Public seminar of team Lifeware

Modeling and analysis of large regulatory networks with the Process Hitting framework

Maxime FOLSCHETTE

FMV team / Department of Electronics & Informatics / University of Kassel
maxime.folschette@uni-kassel.de
http://maxime.folschette.name/

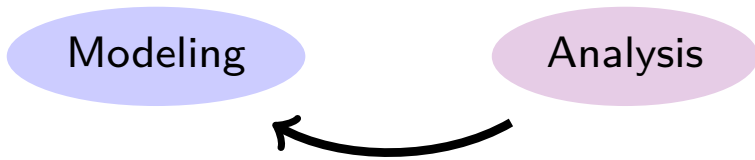
2015/04/01

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

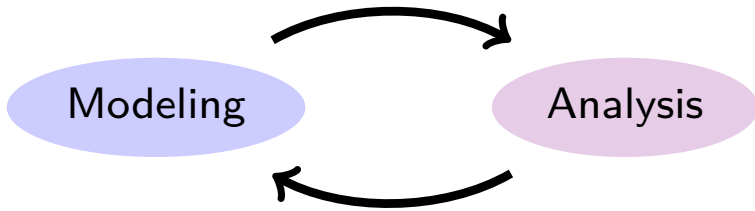


The required analysis has an impact on modeling

- The modeling tools must be adapted to the observed properties

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension



The required analysis has an impact on modeling

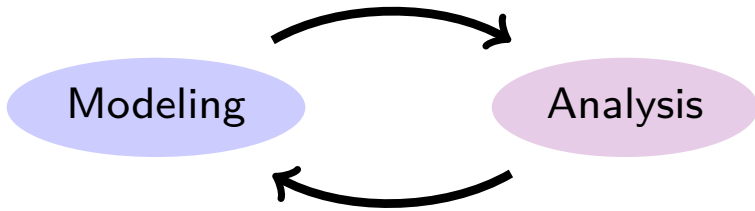
- The modeling tools must be adapted to the observed properties

Modeling choices have an impact on the results of the analysis

- The level of details changes the quantity of obtained info
- The size of the model increases the analysis duration

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension



The required analysis has an impact on modeling

- The modeling tools must be adapted to the observed properties

Modeling choices have an impact on the results of the analysis

- The level of details changes the quantity of obtained info
- The size of the model increases the analysis duration

The modeling and analysis steps of a system are strongly linked

Overview of This Presentation

State of the Art of the modeling of biological regulatory networks

- Discrete asynchronous representations and Thomas modeling
- Standard Process Hitting

Overview of This Presentation

State of the Art of the modeling of biological regulatory networks

- Discrete asynchronous representations and Thomas modeling
- Standard Process Hitting

Enriching the Process Hitting

- Integration of temporal constraints
- Synchronicity between actions
  - $\rightarrow$ Adding of priorities, neutralizing edges or synchronous actions

Overview of This Presentation

State of the Art of the modeling of biological regulatory networks

- Discrete asynchronous representations and Thomas modeling
- Standard Process Hitting

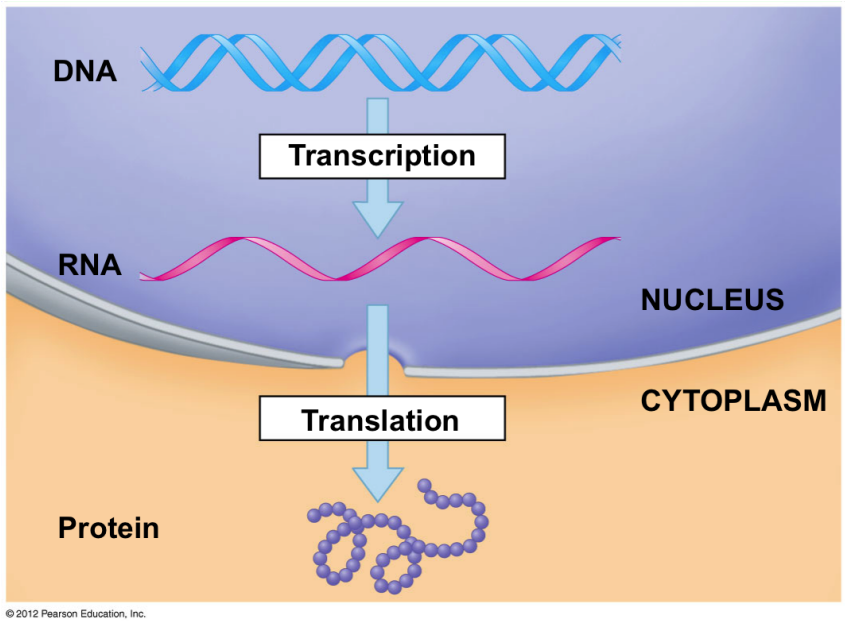
Enriching the Process Hitting

- Integration of temporal constraints
- Synchronicity between actions
  - $\rightarrow$ Adding of priorities, neutralizing edges or synchronous actions

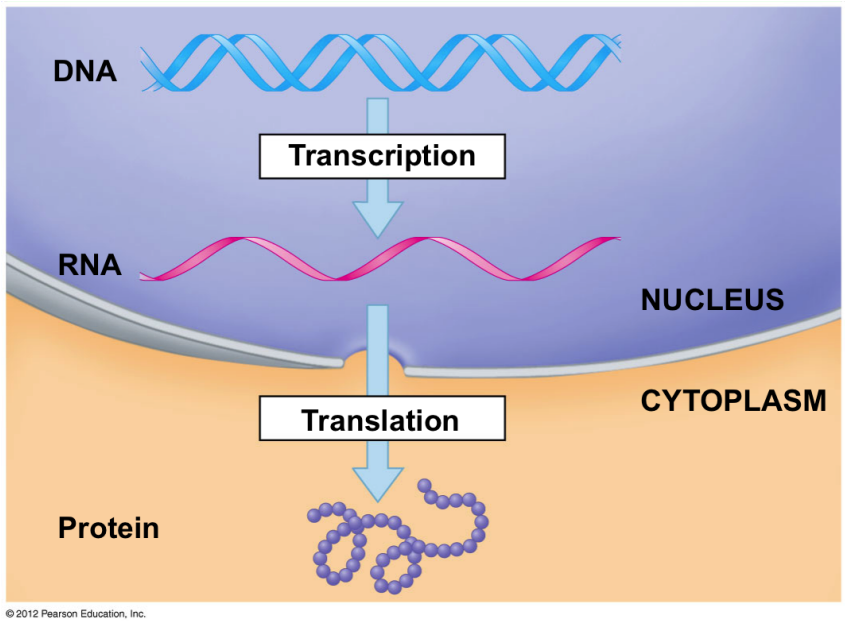
Analysis of the Process Hitting

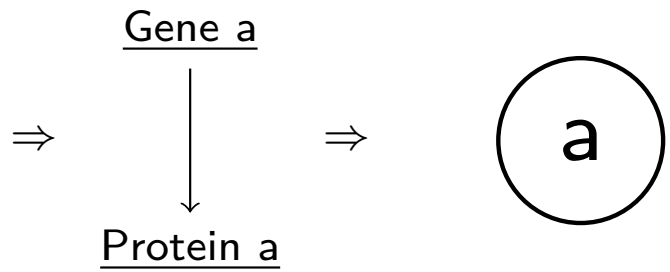
- Correction of the cooperative sorts
- Static analysis of reachability
- Equivalences and links with other formalisms

