fixed Points

[PMR11]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

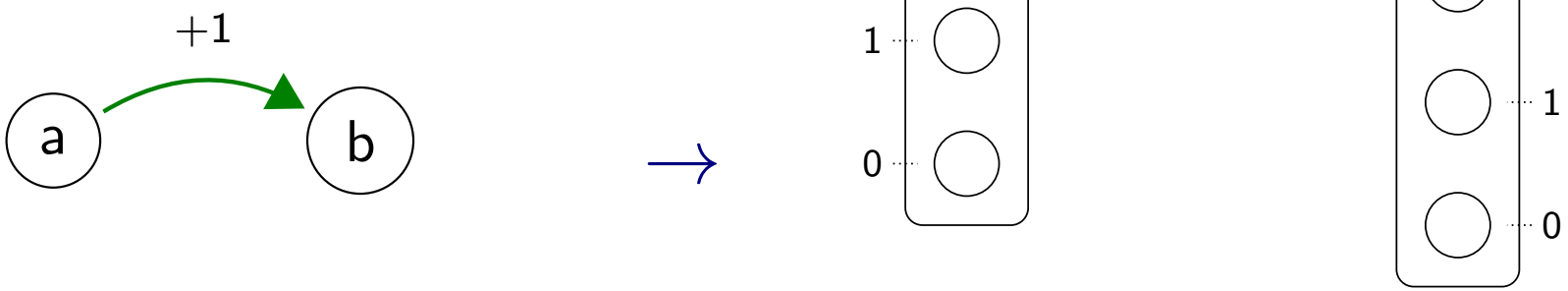
→ Hitless Graph → **n-cliques** = fixed points



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

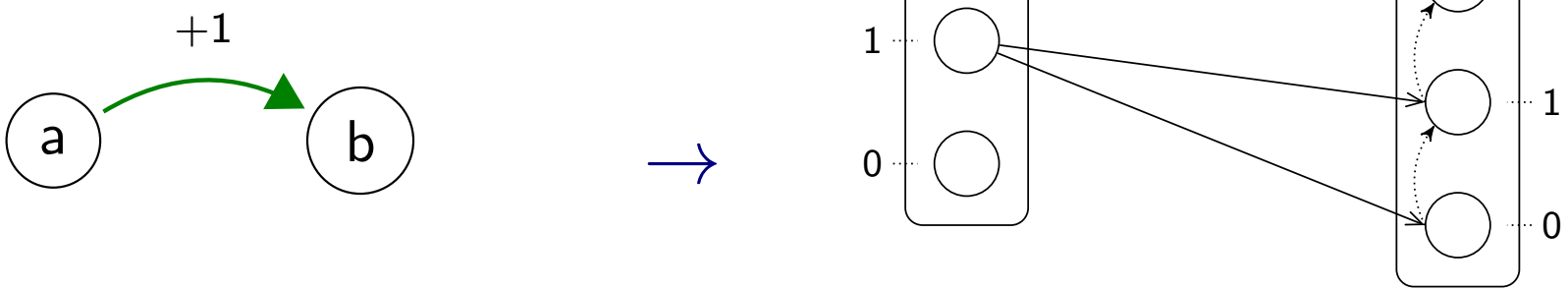
Translation of the Generalized Dynamics

Positive interaction:



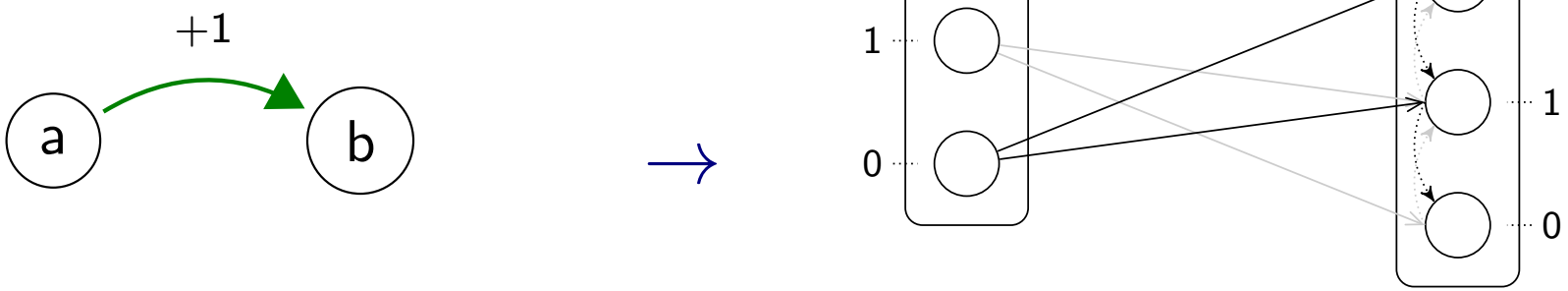
Translation of the Generalized Dynamics

Positive interaction:



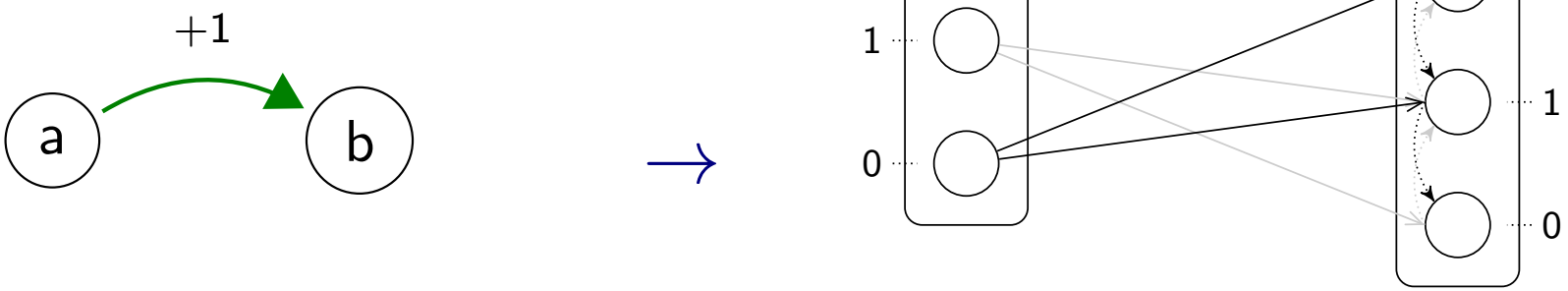
Translation of the Generalized Dynamics

Positive interaction:

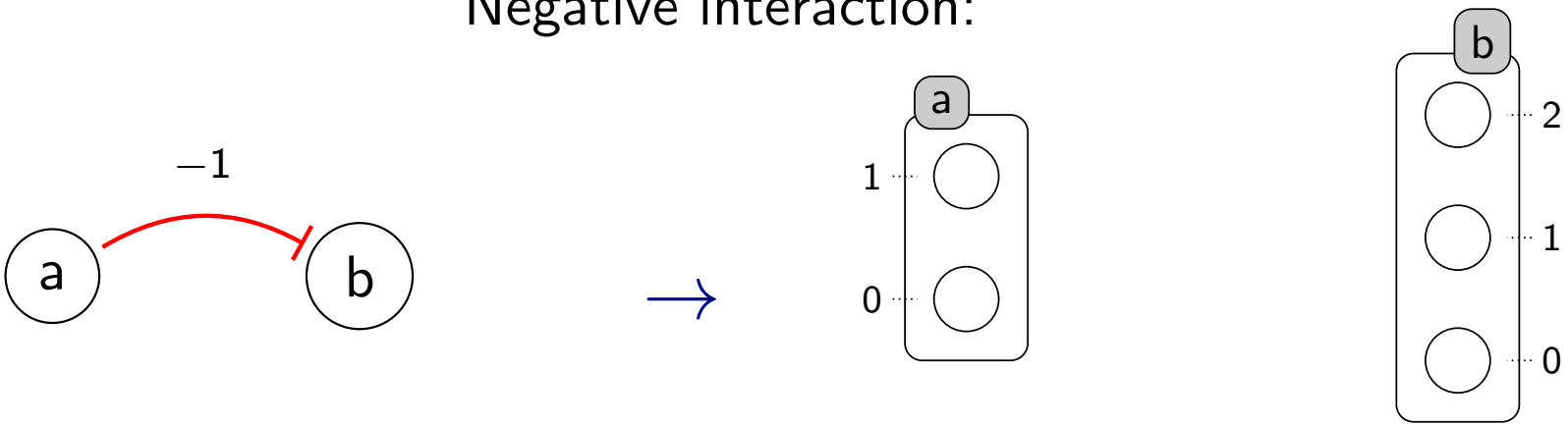


Translation of the Generalized Dynamics

Positive interaction:

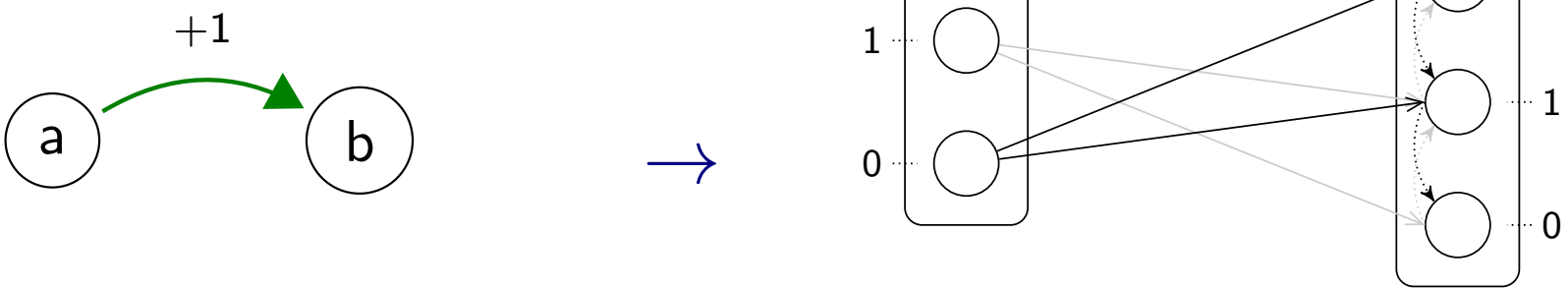


Negative interaction:

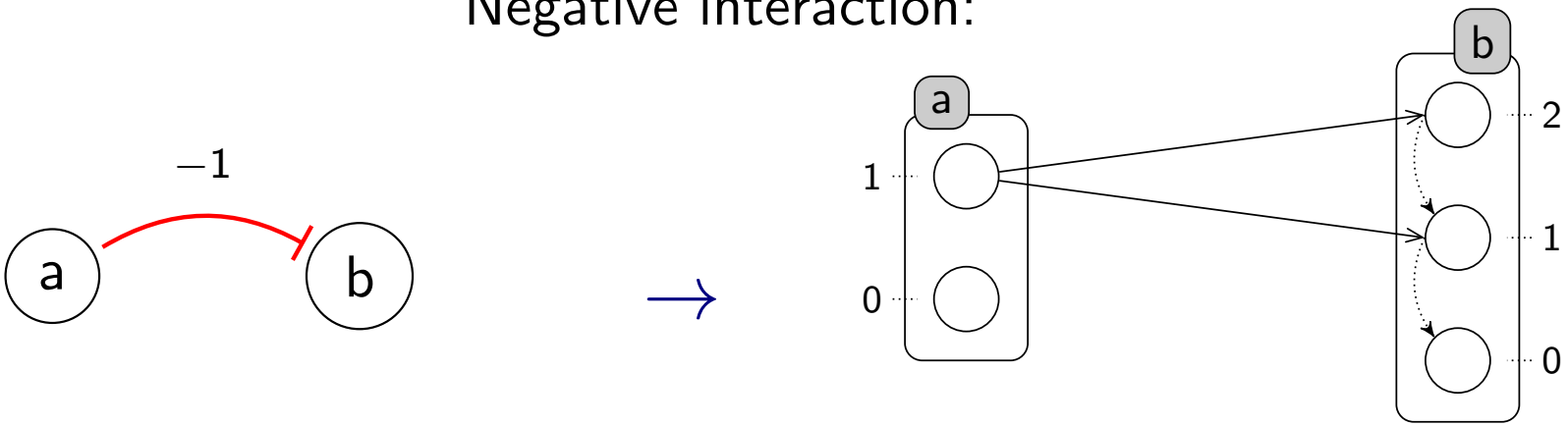


Translation of the Generalized Dynamics

Positive interaction:

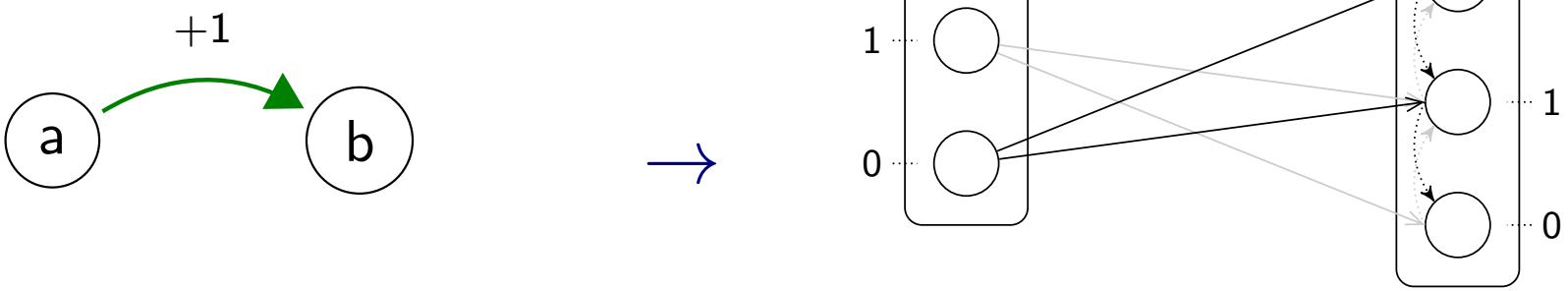


Negative interaction:

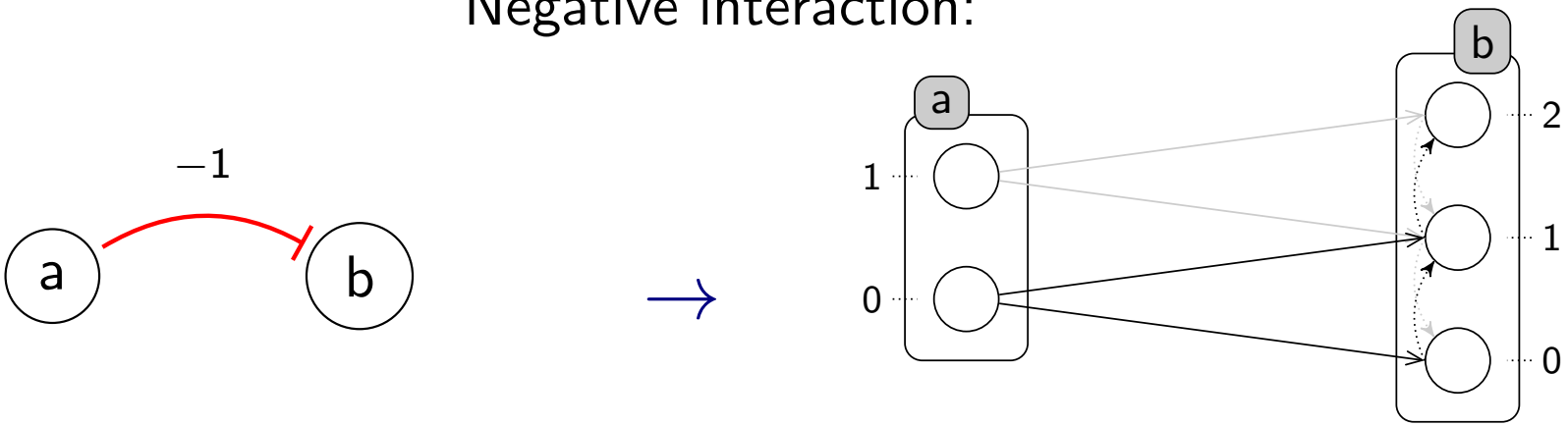


Translation of the Generalized Dynamics

Positive interaction:



Negative interaction:

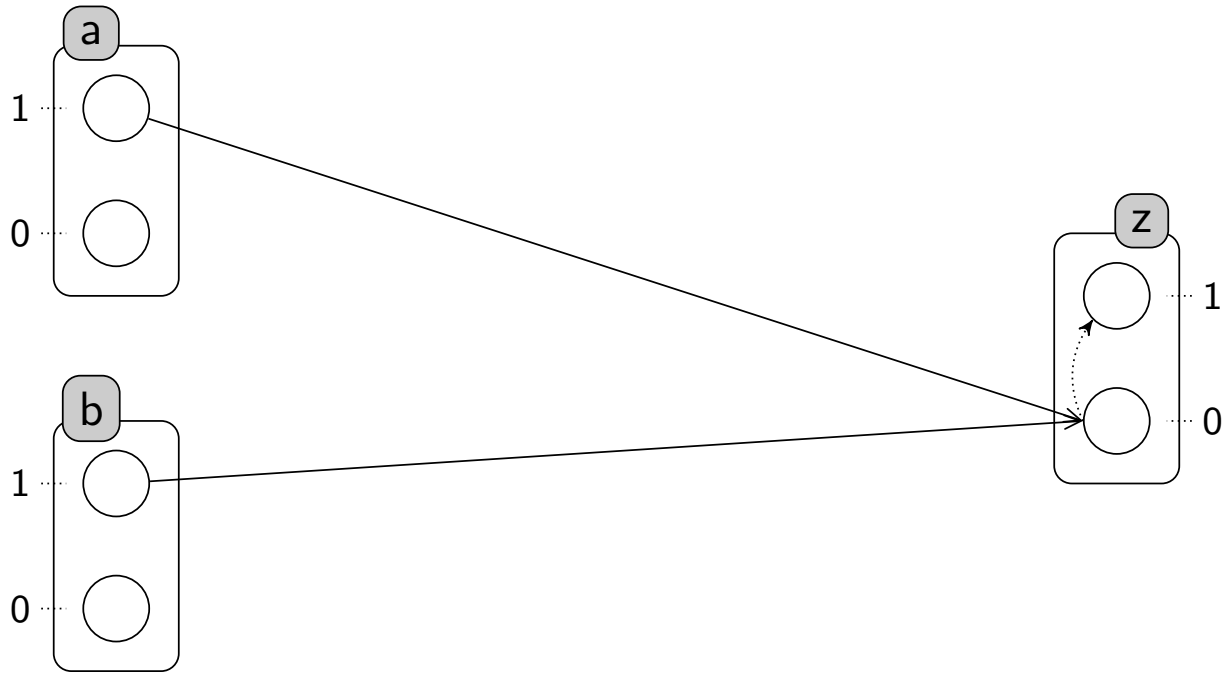


Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \to z_0 \overset{\uparrow}{\uparrow} z_1$?

$\to$ Add a **cooperative sort** reflecting the state of a and b



$\to$ Introduces a temporal shift (over-approximation)

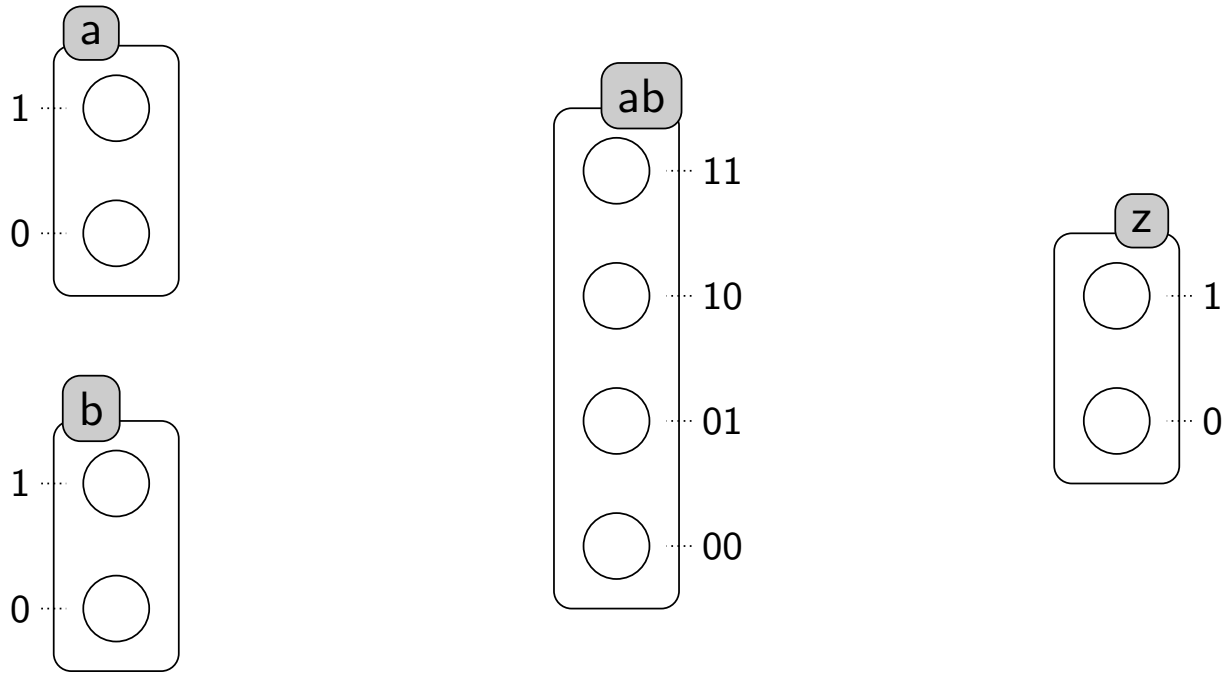
$\to$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \to z_0 \overset{\uparrow}{} z_1$?