Abstractions of the Representation



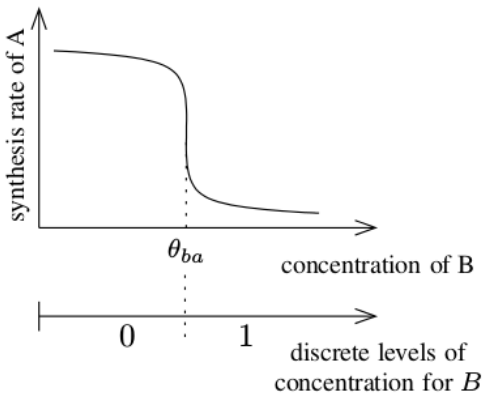
Abstractions of the Representation





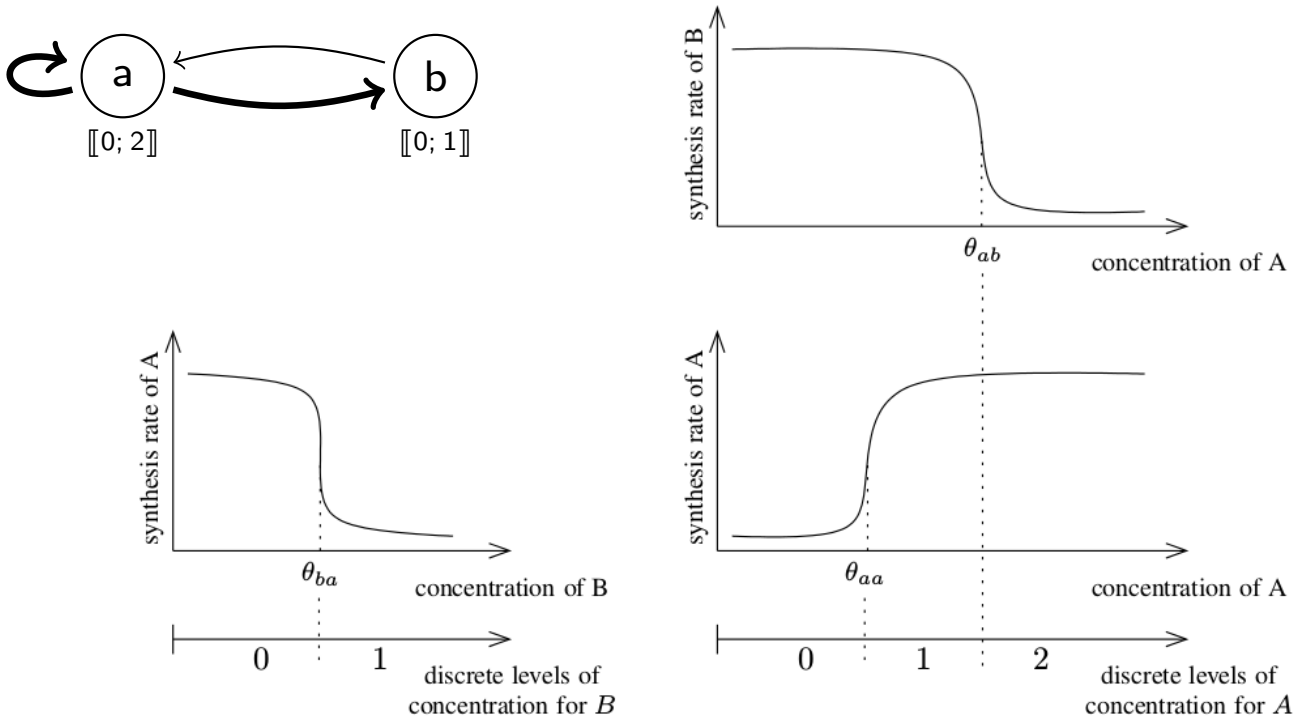
Discretization and Asynchronism

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



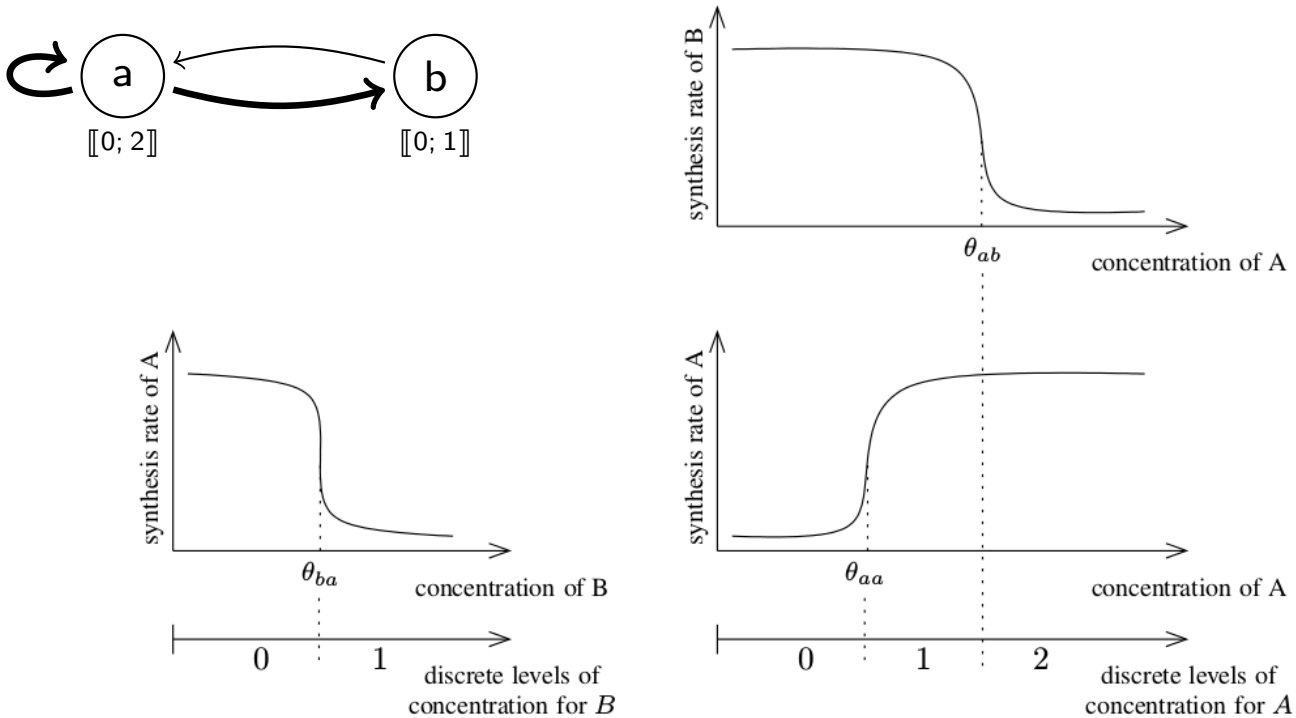
Discretization and Asynchronism

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



Discretization and Asynchronism

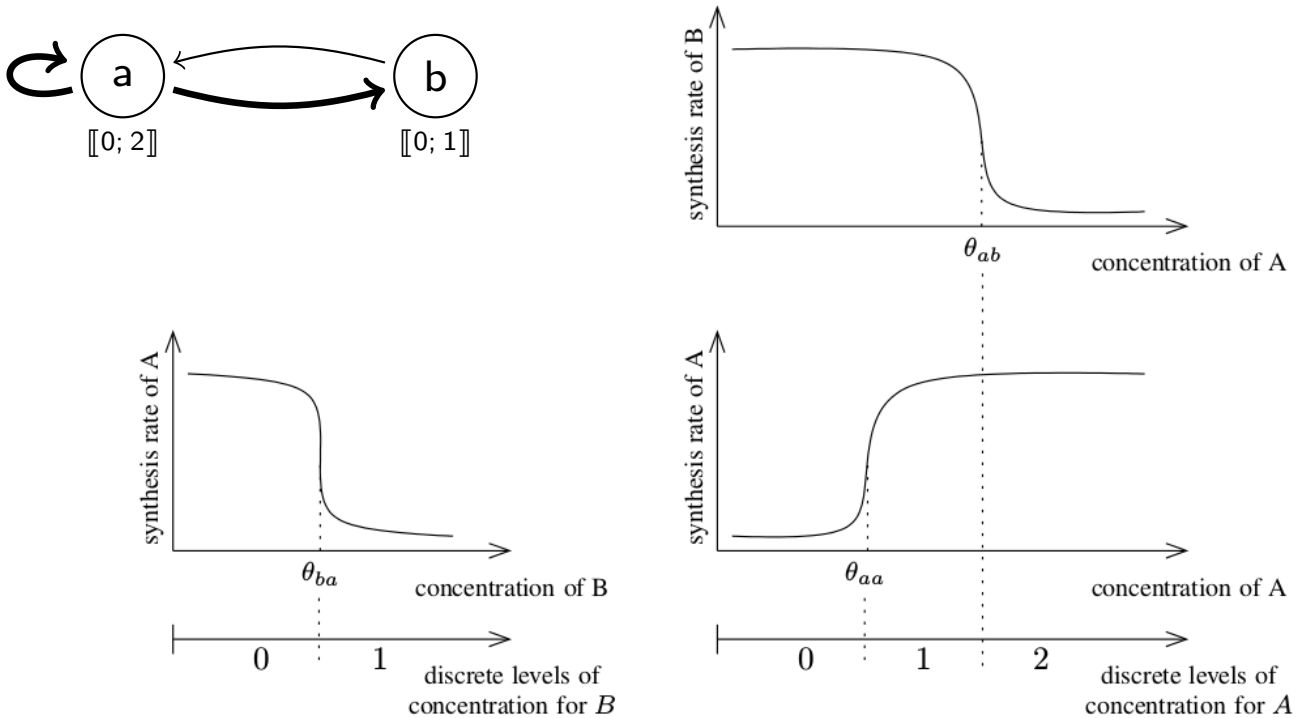
[Richard, Comet, Bernot (tutorial), 2008]



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

Discretization and Asynchronism

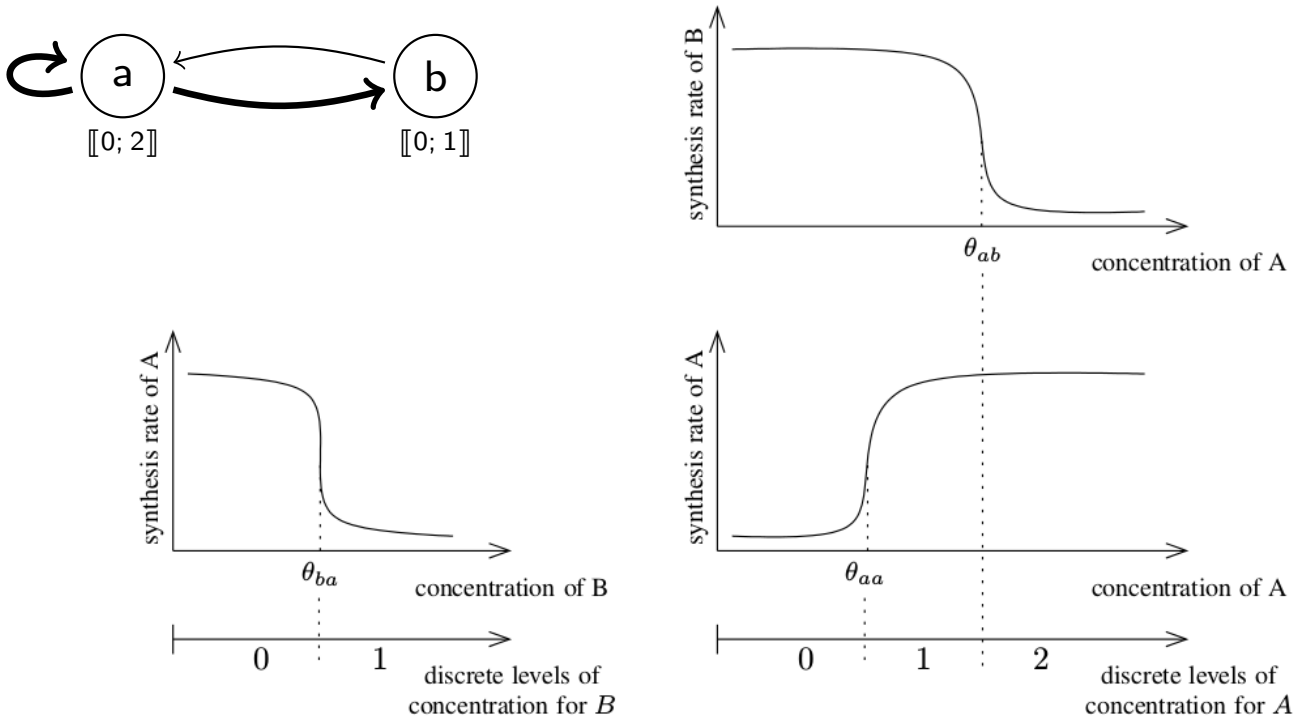
[Richard, Comet, Bernot (tutorial), 2008]



- 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]

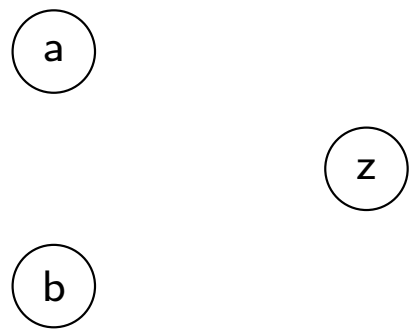


- Unknown real values of concentrations or continuous activity levels
  → Abstracted as thresholds or **discrete levels**
- Continuous variations of the real values
  → **Unitary** dynamics
- Simultaneous crossings of two thresholds never occurs
  → **Asynchronous** dynamics

Discrete Networks / Thomas Modeling

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

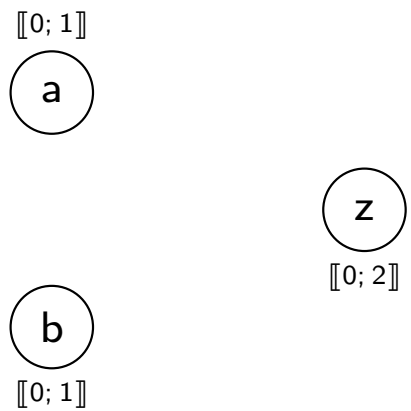
- A set of components $N = \{a, b, z\}$



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

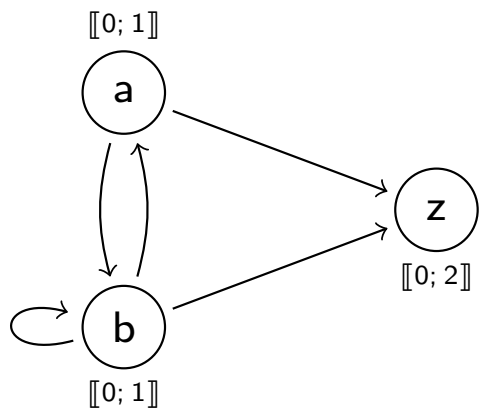
[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- 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

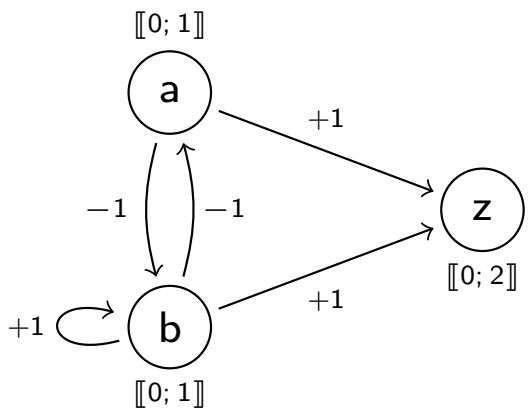
[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- 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**

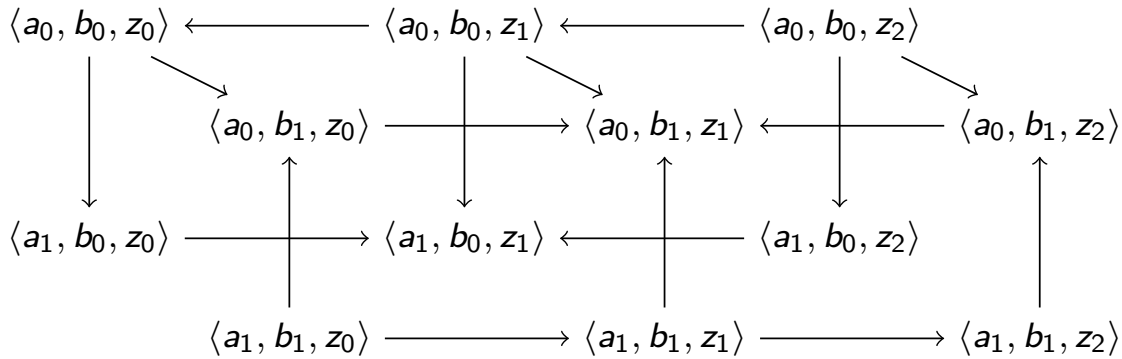
a	b	$f^b(a,b)$
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

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



Analysis of Thomas Modeling

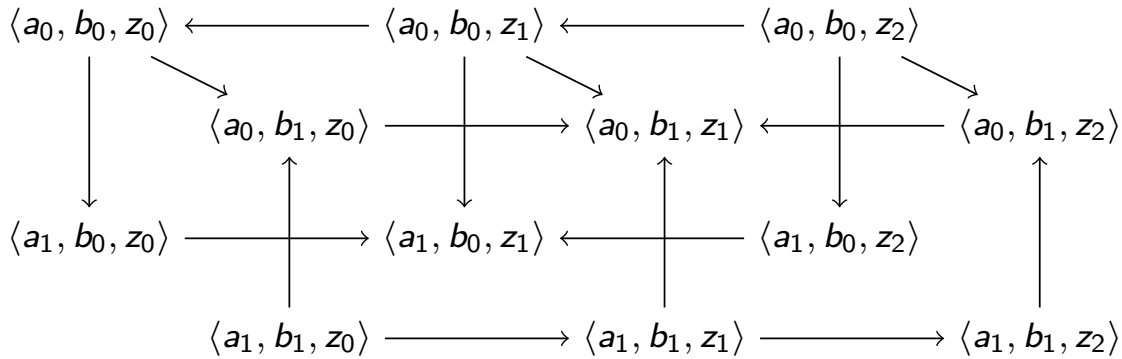
The State graph is computed in a unitary and asynchronous fashion



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

Analysis of Thomas Modeling

The State graph is computed in a unitary and asynchronous fashion



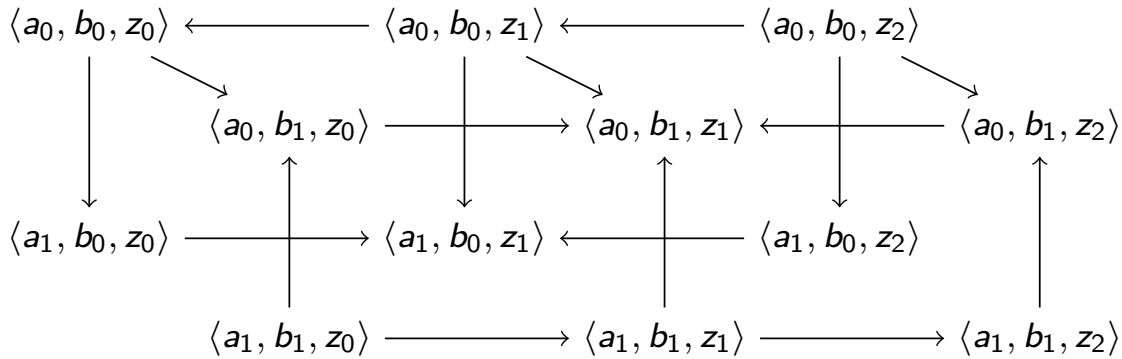
→ **Exponential** size in the number of components

Some works all to link the structure of the model and 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



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

Some works all to link the structure of the model and 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]

Standard Process Hitting

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

Standard Process Hitting is:

- Well-adapted to the modeling of BRNs
- An **atomistic and qualitative** modeling (explicit & discrete expression levels)
- **Simple but powerful** dynamics (constraints on the form of actions)

Standard Process Hitting

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

Standard Process Hitting is:

- Well-adapted to the modeling of BRNs
- An **atomistic and qualitative** 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

→ Well-adapted formalism to study **large BRNs**

Standard Process Hitting

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

Standard Process Hitting is:

- Well-adapted to the modeling of BRNs
- An **atomistic and qualitative** 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

→ Well-adapted formalism to study **large BRNs**

Several missing features:

- Faulty representation **cooperations**
- **Possible enrichment** of the expressivity
  → Which requires to adapt the previous tools

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]

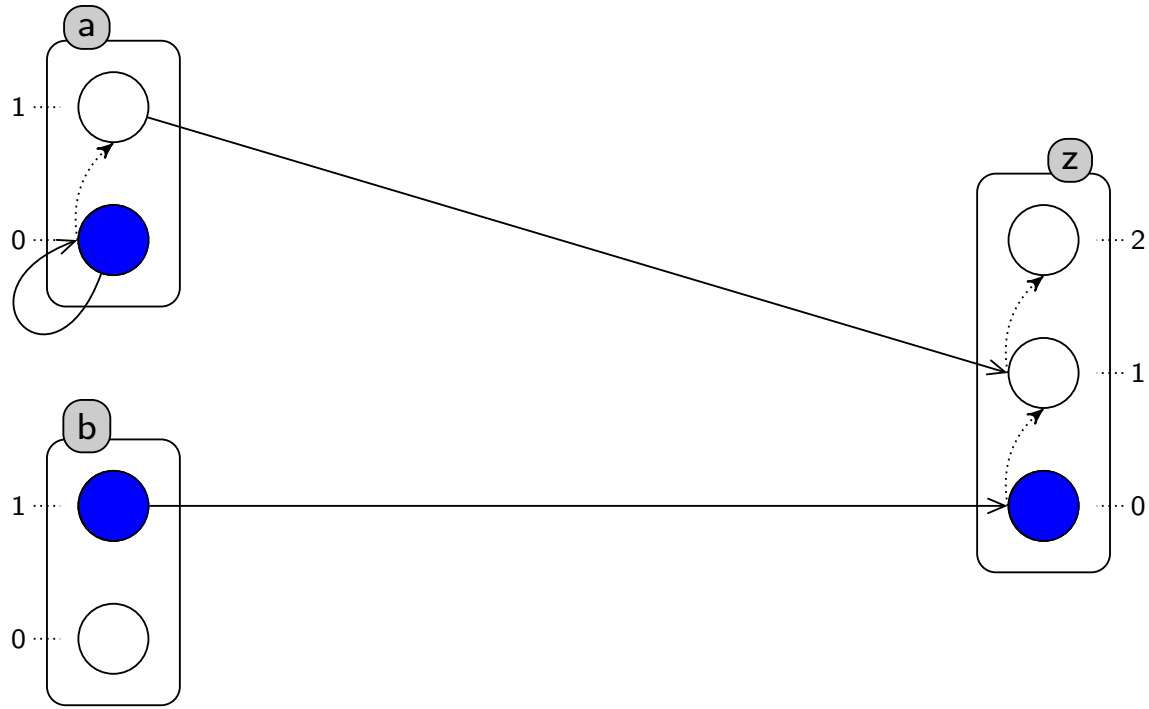
Sorts: components a, b, z

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 $a,\ b,\ z$

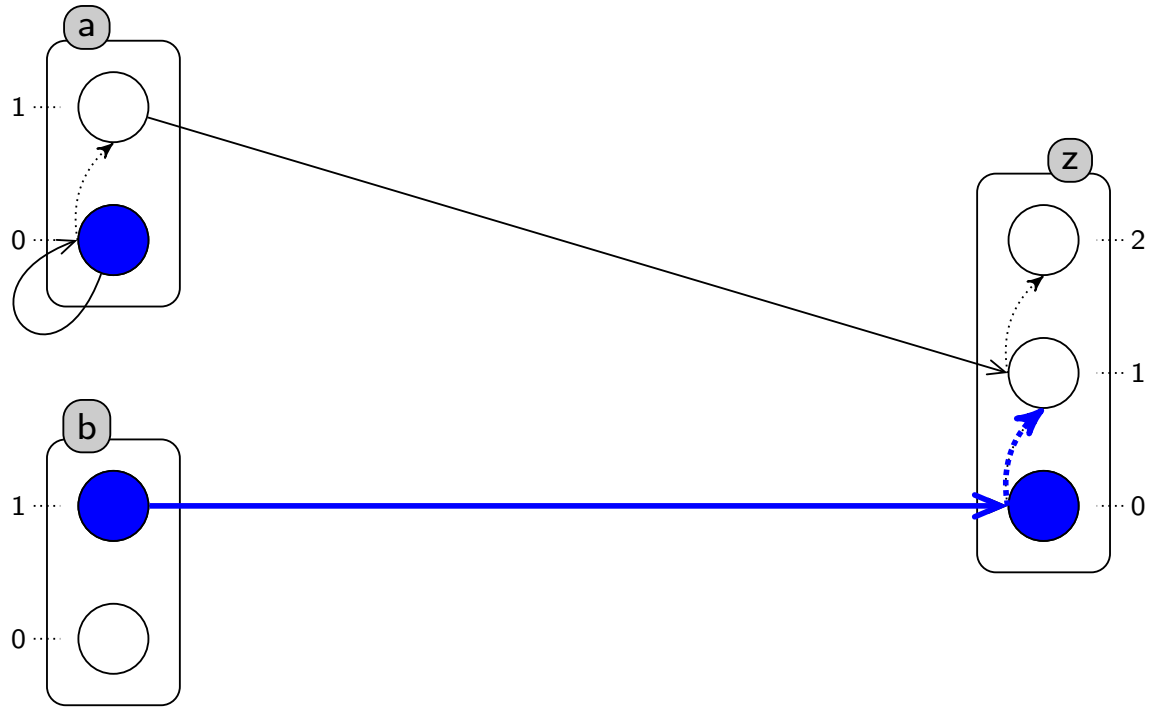
Processes: local states / discrete expression levels $z_0,\ z_1,\ z_2$

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

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

Standard Process Hitting

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



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

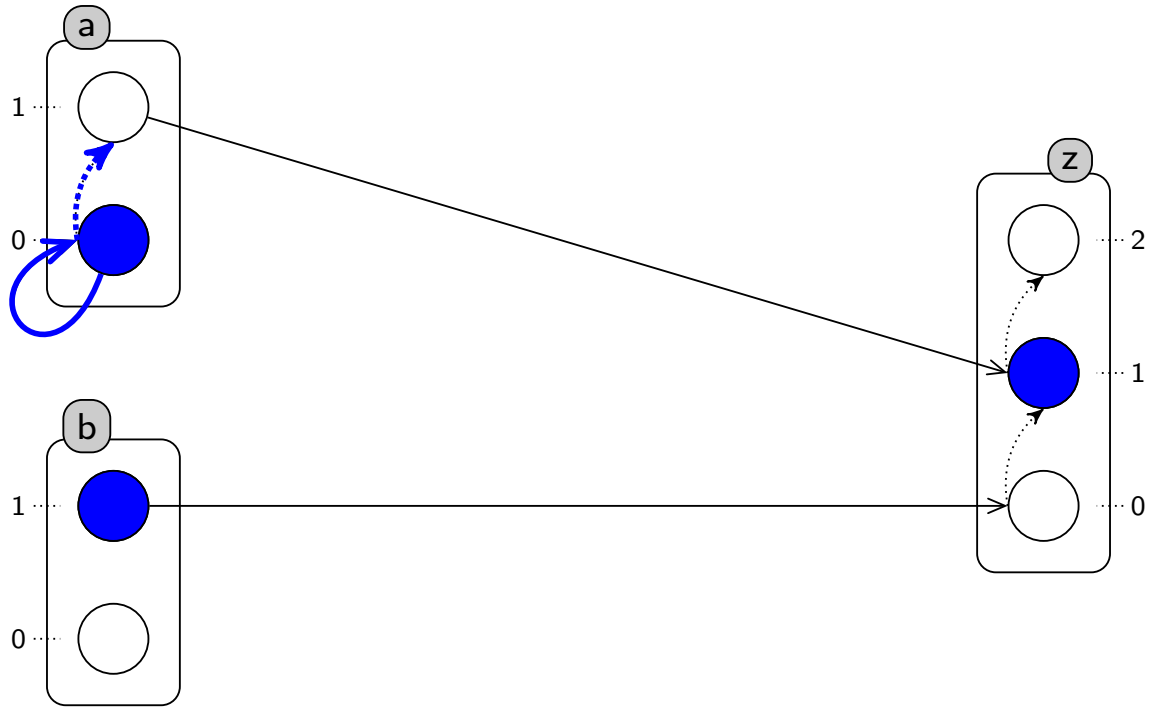
Processes: local states / discrete expression levels $z_0,\ z_1,\ z_2$

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

Actions: dynamics $b_1 \longrightarrow z_0 \uparrow z_1,\ a_0 \longrightarrow a_0 \uparrow a_1,\ a_1 \longrightarrow z_1 \uparrow z_2$

Standard Process Hitting

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



Sorts: components a, b, z

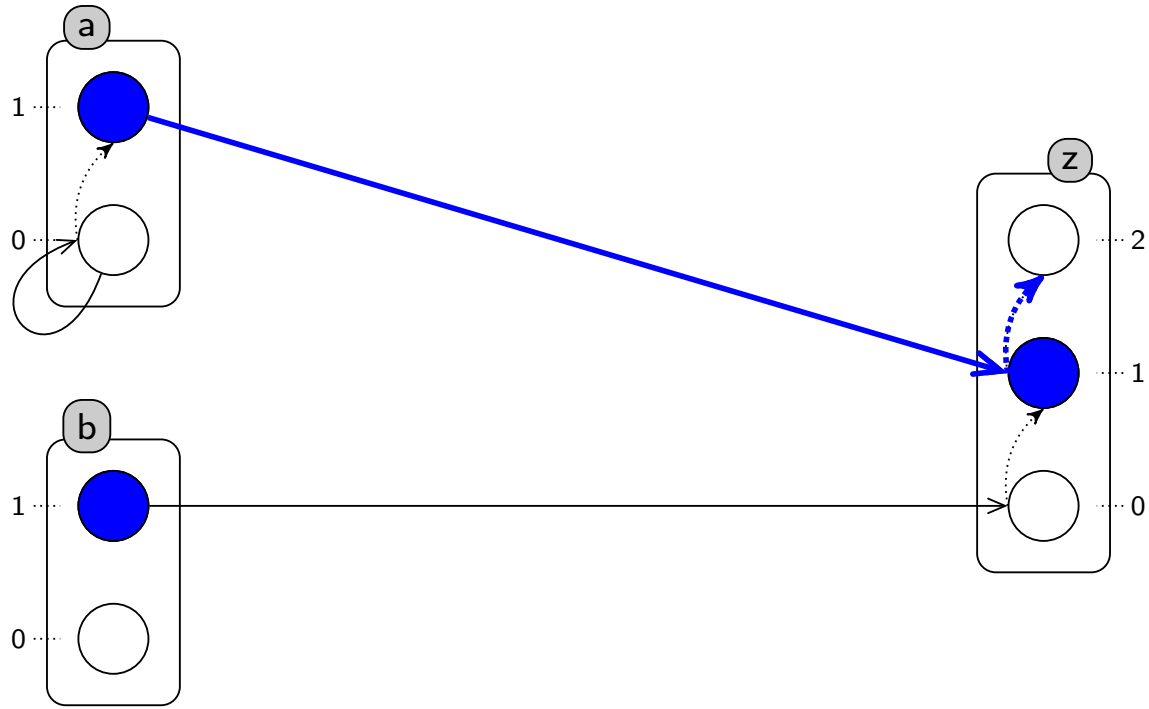
Processes: local states / discrete expression levels z_0, z_1, z_2

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

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

Standard Process Hitting

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



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

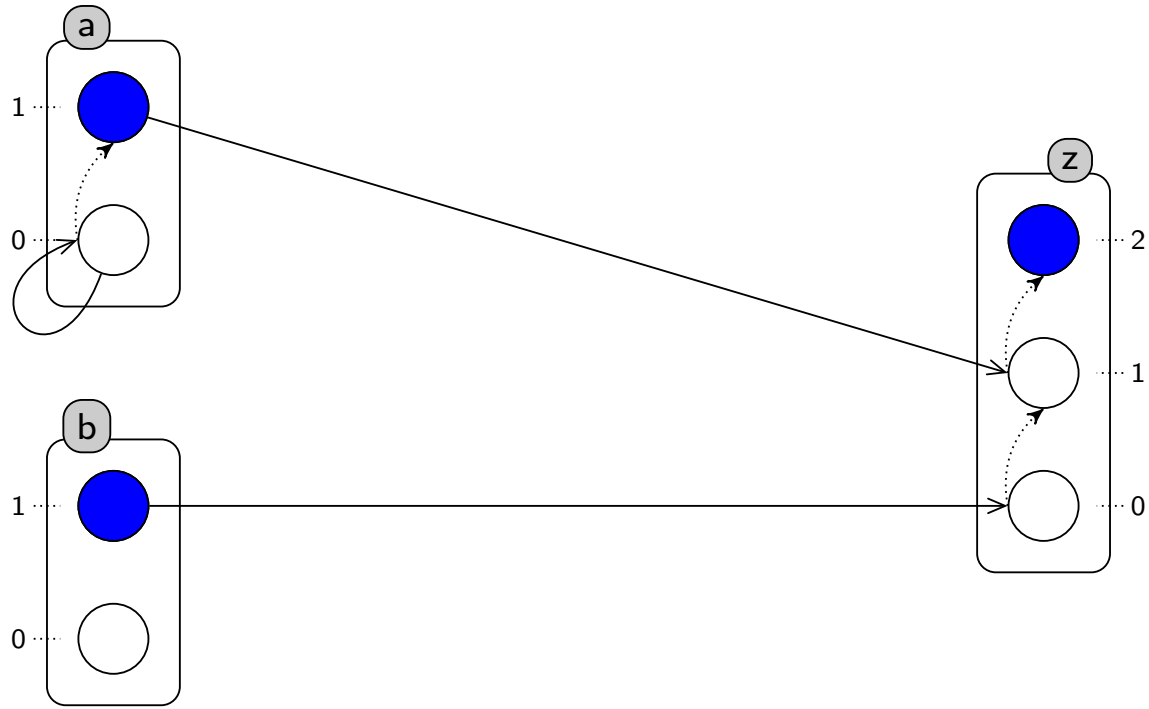
Processes: local states / discrete expression levels $z_0,\ z_1,\ z_2$

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

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

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 $\langle a_1, b_1, z_2 \rangle$