$\to$ Add a **cooperative sort** reflecting the state of a and b



$\to$ Introduces a temporal shift (over-approximation)

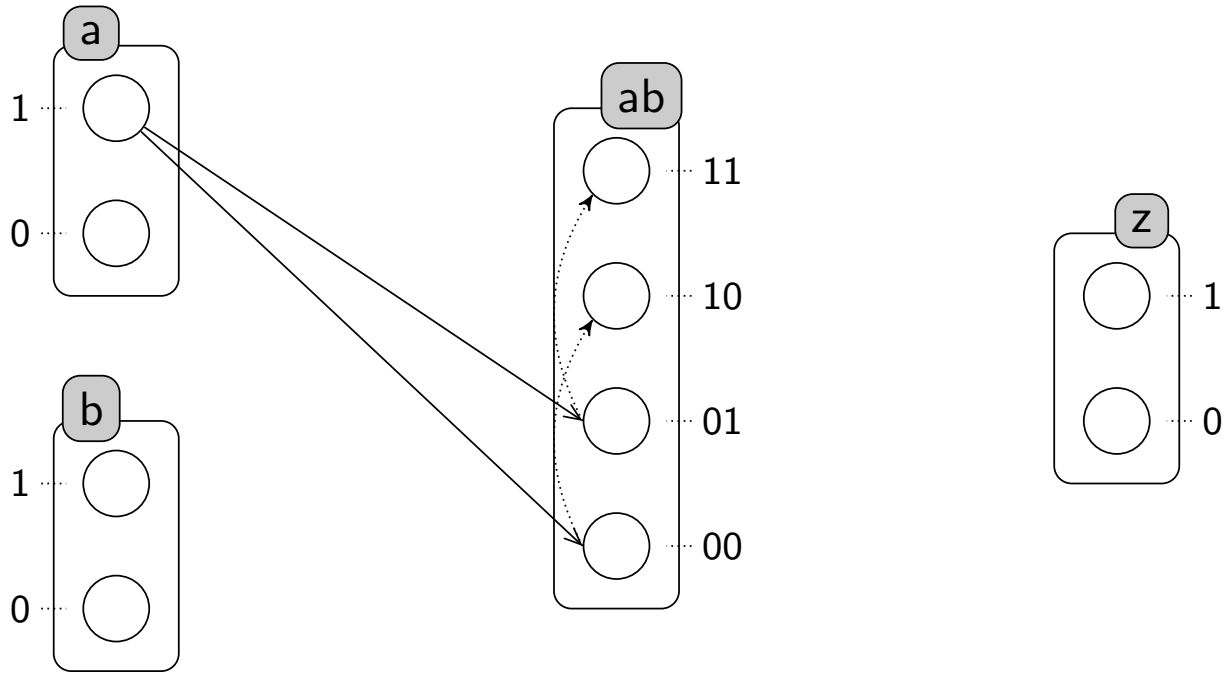
$\to$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \rightarrow z_0 \overset{\uparrow}{\uparrow} z_1$?

$\rightarrow$ Add a **cooperative sort** reflecting the state of a and b



$\rightarrow$ Introduces a temporal shift (over-approximation)

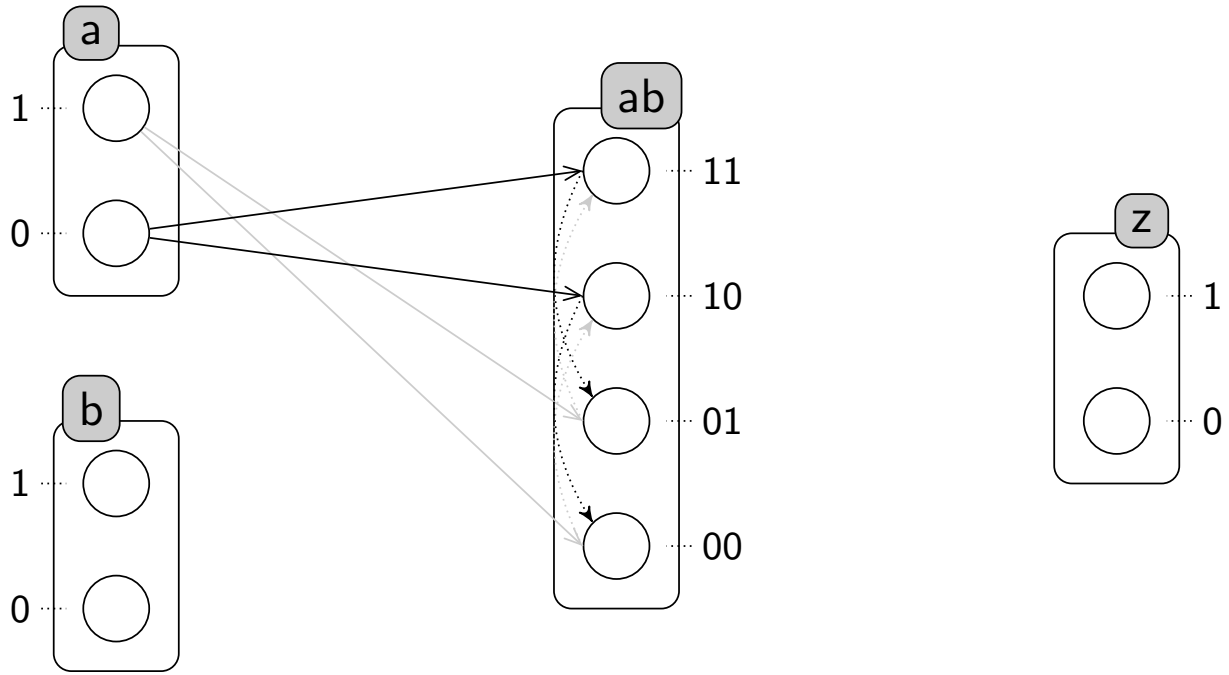
$\rightarrow$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \rightarrow z_0 \overset{\uparrow}{\uparrow} z_1$?

$\rightarrow$ Add a **cooperative sort** reflecting the state of a and b



$\rightarrow$ Introduces a temporal shift (over-approximation)

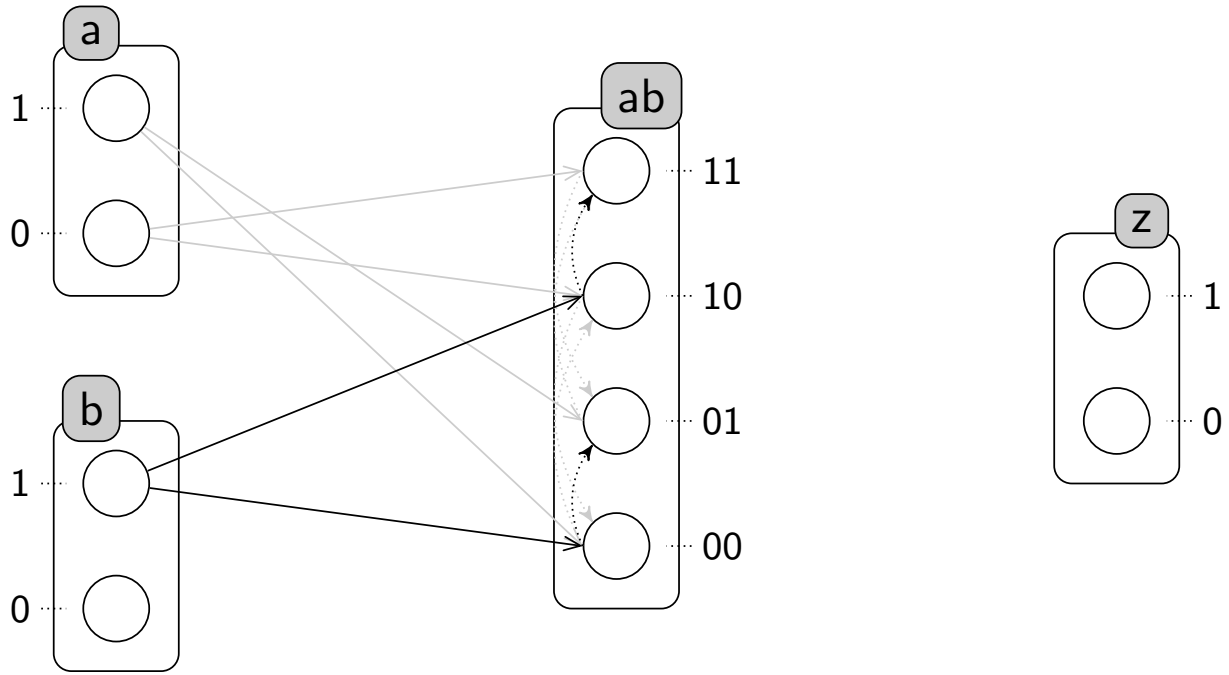
$\rightarrow$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \rightarrow z_0 \overset{\uparrow}{\uparrow} z_1$?

$\rightarrow$ Add a **cooperative sort** reflecting the state of a and b



$\rightarrow$ Introduces a temporal shift (over-approximation)

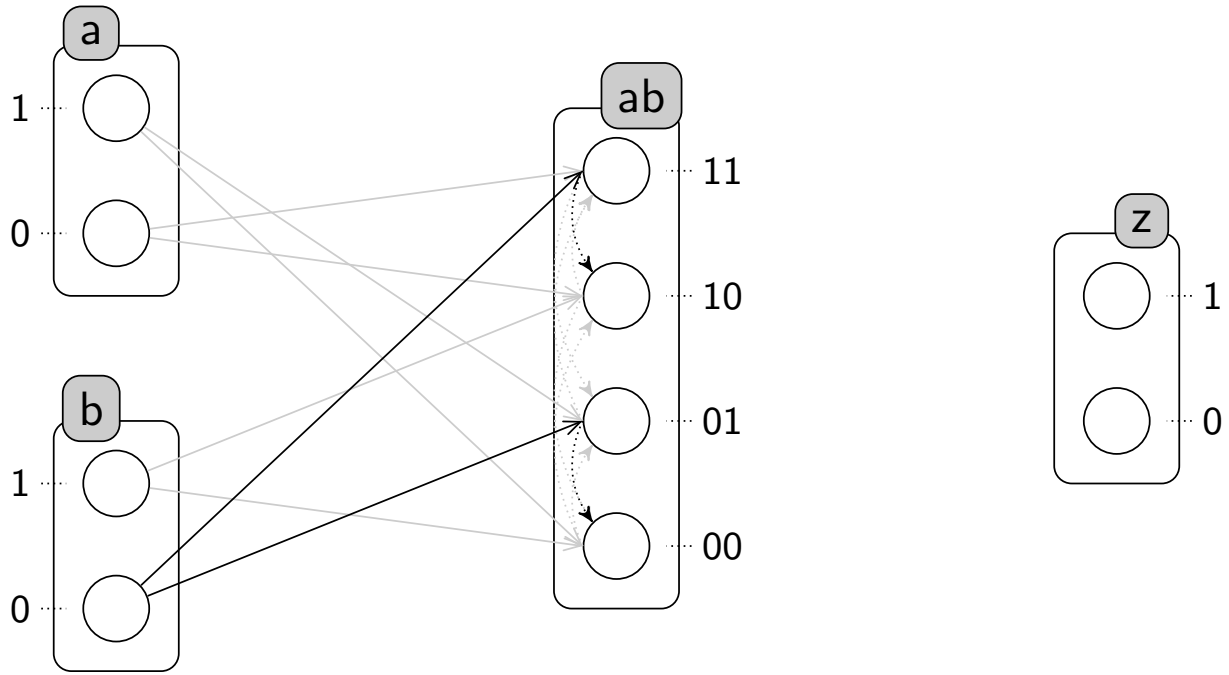
$\rightarrow$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \rightarrow z_0 \overset{\uparrow}{\uparrow} z_1$?