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

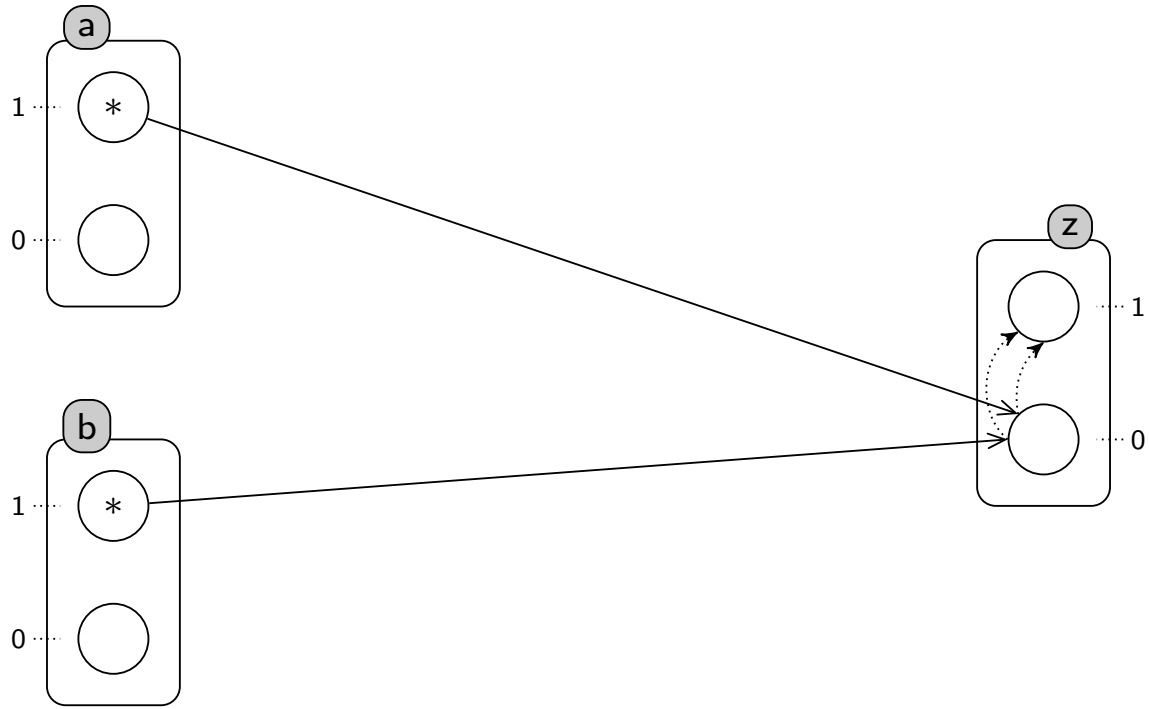
Cooperations

[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

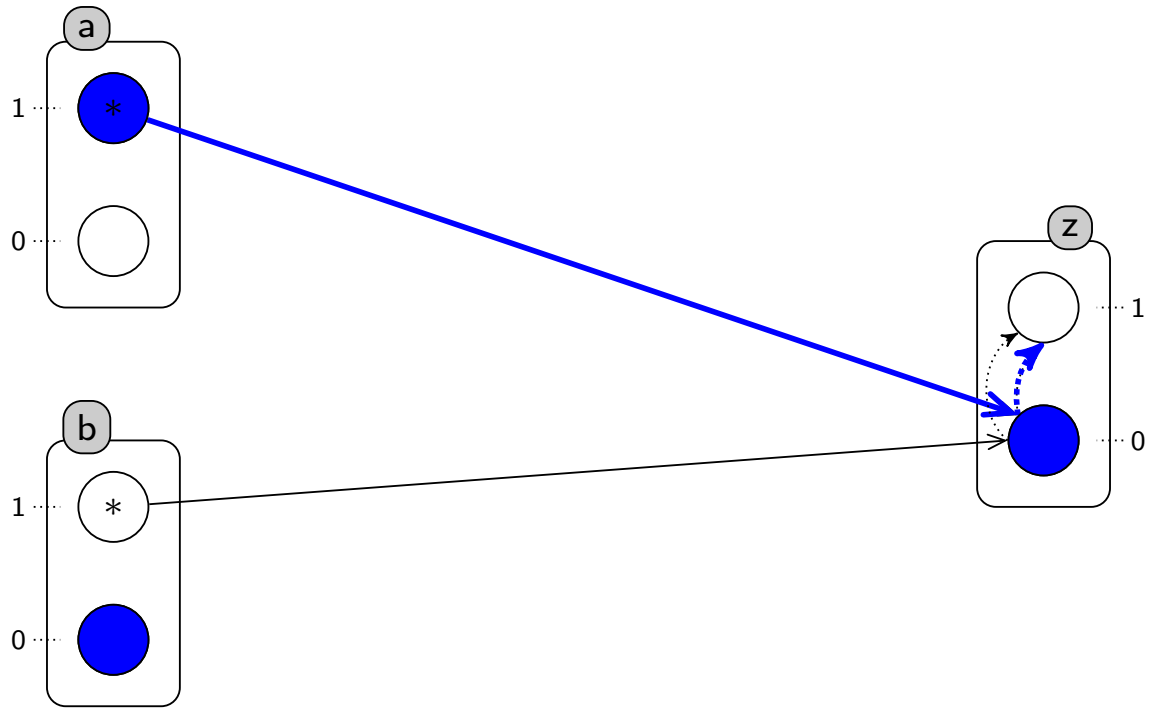
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



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

Cooperations

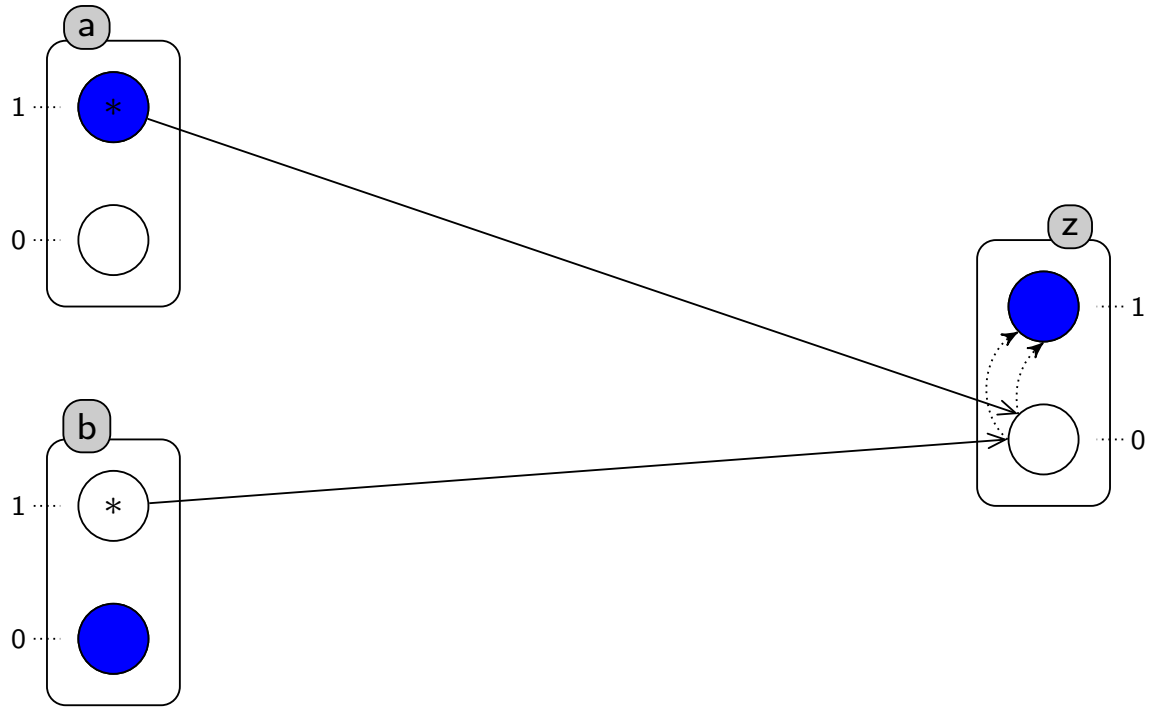
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



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

Cooperations

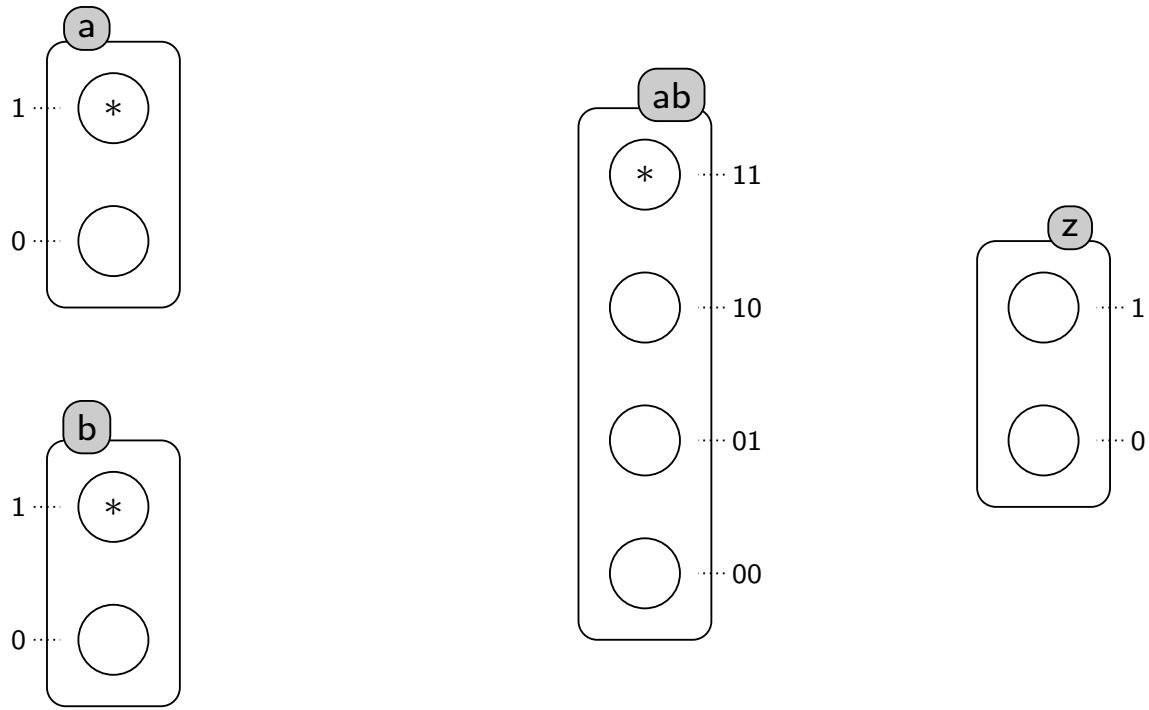
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



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

Cooperations

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

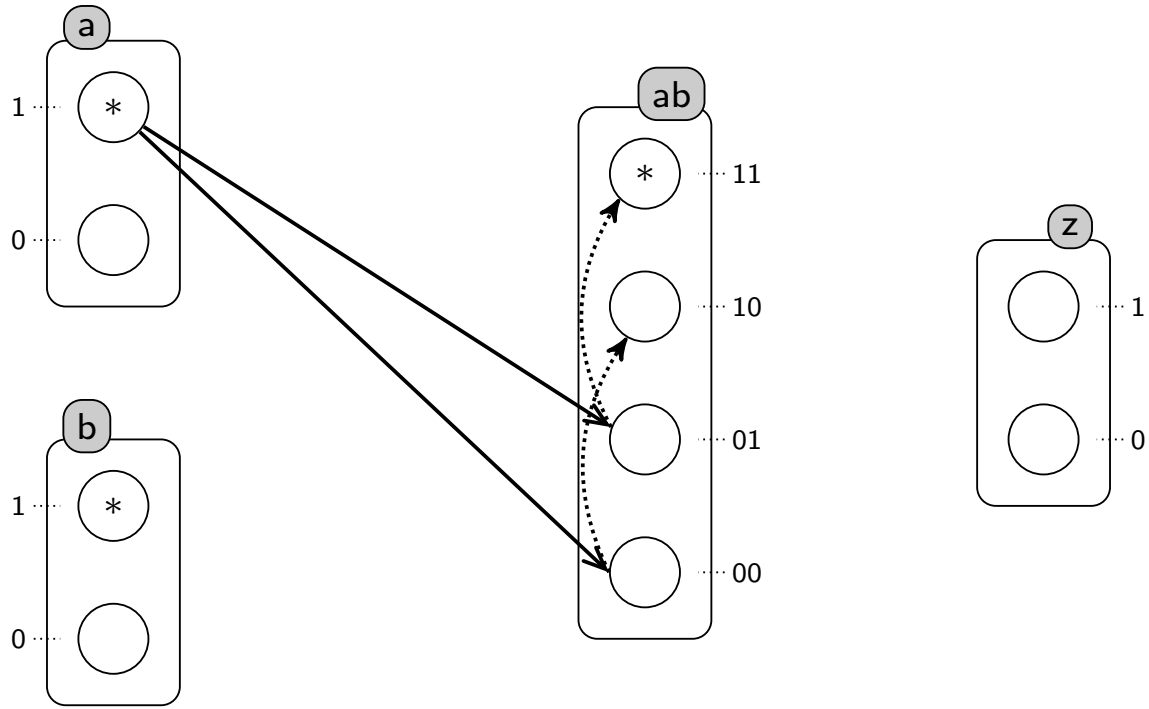


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

Cooperations

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

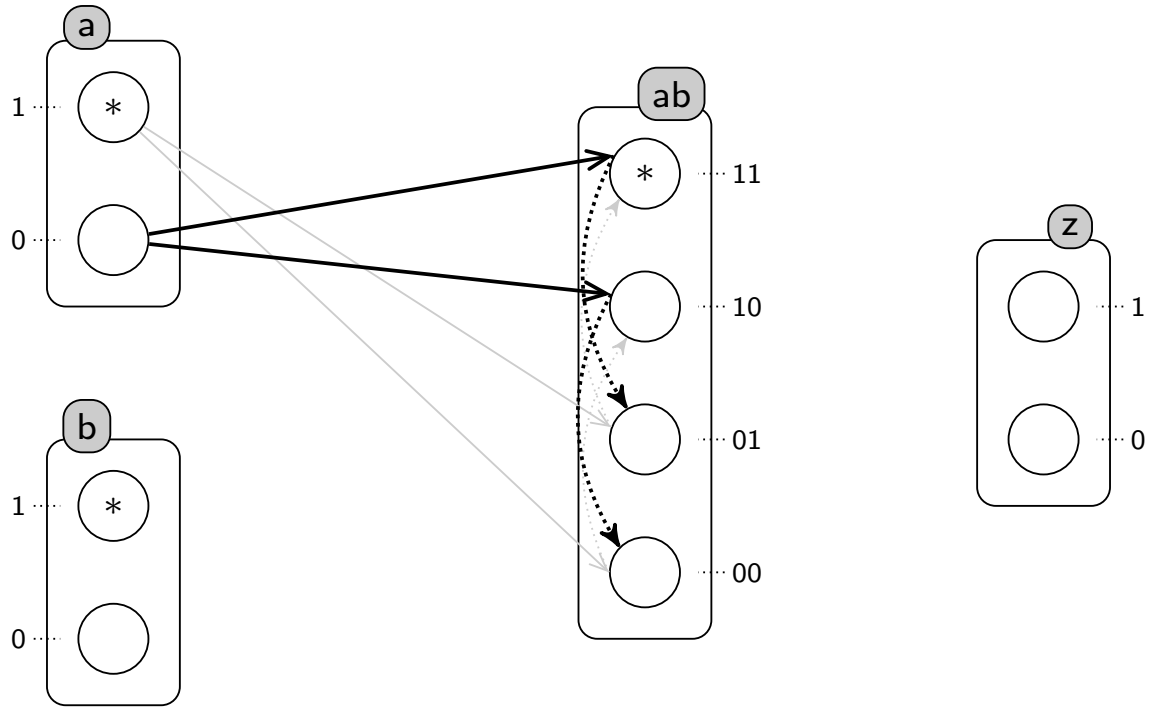


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

Cooperations

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

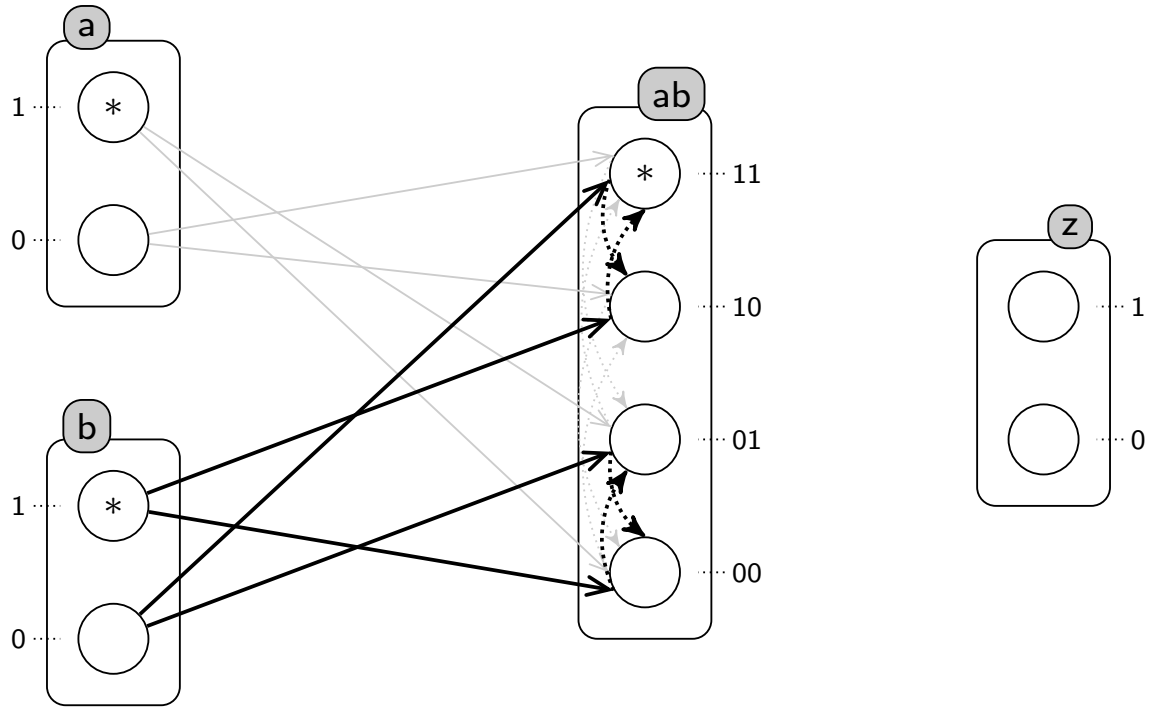


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

Cooperations

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

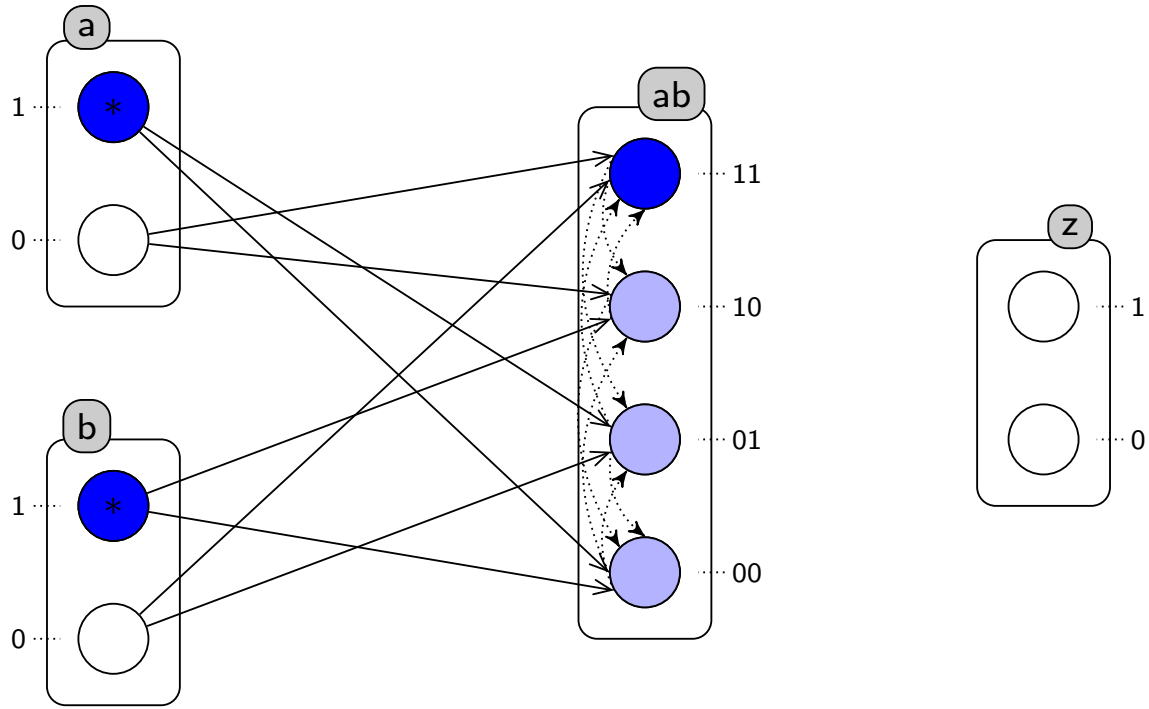


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

Cooperations

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



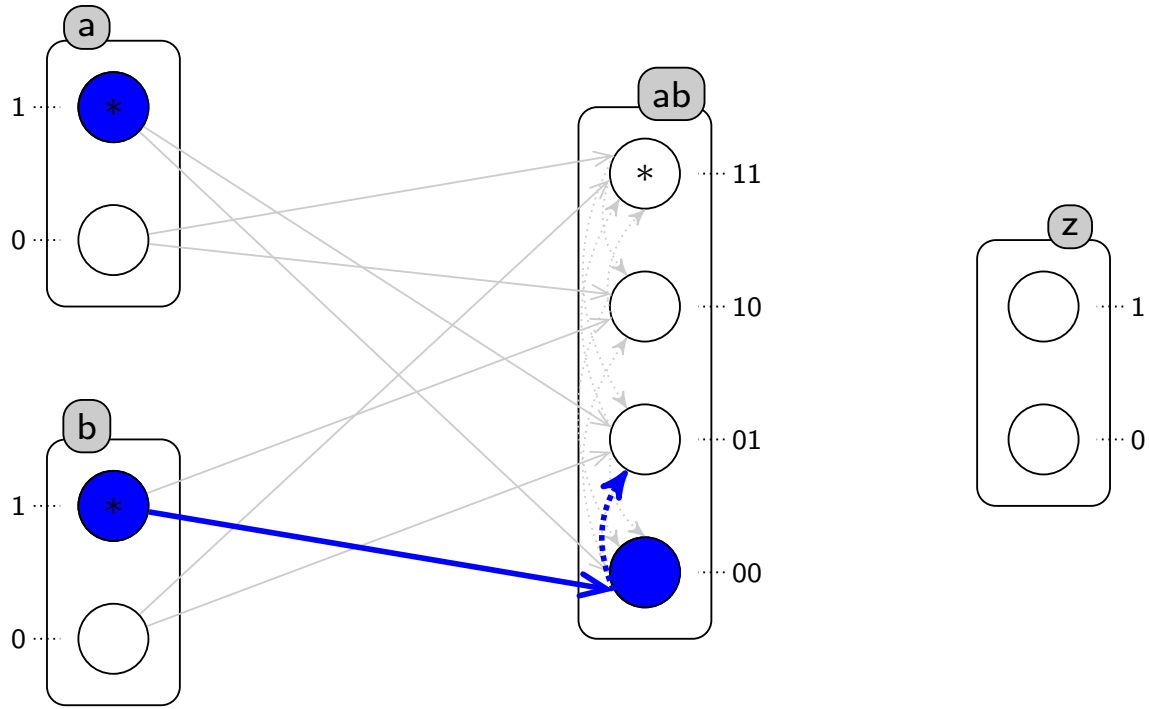
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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]



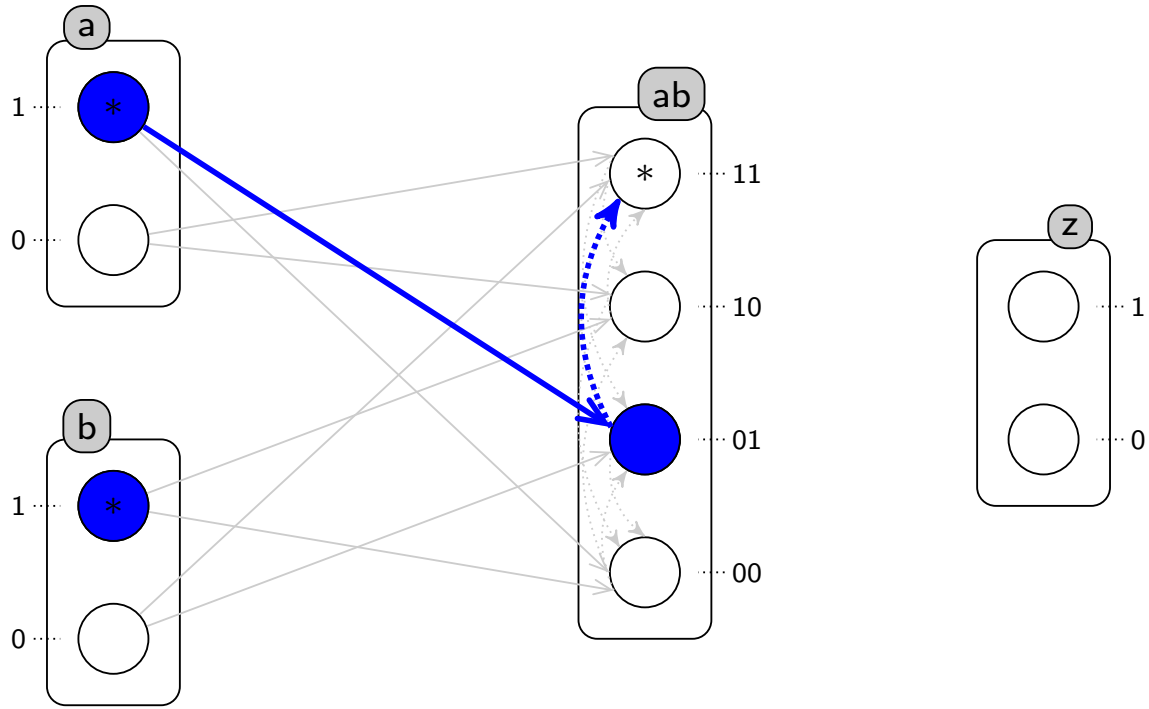
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \nearrow z_1$

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]



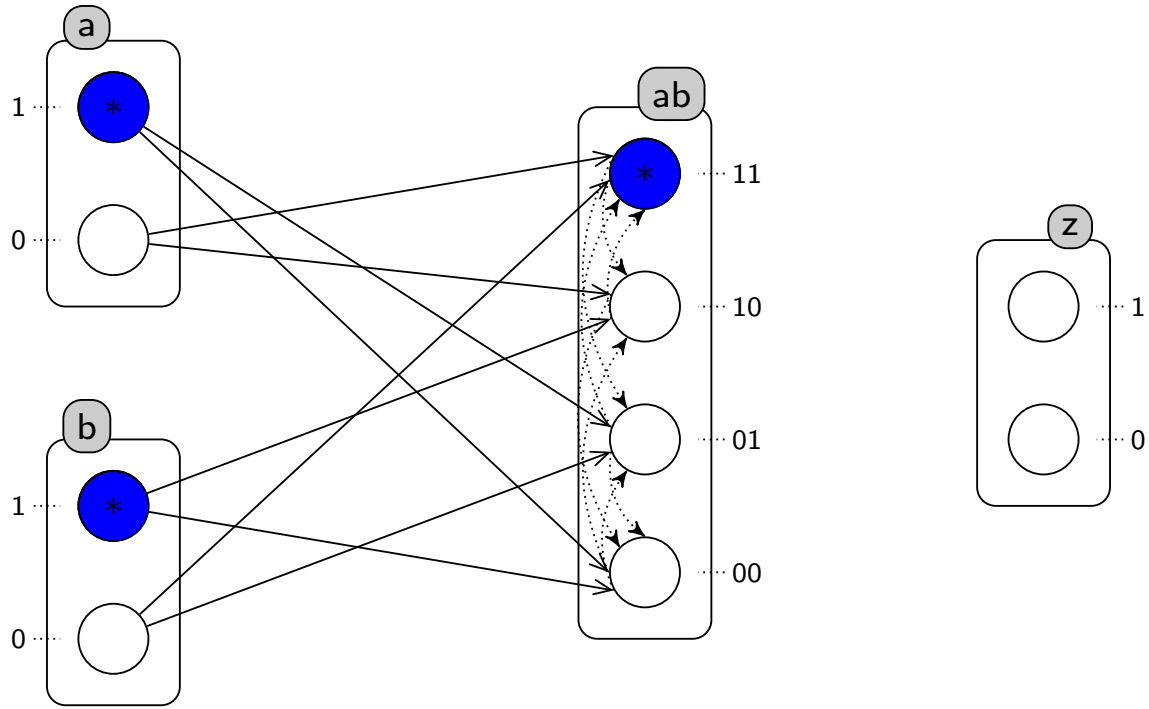
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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]