$\rightarrow$ Add a **cooperative sort** reflecting the state of a and b



$\rightarrow$ Introduces a temporal shift (over-approximation)

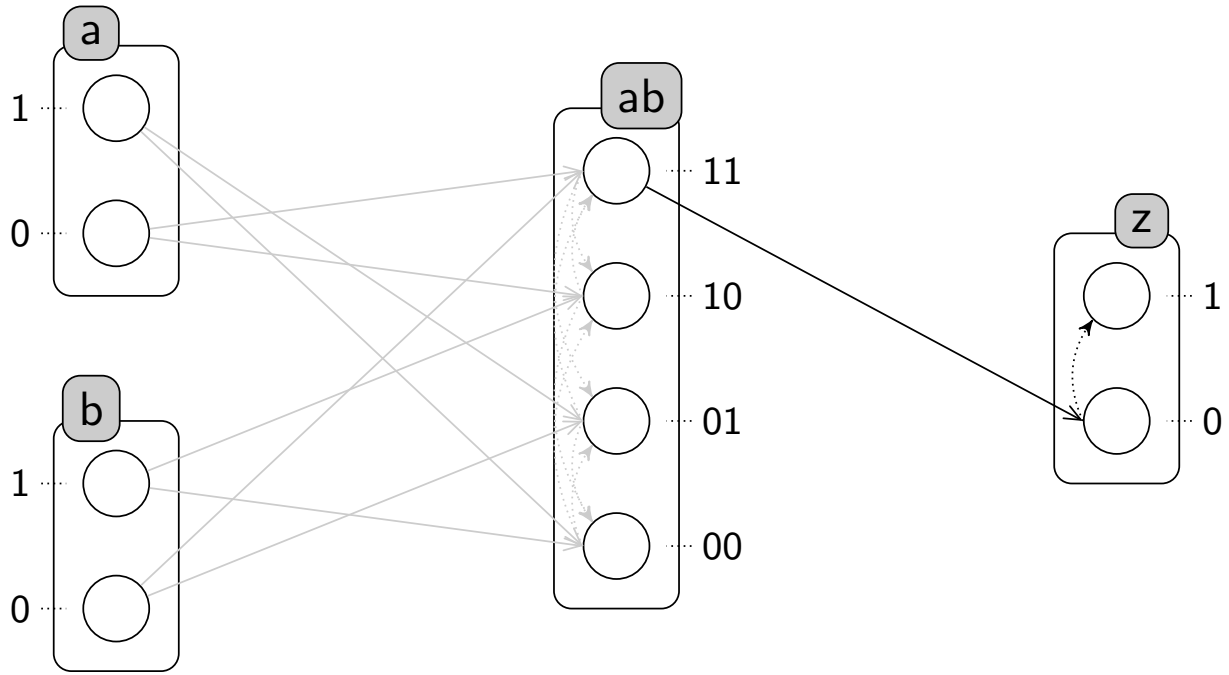
$\rightarrow$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \rightarrow z_0 \overset{\uparrow}{\uparrow} z_1$?

$\rightarrow$ Add a **cooperative sort** reflecting the state of a and b



$\rightarrow$ Introduces a temporal shift (over-approximation)

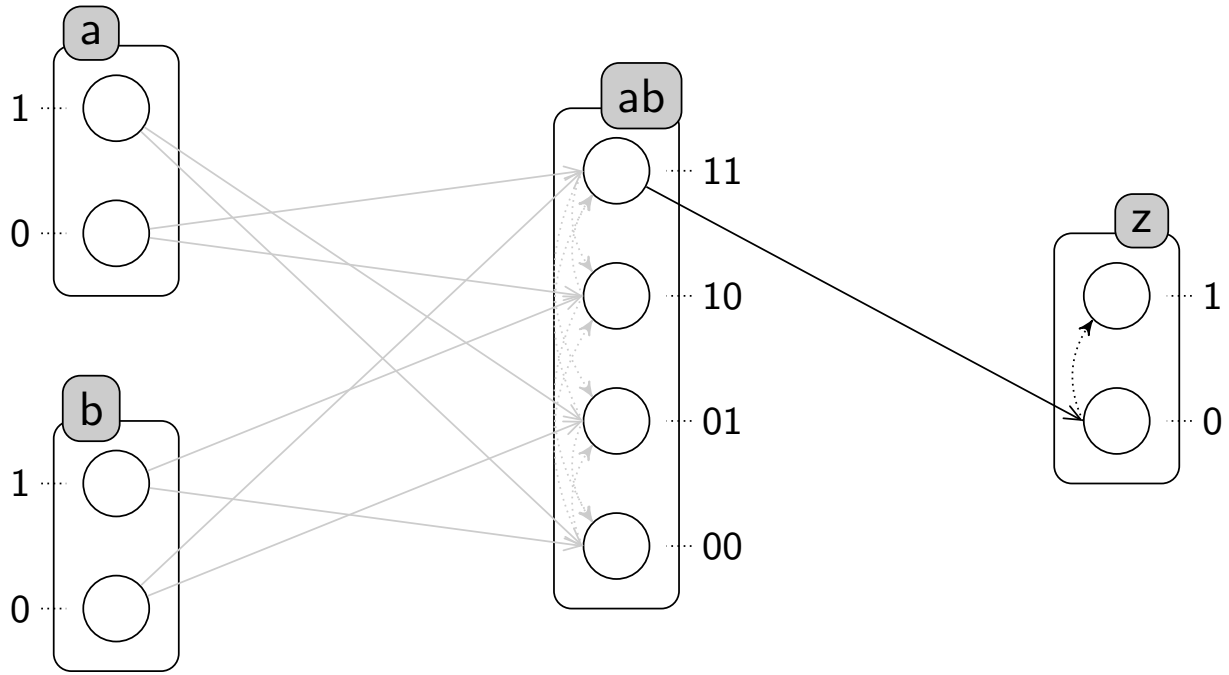
$\rightarrow$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \to z_0 \overset{\uparrow}{} z_1$?

$\to$ Add a **cooperative sort** reflecting the state of a and b



$\to$ Introduces a temporal shift (over-approximation)

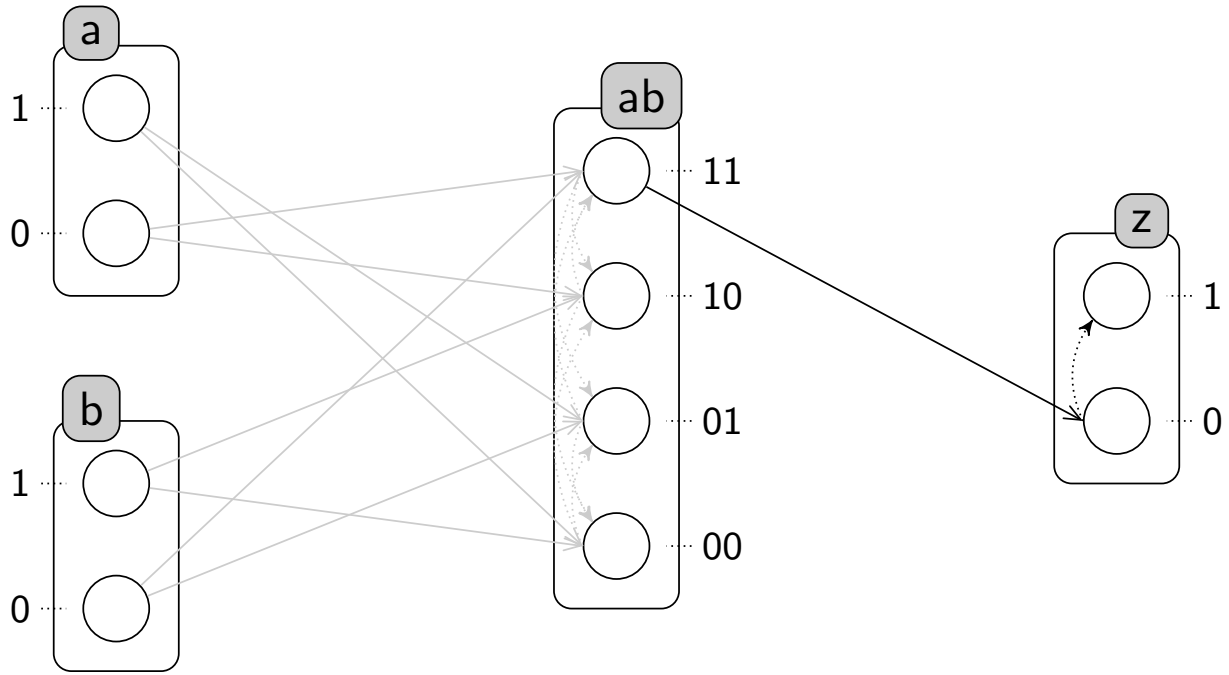
$\to$ Solution = Prioritized actions

Refining with Cooperation

Allow **cooperation** between two genes

- How to express $(a_1 \wedge b_1) \rightarrow z_0 \overset{\uparrow}{\uparrow} z_1$?

$\rightarrow$ Add a **cooperative sort** reflecting the state of a and b



$\rightarrow$ Introduces a temporal shift (over-approximation)

$\rightarrow$ Solution = Prioritized actions

Using Process Hitting for Interaction Graphs Study

Motivation

- Interaction Graph is the **historical discrete model** (suitable and widespread in biological research)
- Several tools exist of the analysis of interaction graphs, but the **state graph** is needed for some results $\Rightarrow$ **combinatorial explosion**

Contribution: Process Hitting to study large Biological Regulatory Networks

- Translation from Interaction Graphs + Refining
- **Efficient static analysis**

The Process Hitting modeling

Key features

- **Dynamic** modeling with an **atomistic** point of view
  - → Independent actions
  - → Cooperation modeled with cooperative sorts
- Efficient **static analysis**
  - → Reachability of a process can be computed in **linear time** in the number of sorts
- Useful for the study of **large biological models**
  - → Up to hundreds of sorts

(Future) extensions

- Actions with stochasticity
- Actions with priorities
- Continuous time with clocks?

The Pint Tool

[http://loicpauleve.name/pint/]

Features

- Free software (API available for future developments)
- **Textual language** to describe a Process Hitting (GUI currently under development)
- **Implemented tools**:
  - Translations from and to various other models
  - Fixed points research
  - Stochastic simulation
  - Reachability checker

The Mobyle portal

[http://mobyle.biotempo.univ-nantes.fr/cgi-bin/portal.py]

Presentation

- **Web application unifying** tools for systems biology analysis
- Powered by the Mobyle framework
- Project led by Julien GRAS (French ANR "BIOTempo")

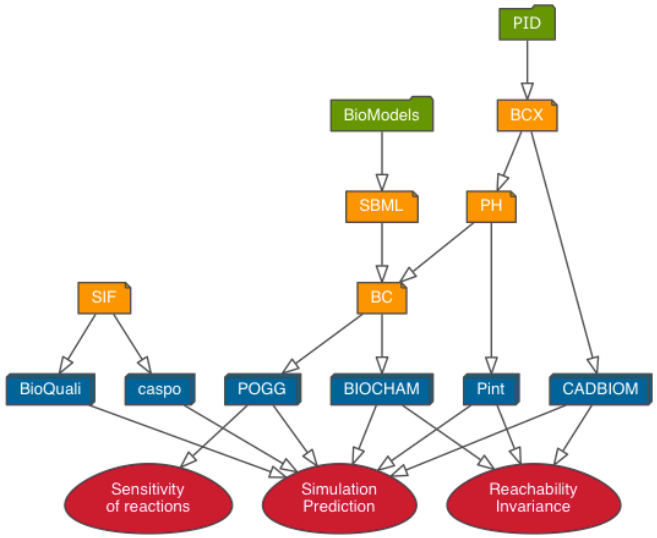


Figure: General architecture of the BIOtempo Mobyle server

The Mobyle portal

[http://mobyle.biotempo.univ-nantes.fr/cgi-bin/portal.py]

Presentation

- **Web application unifying** tools for systems biology analysis
- Powered by the Mobyle framework
- Project led by Julien GRAS (French ANR "BIOTempo")

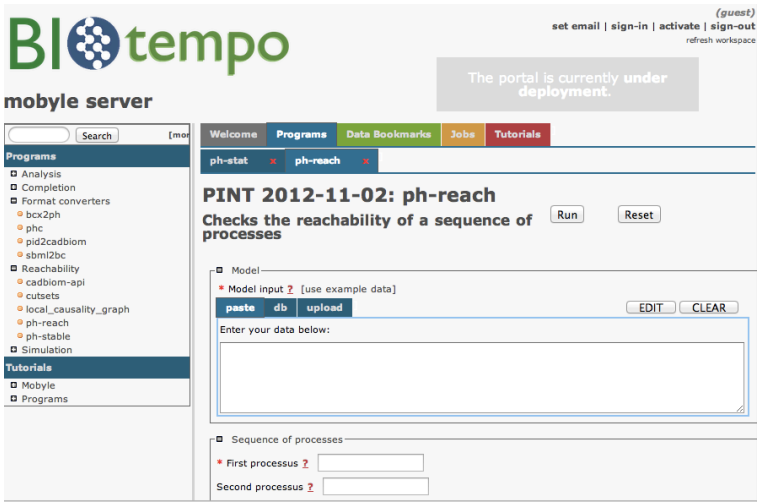
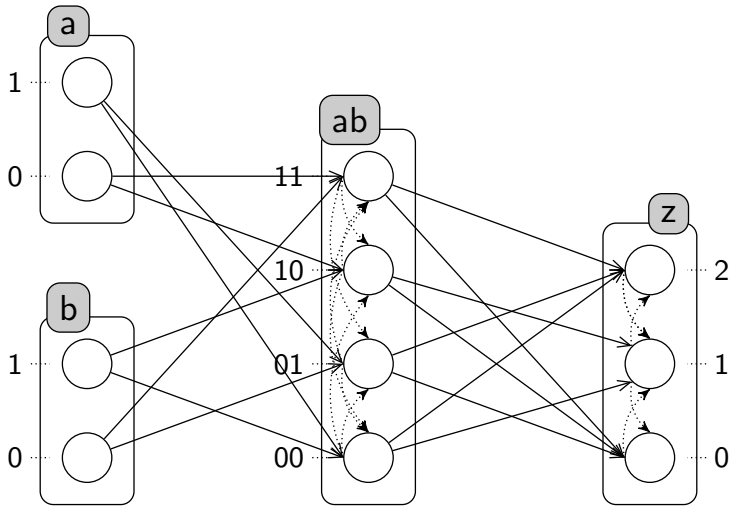


Figure: Screenshot from the BIOtempo Mobyle server: http://mobyle.biotempo.univ-nantes.fr/cgi-bin/portal.py

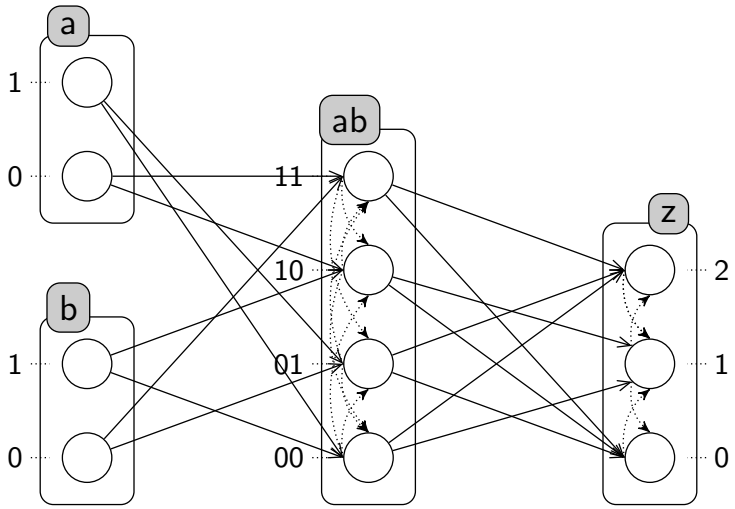
Overview

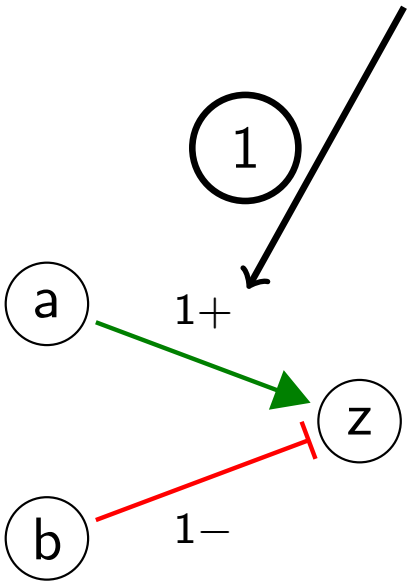
Inferring a BRN with Thomas' parameters



ω	$k_{z,\omega}$
$\varnothing$	1
$\{b\}$	0
$\{a\}$	2
$\{a; b\}$	1

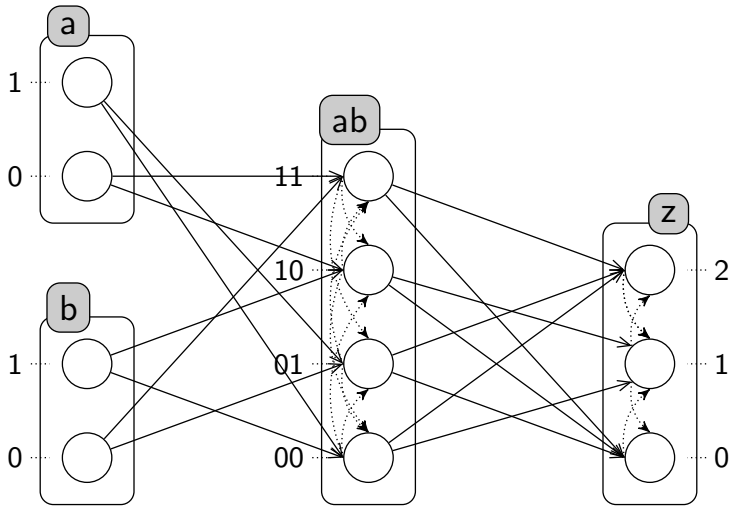
Inferring a BRN with Thomas' parameters

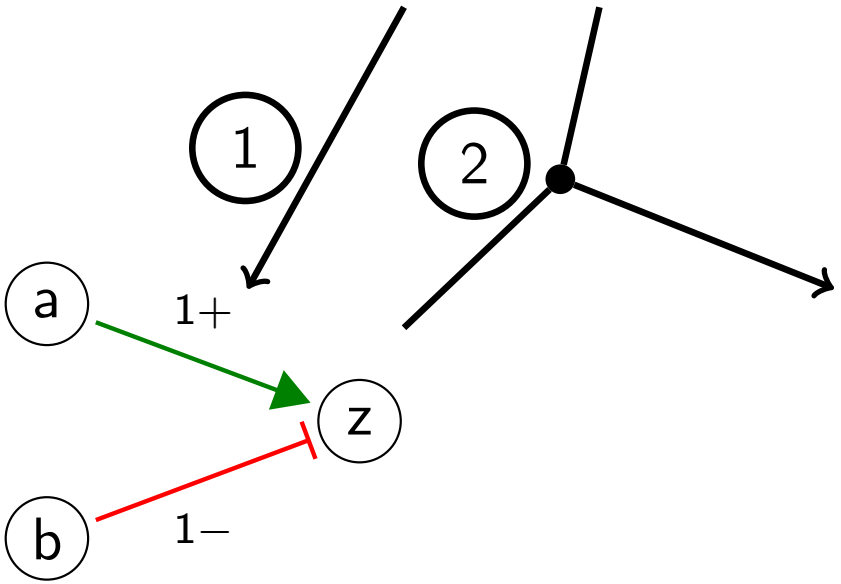




ω	$k_{z,\omega}$
$\varnothing$	1
$\{b\}$	0
$\{a\}$	2
$\{a; b\}$	1

Inferring a BRN with Thomas' parameters

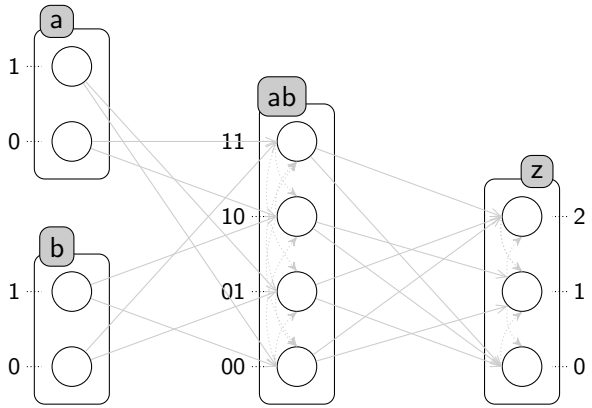


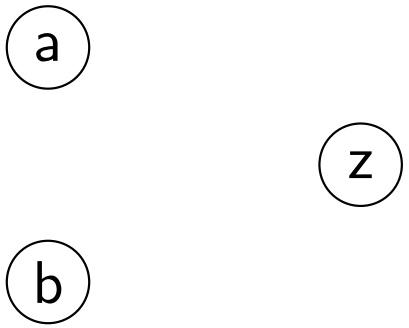


ω	$k_{z,\omega}$
$\varnothing$	1
$\{b\}$	0
$\{a\}$	2
$\{a; b\}$	1

Inferring the Interaction Graph

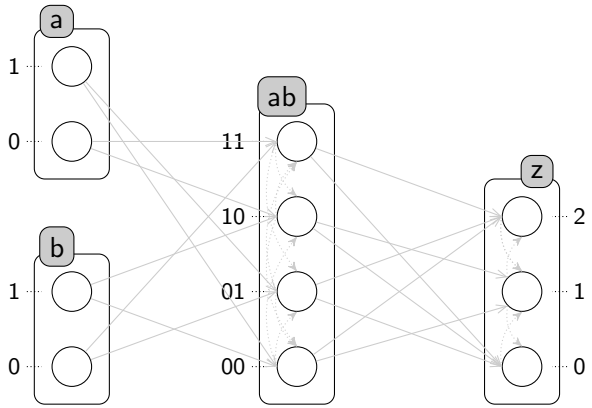
[FPI+12]