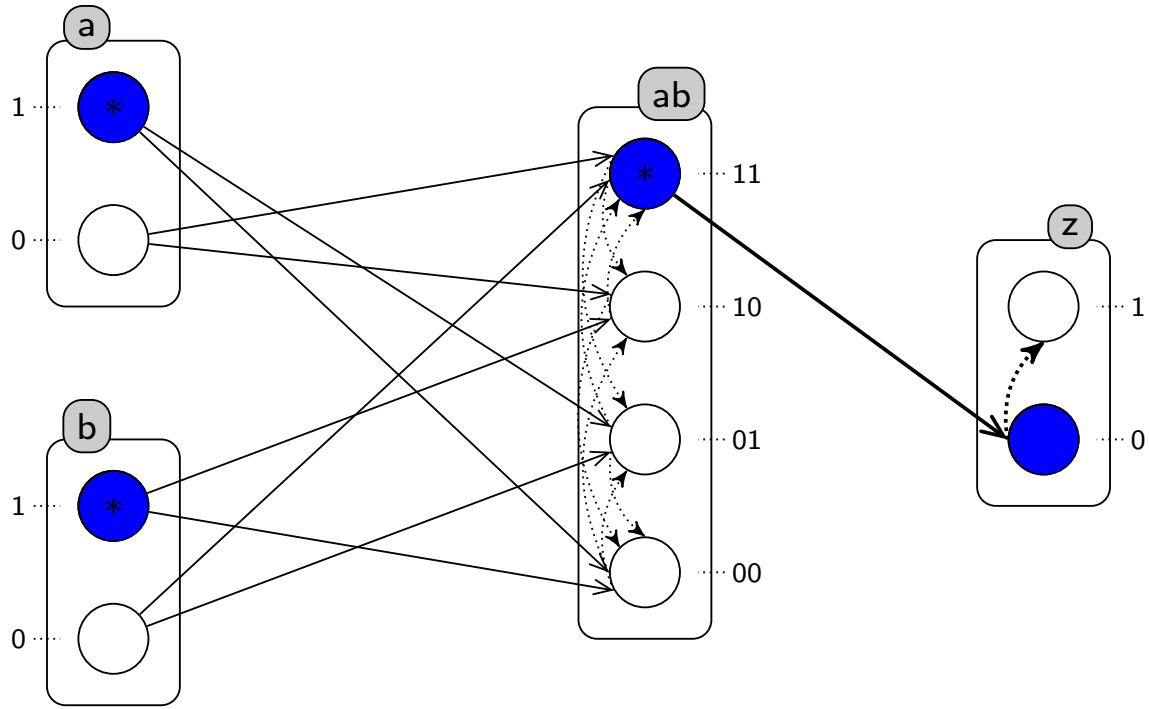
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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]



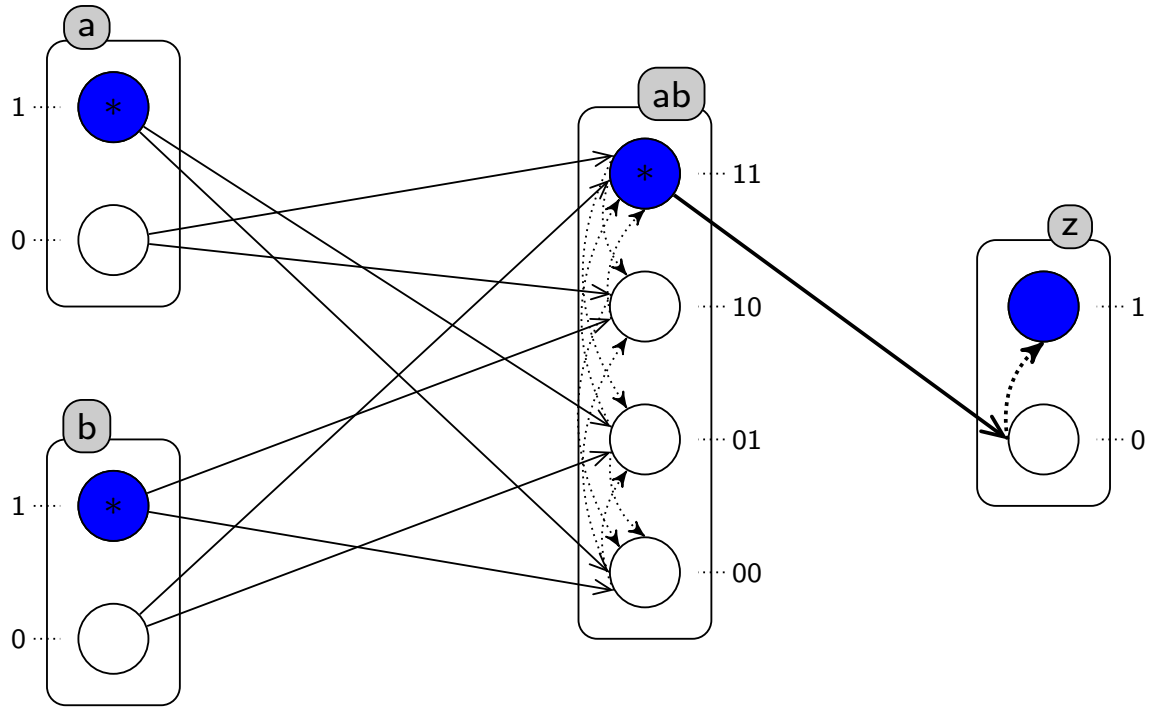
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

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

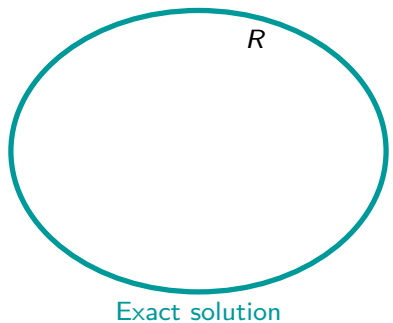
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



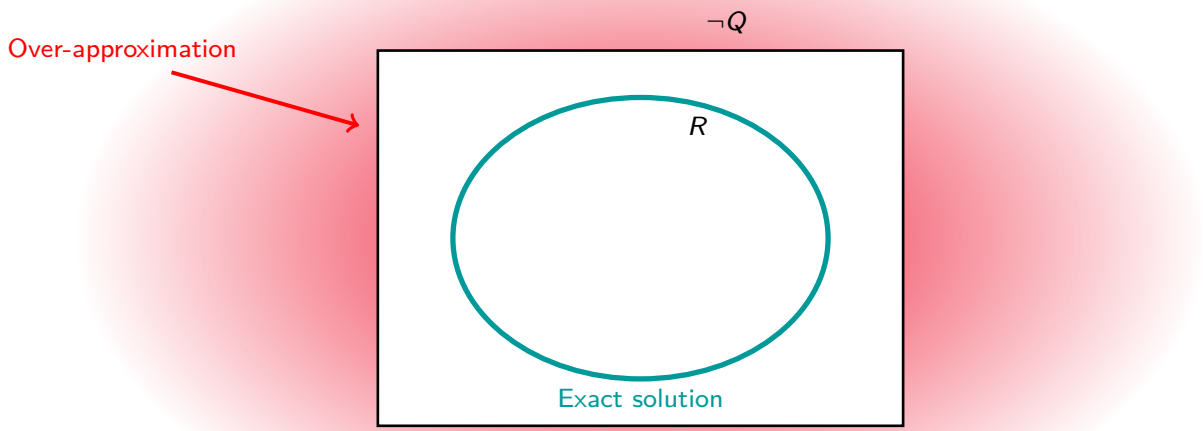
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



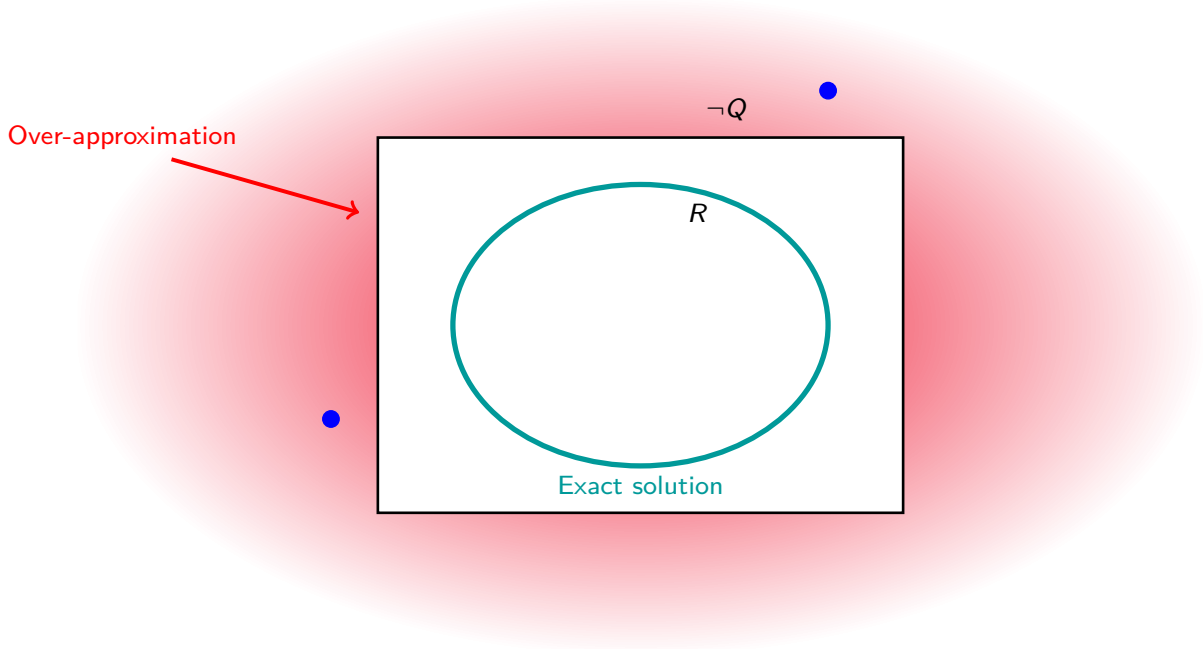
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



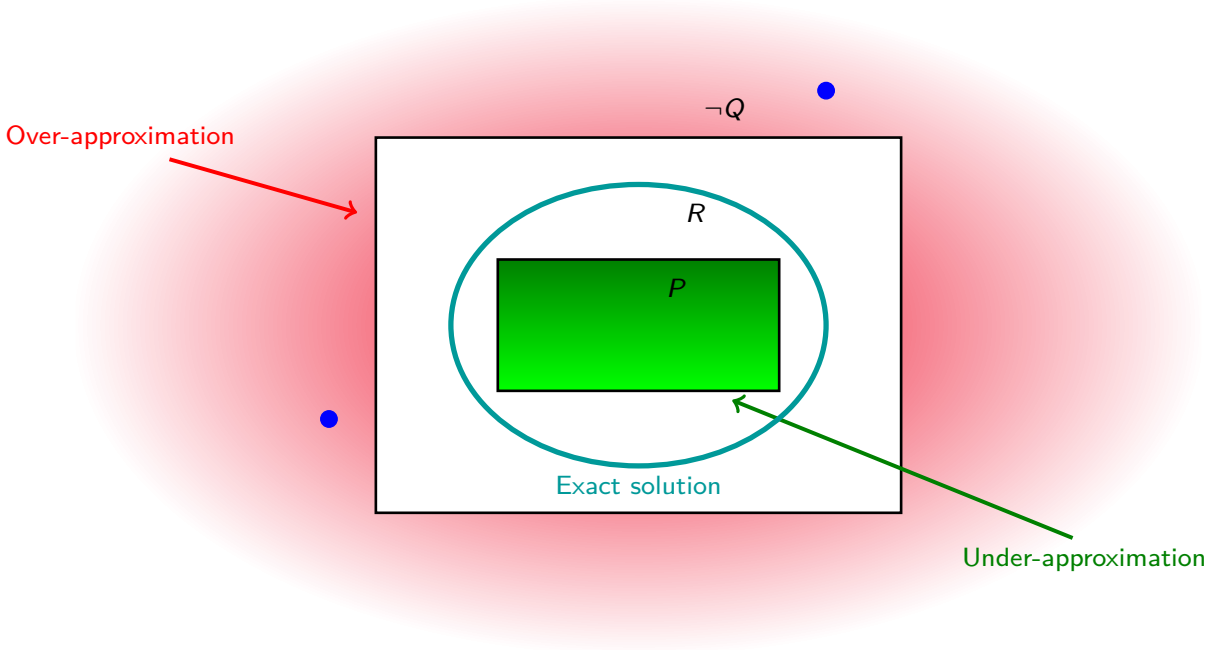
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



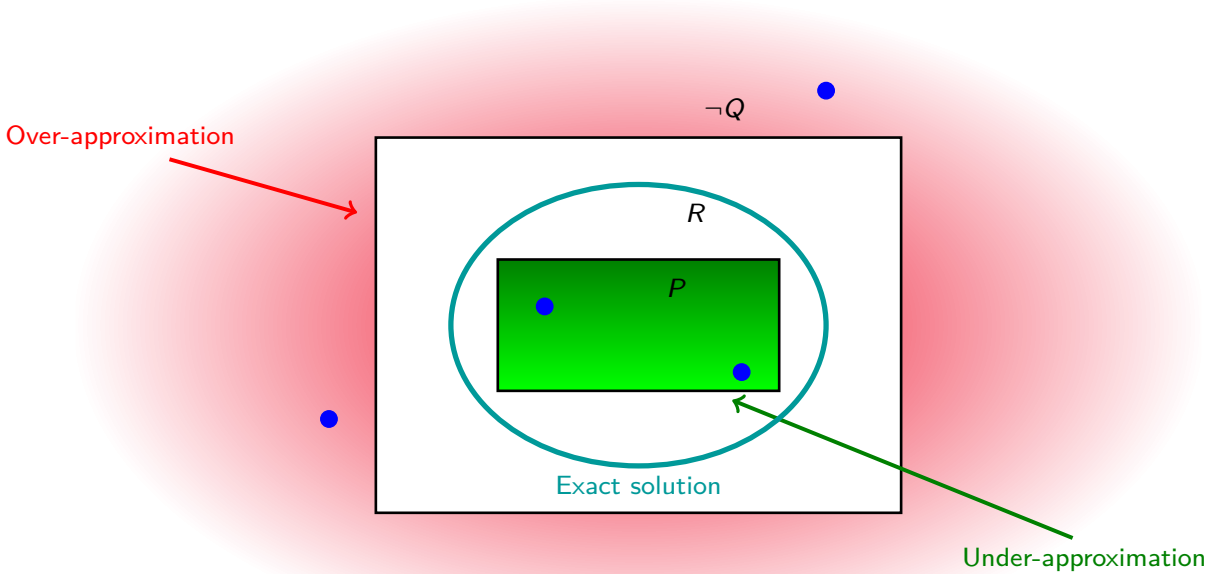
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