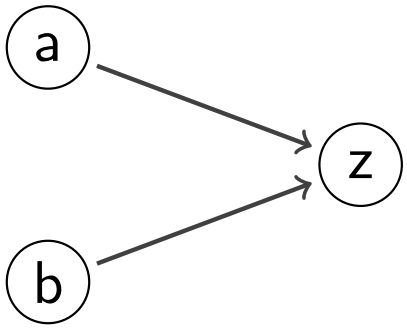




Inferring the Interaction Graph

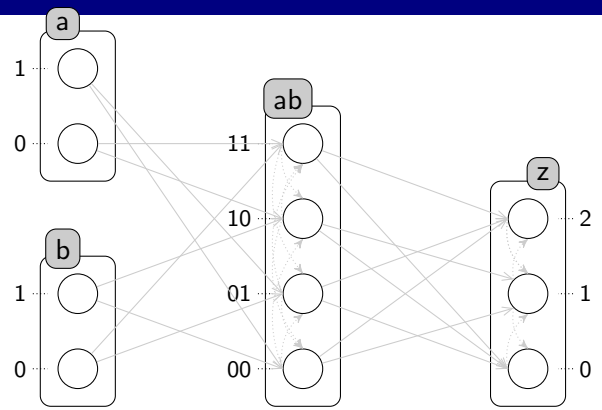
[FPI+12]

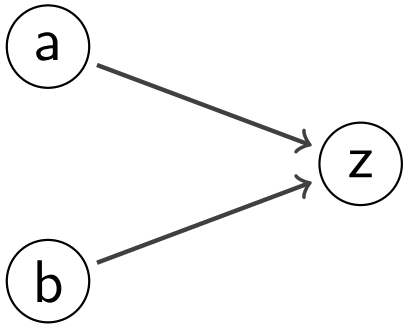




Inferring the Interaction Graph

[FPI+12]



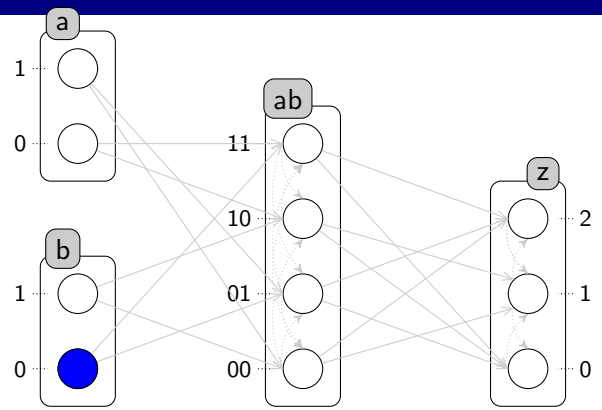


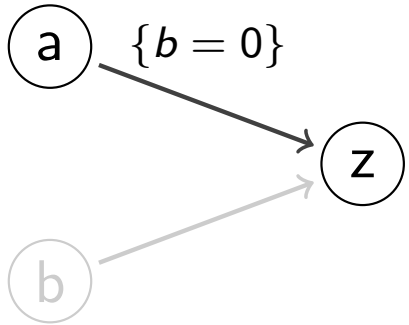
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rightarrow z_0 \nearrow z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



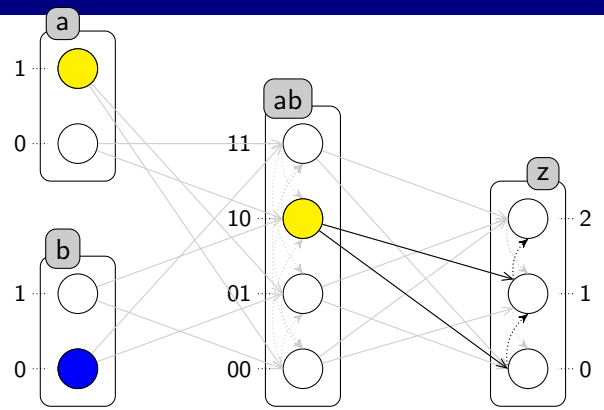


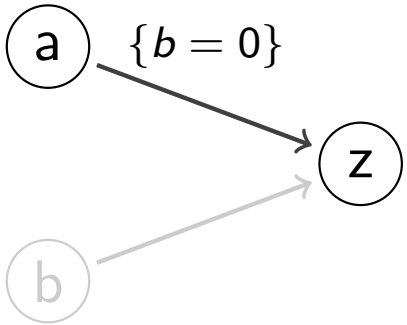
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \nearrow a_1 \Rightarrow z_0 \nearrow z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



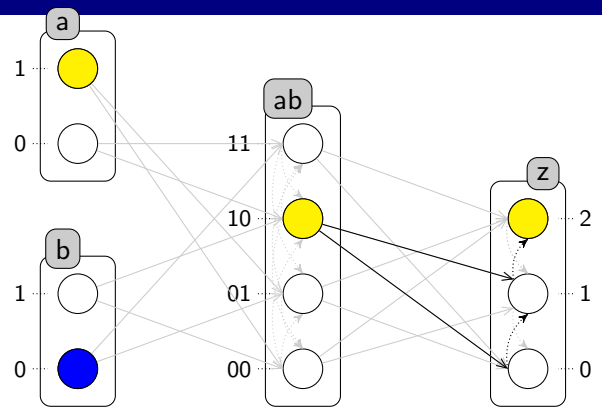


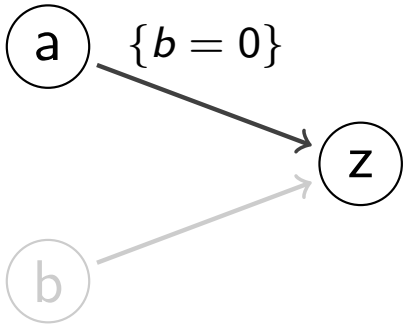
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



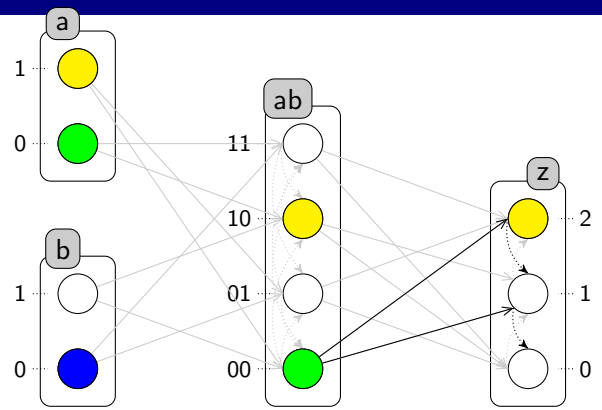


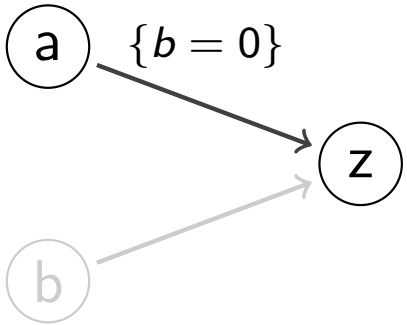
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \nearrow a_1 \Rightarrow z_0 \nearrow z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



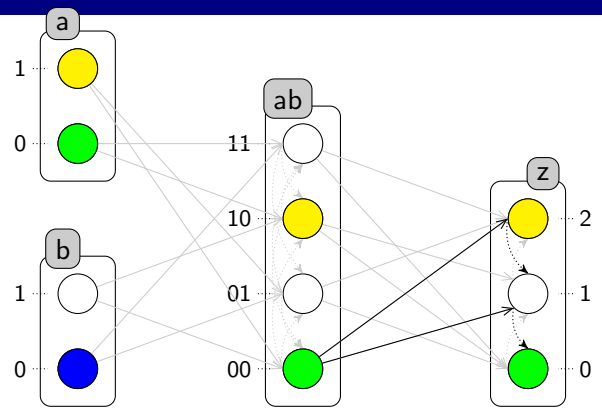


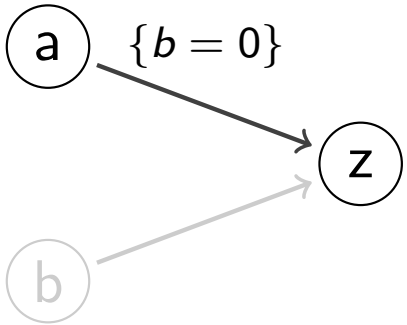
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



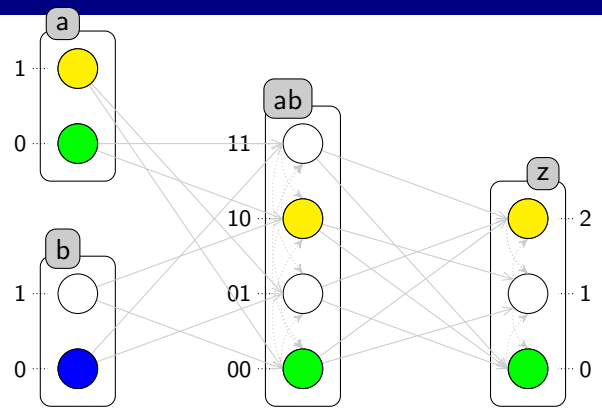


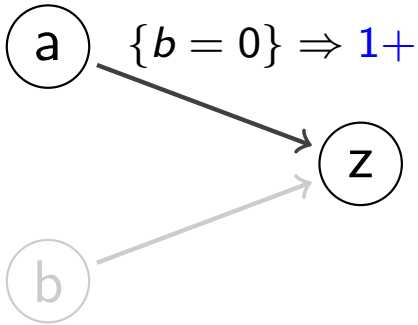
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



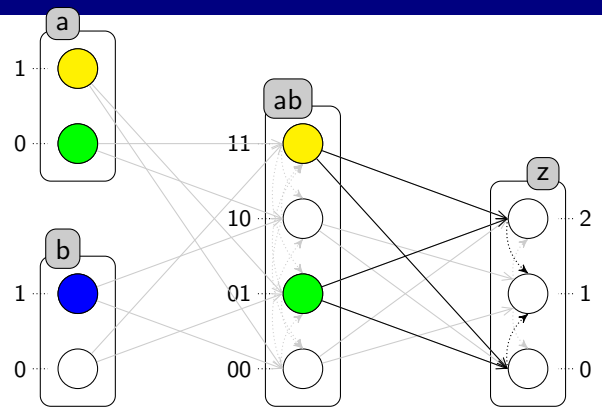


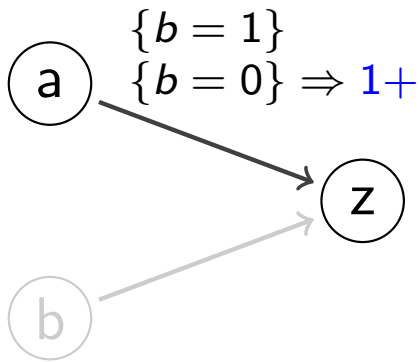
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