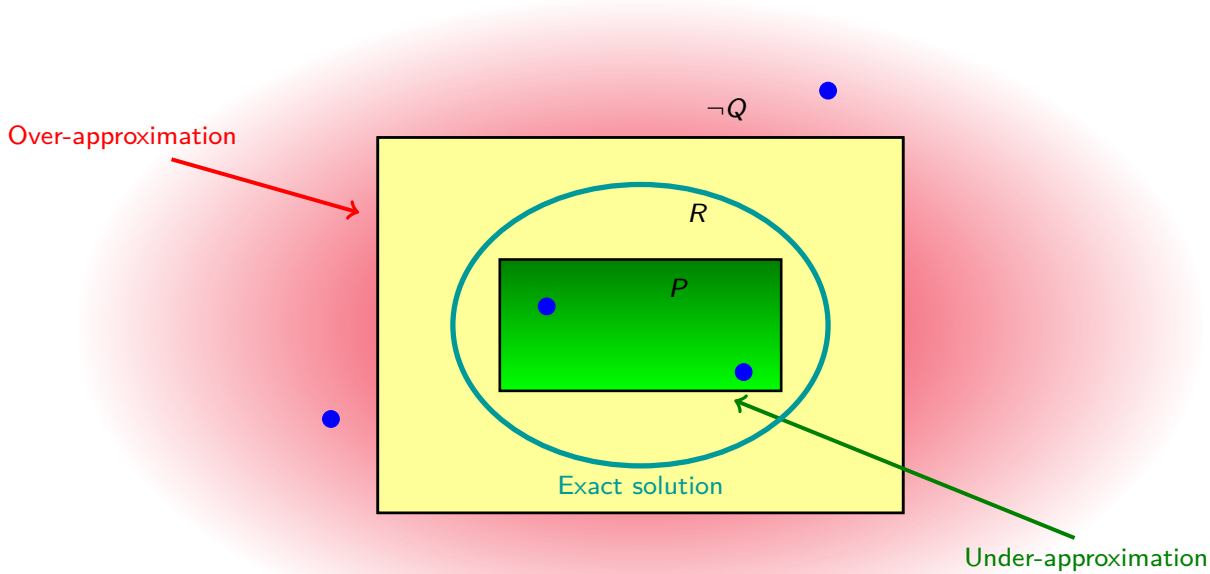
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



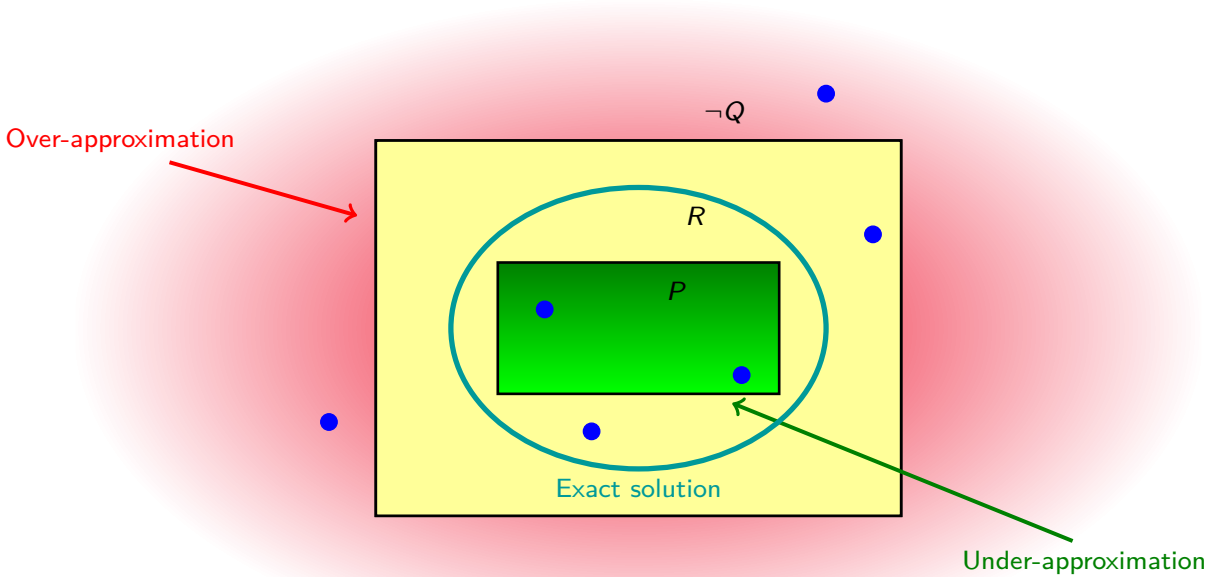
Approximations for the Reachability Analysis

[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 $P \Rightarrow R \Rightarrow Q$



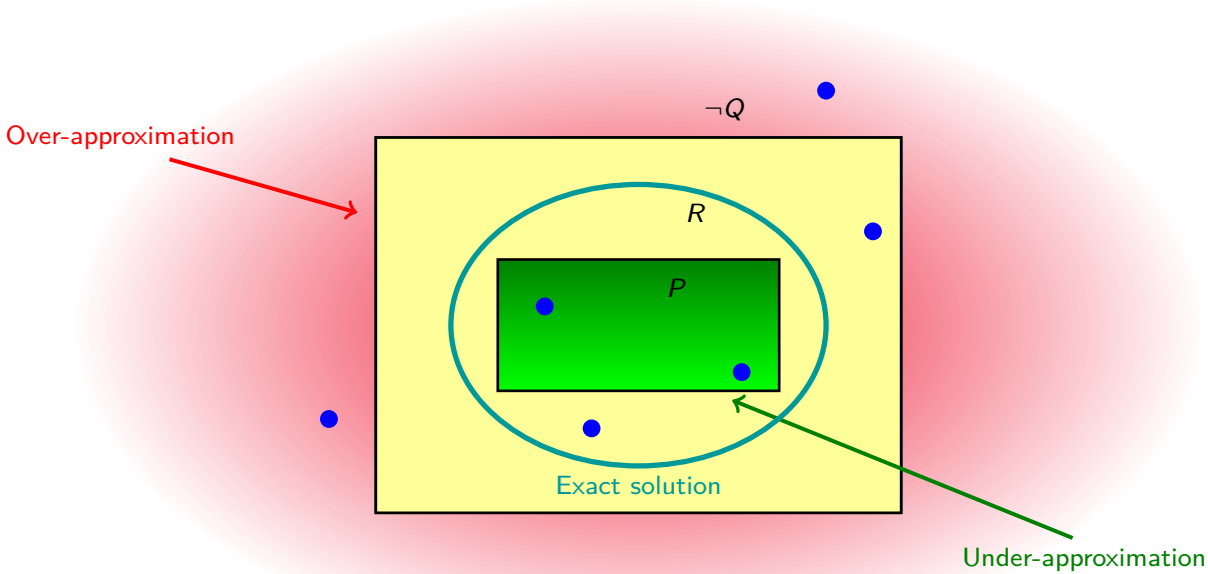
Approximations for the Reachability Analysis

[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? »

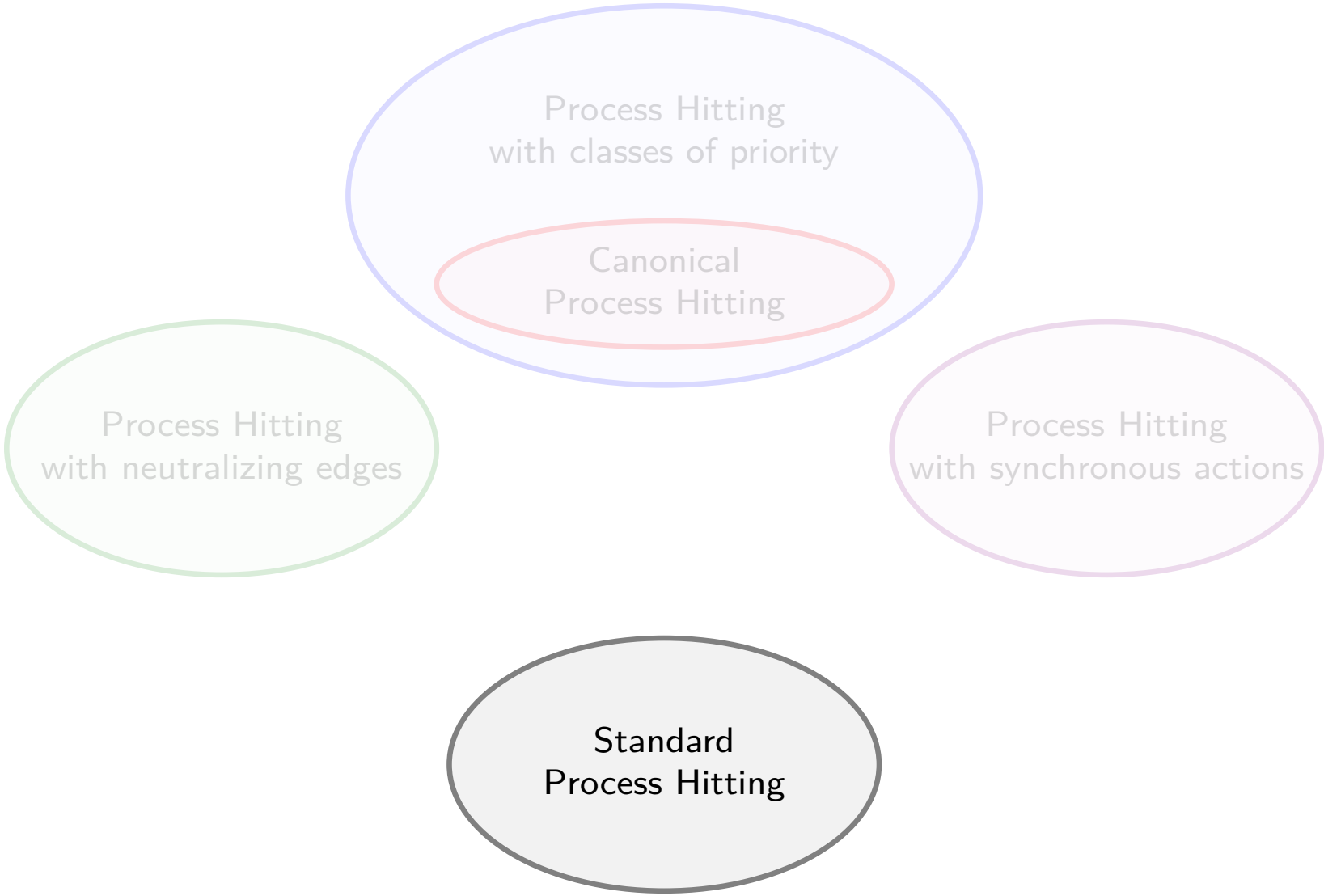
Approximations: P and Q, built so that $P \Rightarrow R \Rightarrow Q$



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

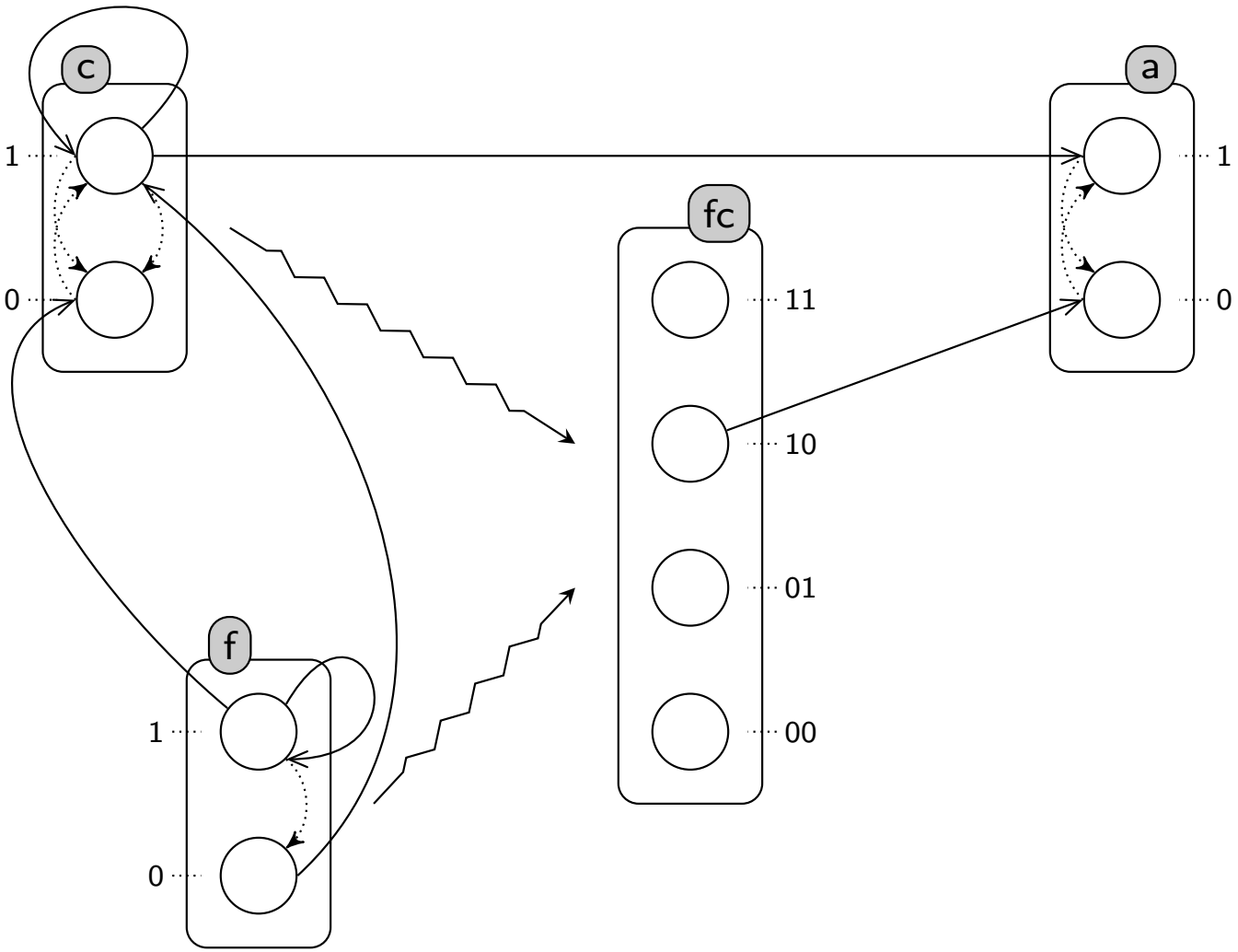
→ Efficient for big models with few expression levels