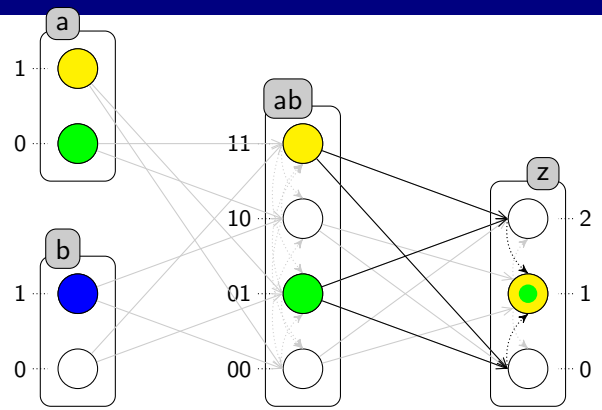


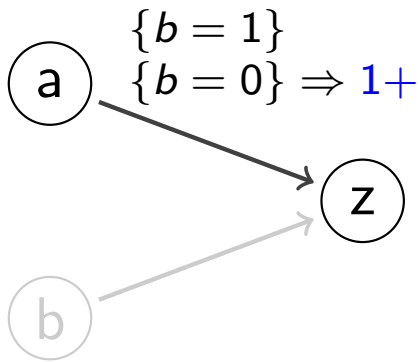
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: ($a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2$)
   $\Rightarrow$ activation ($+$) & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



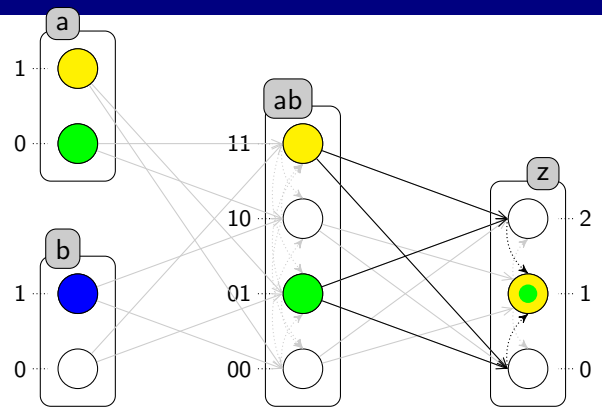


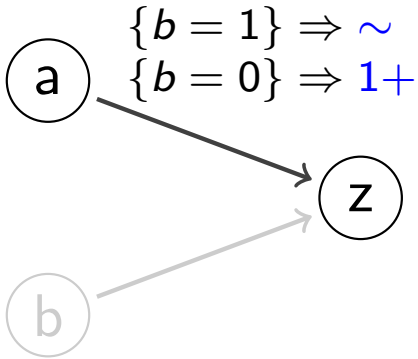
→ **Exhaustive search in all possible configurations**

1. Pick one regulator $[a]$, and choose an active process for all the others $[b_0]$.
2. Change the active process of this regulator $[a_0, a_1]$ and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



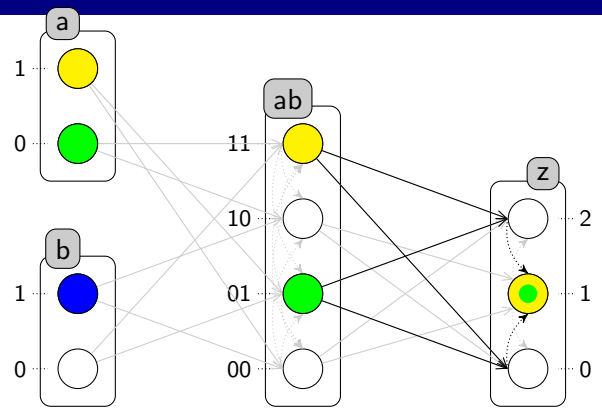


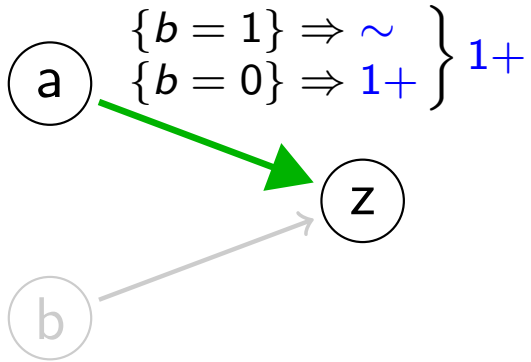
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \nearrow a_1 \Rightarrow z_0 \nearrow z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]



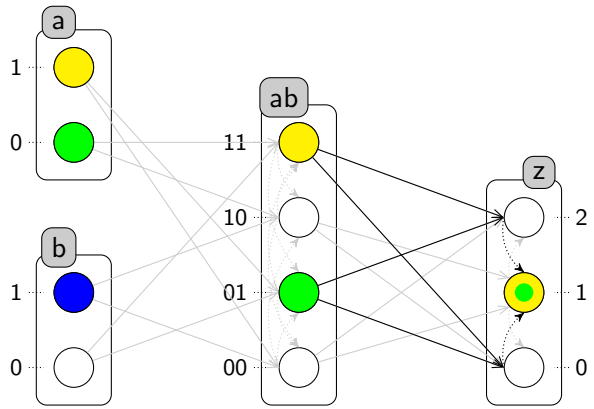


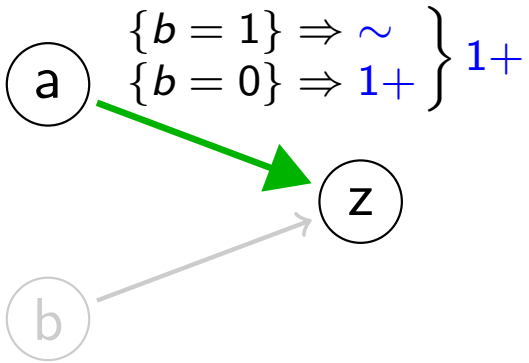
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \nearrow a_1 \Rightarrow z_0 \nearrow z_2)$
   $\Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and conclude globally.

Inferring the Interaction Graph

[FPI+12]





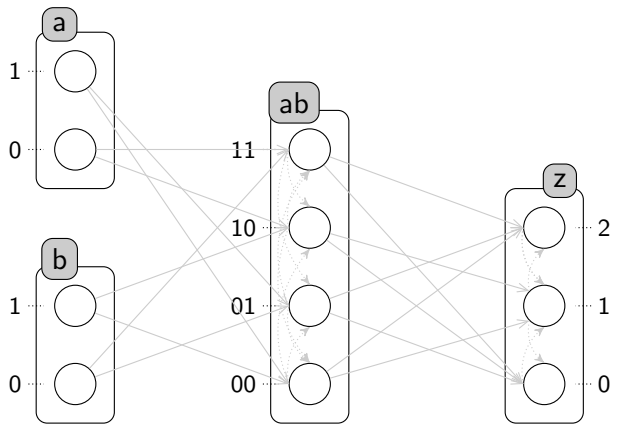
Problematic cases:

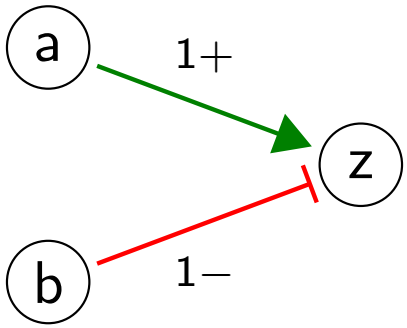
→ No focal processes (**cycle**)
→ Opposite influences ($+$ & $-$)

$\Rightarrow$ **Unsigned edge**

Inferring Parameters

[FPI+12]





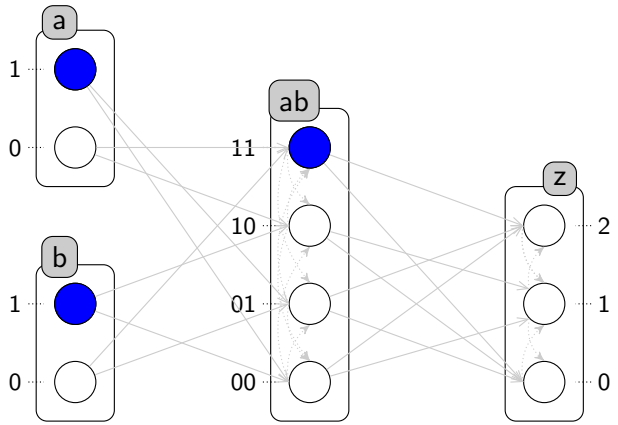
ω		$k_{z,\omega}$
a	b	
$-$	$+$	
$-$	$-$	
$+$	$+$	
$+$	$-$	1

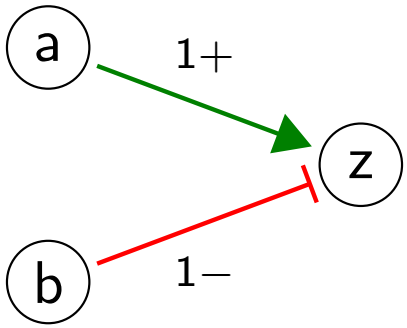
1. For each configuration of resources $[\omega = \{a^+, b^-\}]$ find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

   Inconclusive cases:
   - Behavior cannot be represented as a BRN
   - Lack of cooperation (no focal processes)

Inferring Parameters

[FPI+12]



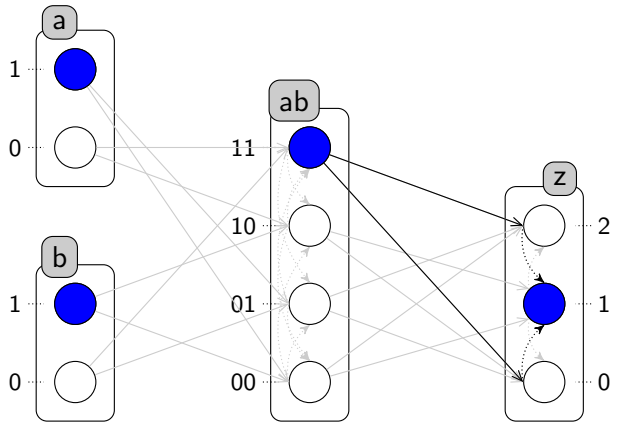


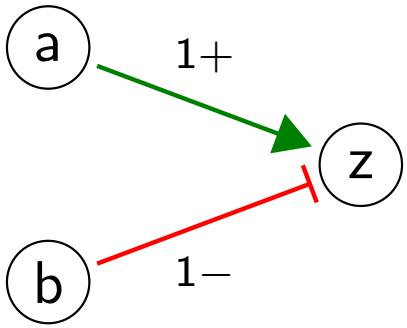
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

1. For each configuration of resources $[\omega = \{a^+, b^-\}]$
   find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$
   Inconclusive cases:
   - Behavior cannot be represented as a BRN
   - Lack of cooperation (no focal processes)

Inferring Parameters

[FPI+12]





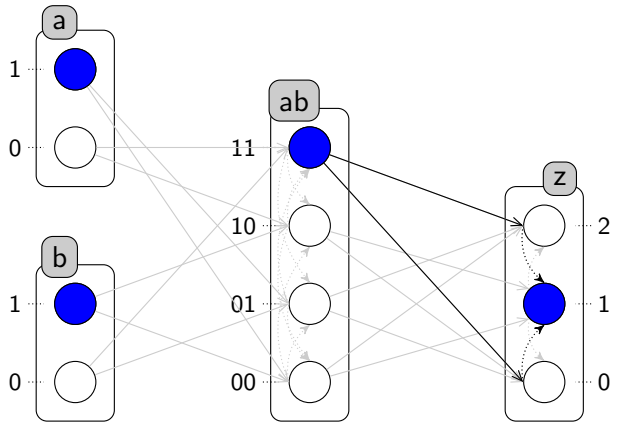
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

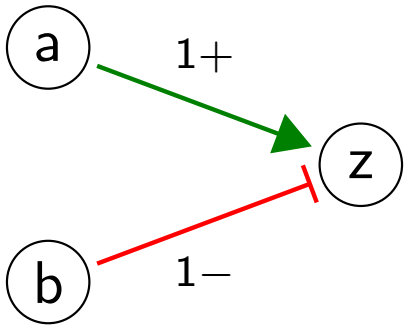
1. For each configuration of resources $[\omega = \{a^+, b^-\}]$ find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

   Inconclusive cases:
   - Behavior cannot be represented as a BRN
   - Lack of cooperation (no focal processes)

Inferring Parameters

[FPI+12]





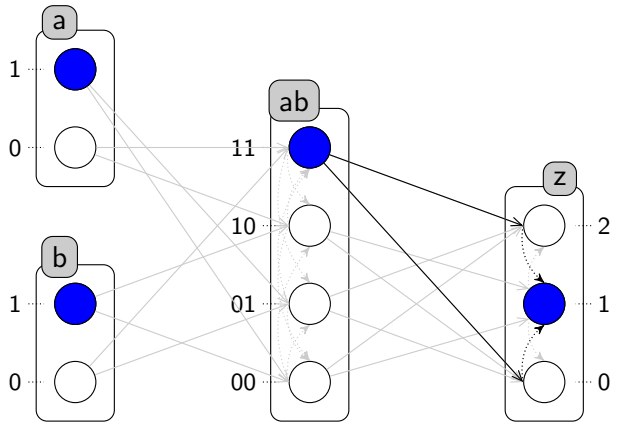
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

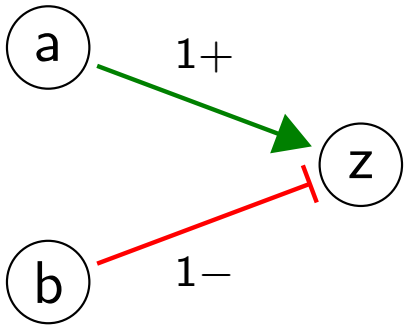
1. For each configuration of resources $[\omega = \{a^+, b^-\}]$
   find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

   Inconclusive cases:
   - Behavior cannot be represented as a BRN
   - Lack of cooperation (no focal processes)

Inferring Parameters

[FPI+12]



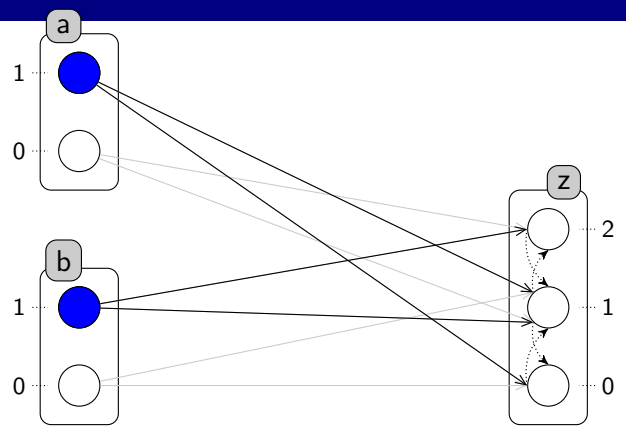


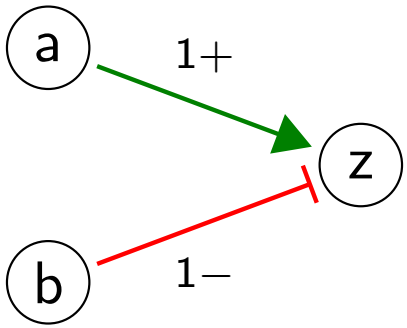
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

1. For each configuration of resources $[\omega = \{a^+, b^-\}]$ find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

   Inconclusive cases:
   - Behavior cannot be represented as a BRN
   - Lack of cooperation (no focal processes)

Inferring Parameters





ω		$k_{z,\omega}$
a	b	
$-$	$+$	$?$
$-$	$-$	0
$+$	$+$	2
$+$	$-$	$?$

1. For each configuration of resources $[\omega = \{a^+, b^-\}]$ find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

   Inconclusive cases:
   - Behavior cannot be represented as a BRN
   - Lack of cooperation (no focal processes)

2. If some parameters could not be inferred, enumerate all admissible parametrizations, regarding **biological constraints** and **the dynamics of the Process Hitting** $\Rightarrow k_{z,\{a^+,b^-\}} \in \{0;1;2\}$; $k_{z,\{a^-,b^+\}} \in \{0;1;2\}$

Overview

Implementation

Workflow:

- Read and translate the models with **OCaml**
  - → Uses the existing free library **Pint**
  - → Documentation + examples: `http://processhitting.wordpress.com/`
- Express the problem in **ASP** (logic programming)
  - → Solve with **Clingo** (**Gringo** + **Clasp**)

Model specifications				IG inference		Parameters inference	
Name	S+CS	P	A	Δt	Edges	Δt	Parameters
[EGFR20]	**20**+22	152	399	**1s**	50	**1s**	191
[TCRSIG40]	**40**+14	156	301	**1s**	54	**1s**	143
[TCRSIG94]	**94**+39	448	1124	**13s**	169	∞	2.10^9
[EGFR104]	**104**+89	748	2356	**4min**	241	**1min 30s**	$1.10^6/2.10^6$

S = Sorts CS = Cooperative sorts P = Processes A = Actions

[EGFR20]: Epidermal Growth Factor Receptor, by Özgür Sahin et al.
[EGFR104]: Epidermal Growth Factor Receptor, by Regina Samaga et al.
[TCRSIG40]: T-Cell Receptor Signaling, by Steffen Klamt et al.
[TCRSIG94]: T-Cell Receptor Signaling, by Julio Saez-Rodriguez et al.

Implementation

Workflow:

- Read and translate the models with **OCaml**
  - → Uses the existing free library **Pint**
  - → Documentation + examples: `http://processhitting.wordpress.com/`
- Express the problem in **ASP** (logic programming)
  - → Solve with **Clingo** (**Gringo** + **Clasp**)

Model specifications				IG inference		Parameters inference	
Name	S+CS	P	A	Δt	Edges	Δt	Parameters
[EGFR20]	**20**+22	152	399	**1s**	50	**1s**	191
[TCRSIG40]	**40**+14	156	301	**1s**	54	**1s**	143
[TCRSIG94]	**94**+39	448	1124	**13s**	169	∞	2.10^9
[EGFR104]	**104**+89	748	2356	**4min**	241	**1min 30s**	$1.10^6/2.10^6$

S = Sorts CS = Cooperative sorts P = Processes A = Actions

[EGFR20]: Epidermal Growth Factor Receptor, by Özgür Sahin et al.
[EGFR104]: Epidermal Growth Factor Receptor, by Regina Samaga et al.
[TCRSIG40]: T-Cell Receptor Signaling, by Steffen Klamt et al.
[TCRSIG94]: T-Cell Receptor Signaling, by Julio Saez-Rodriguez et al.

Summary

Process Hitting and ASP

- Inference of the **complete Interaction Graph**
- Inference of the **possibly partial Parametrization**
- Enumerate all full & **admissible Parametrizations**

$\rightarrow$ Exhaustive approaches

Complexity: linear in the number of genes, exponential in the number of regulators of one gene

Summary

Contribution: new translation Process Hitting $\rightsquigarrow$ René Thomas

- $\rightarrow$ New **formal link** between the two models
- $\rightarrow$ More **visibility** to the Process Hitting
- $\rightarrow$ Inference approach that takes benefit from both the Process Hitting compact structure and the power of ASP

Further work

Models and algorithms

- Add **priorities** in the Process Hitting framework and adapt the static analyses approaches for this enriched model [FPMR13]
- From priorities to **quantitative timing information**
- Connect Process Hitting compact structure with **decomposition techniques** in continuous approaches [ACC12, CAC+13]

Application

- Use the approach for the analysis of **larger** biological networks
- Contribute to the **discovery** of biological regulatory networks based on biological data
- Study key properties (e.g. concept of **resilience**)


Amine Ammar, Elias Cueto, and Francisco Chinesta.
Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions.
International Journal for Numerical Methods in Biomedical Engineering, 28(9):960–973, 2012.


Courtney Chancellor, Amine Ammar, Francisco Chinesta, Morgan Magnin, and Olivier Roux.
Linking discrete and stochastic models: The chemical master equation as a bridge between process hitting and proper generalized decomposition.
In CMSB, pages 50–63, 2013.


Maxime Folschette, Loïc Paulevé, Katsumi Inoue, Morgan Magnin, and Olivier Roux.
Concretizing the process hitting into biological regulatory networks.
In Proceedings of the 10th international conference on Computational Methods in Systems Biology, CMSB'12, pages 166–186, Berlin, Heidelberg, 2012. Springer-Verlag.


Maxime Folschette, Loïc Paulevé, Morgan Magnin, and Olivier Roux.
Under-approximation of reachability in multivalued asynchronous networks.
In 4th International Workshop on Interactions between Computer Science and Biology (CS2Bio'13), 2013.
accepted.


Loïc Paulevé, Morgan Magnin, and Olivier Roux.
Refining dynamics of gene regulatory networks in a stochastic π-calculus framework.
In Transactions on Computational Systems Biology XIII, volume 6575 of Lecture Notes in Comp Sci, pages 171–191. Springer, 2011.


Loïc Paulevé, Morgan Magnin, and Olivier Roux.
Static analysis of biological regulatory networks dynamics using abstract interpretation.
Mathematical Structures in Computer Science, 22(04):651–685, 2012.


R. Thomas, A.M. Gathoye, and L. Lambert.

A complex control circuit. regulation of immunity in temperate bacteriophages.
Eur J Biochem, 71(1):211–27, 1976.