Standard Process Hitting



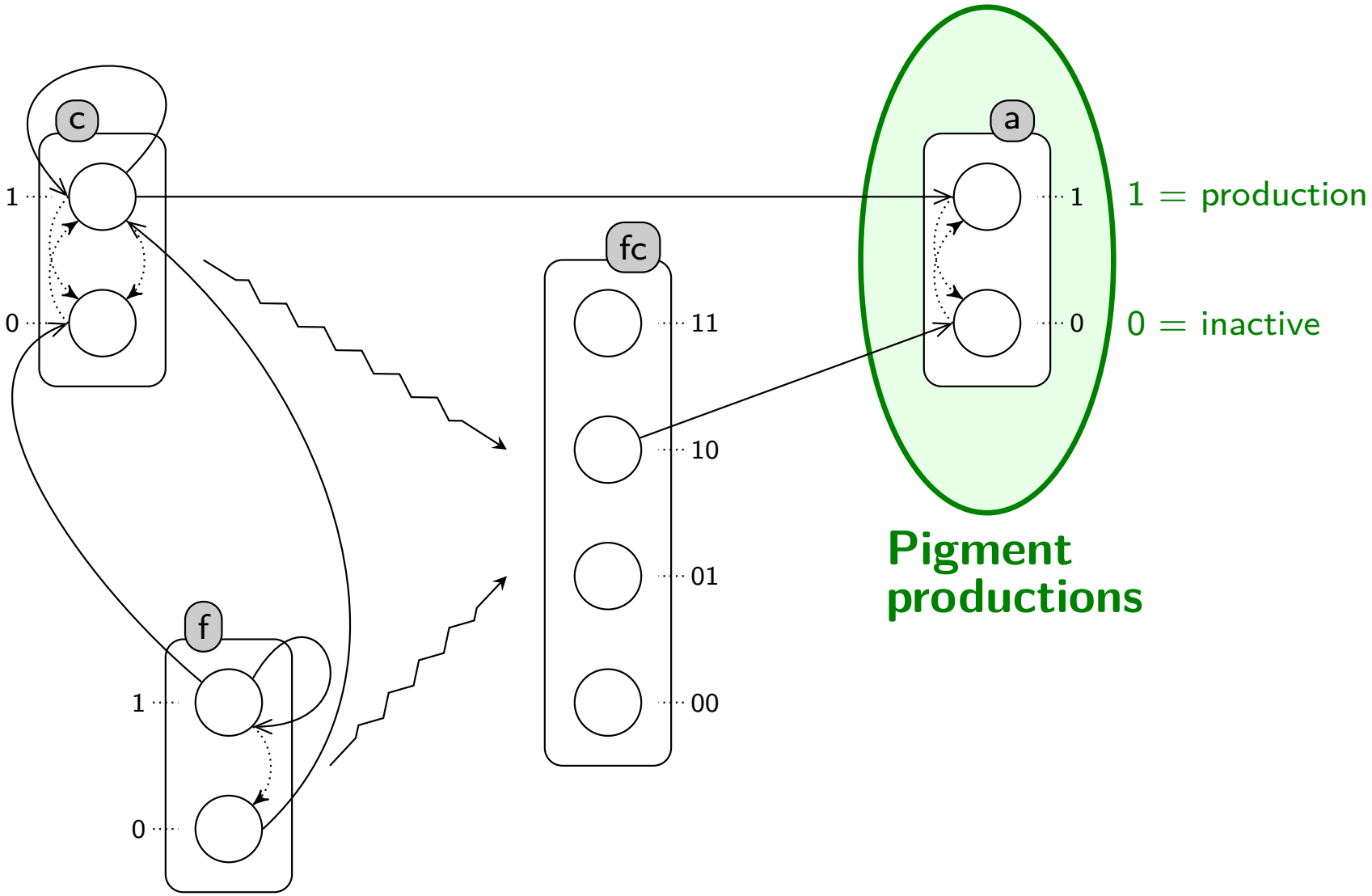
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



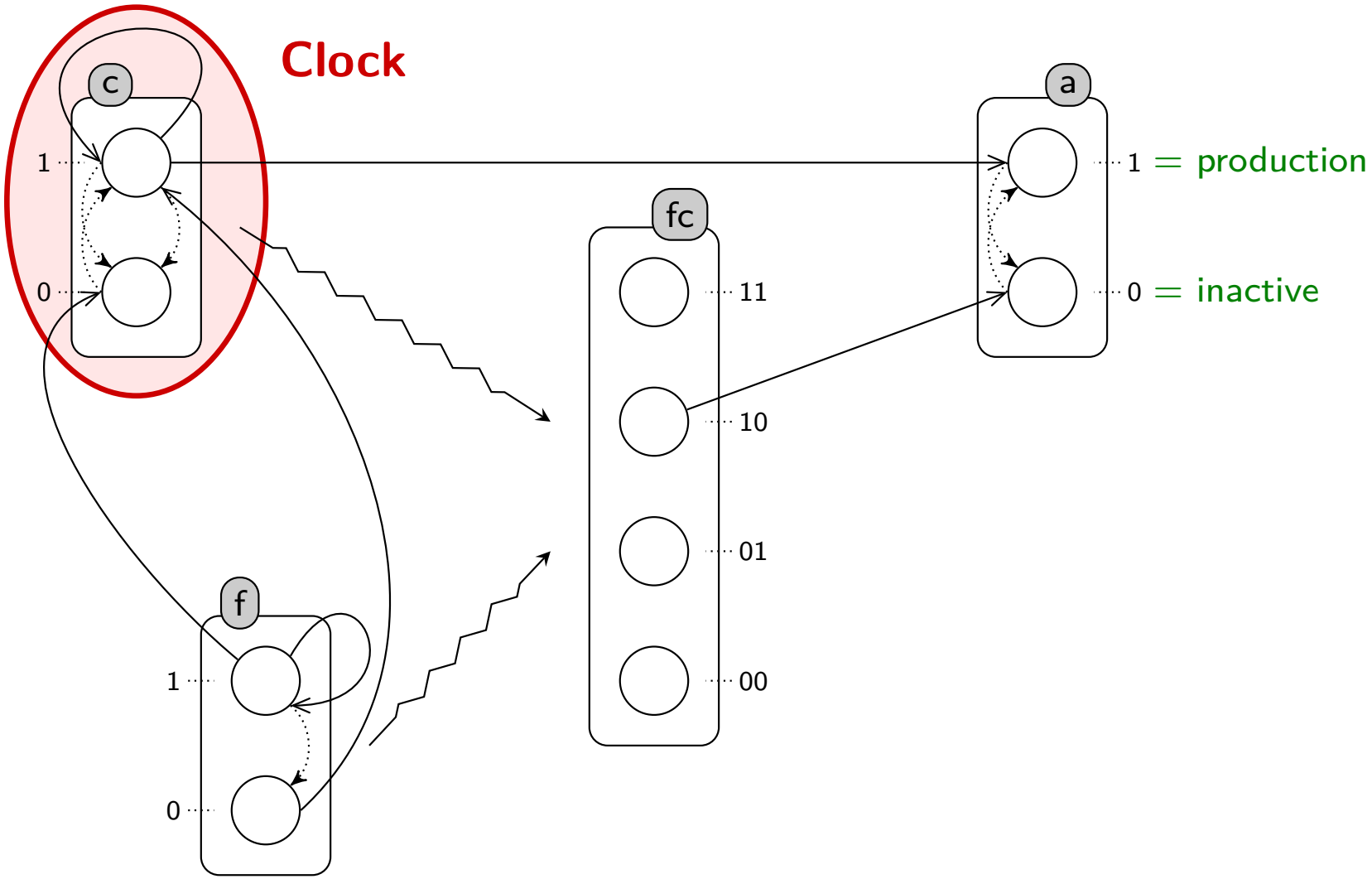
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



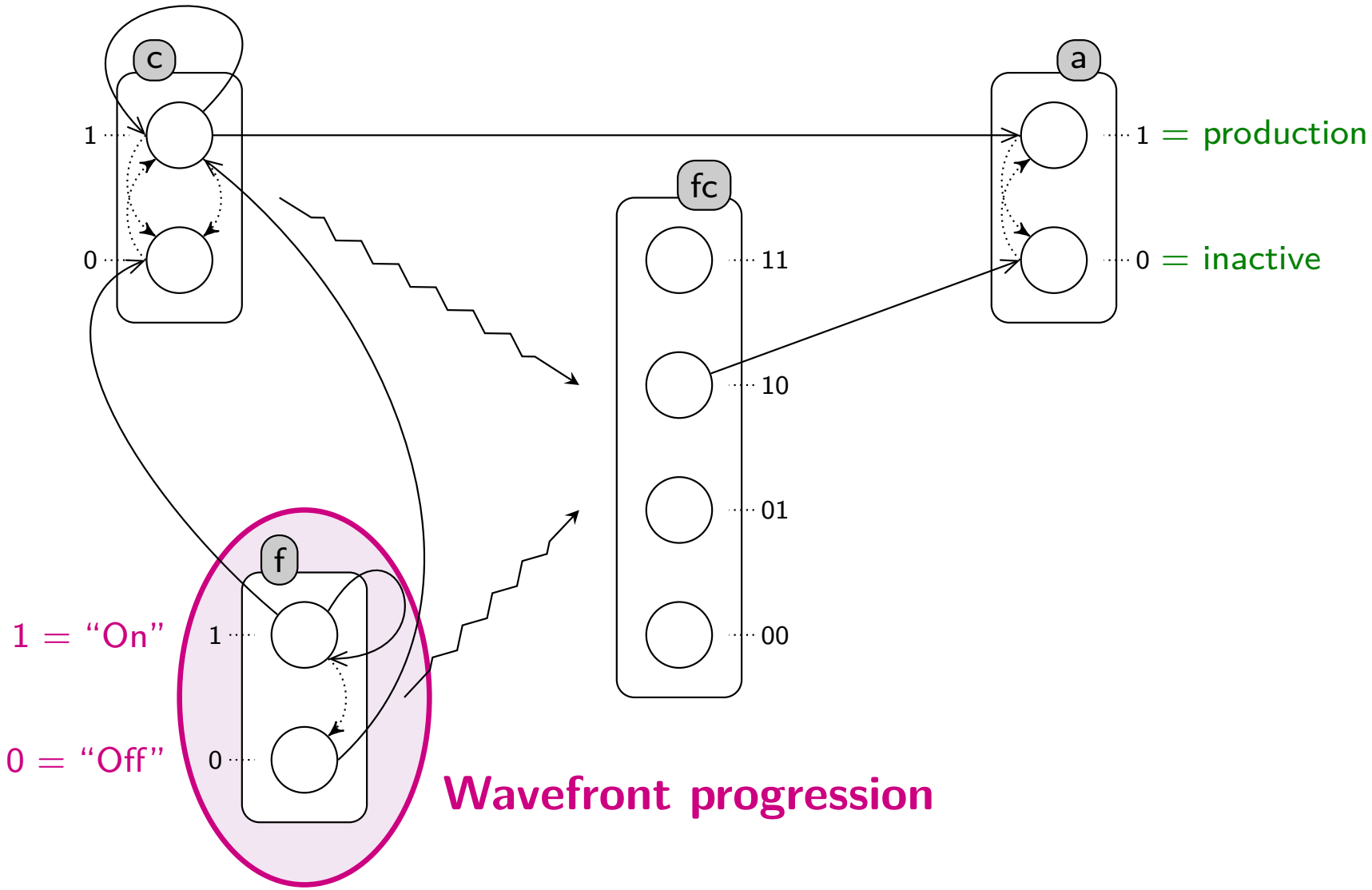
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



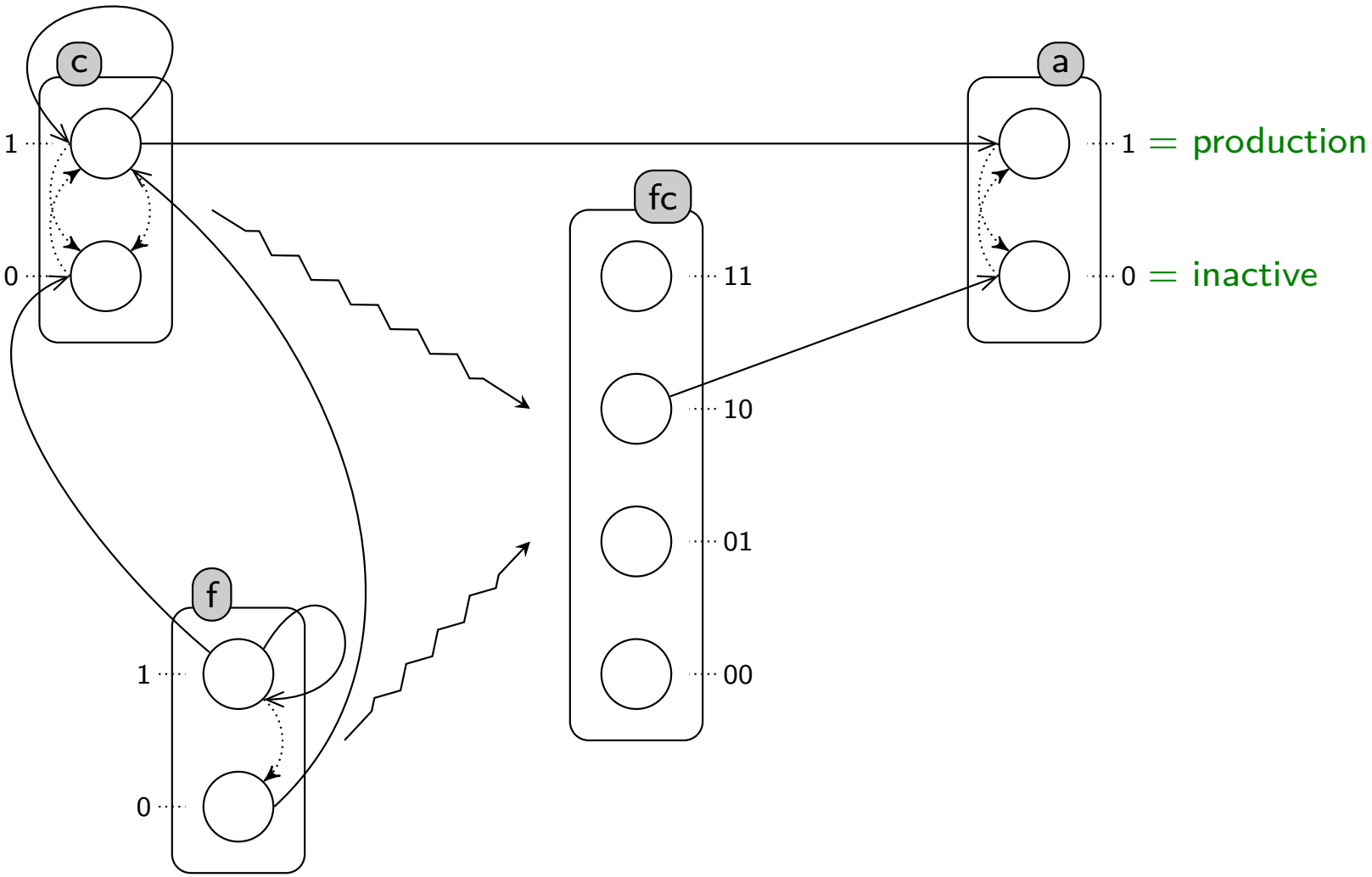
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



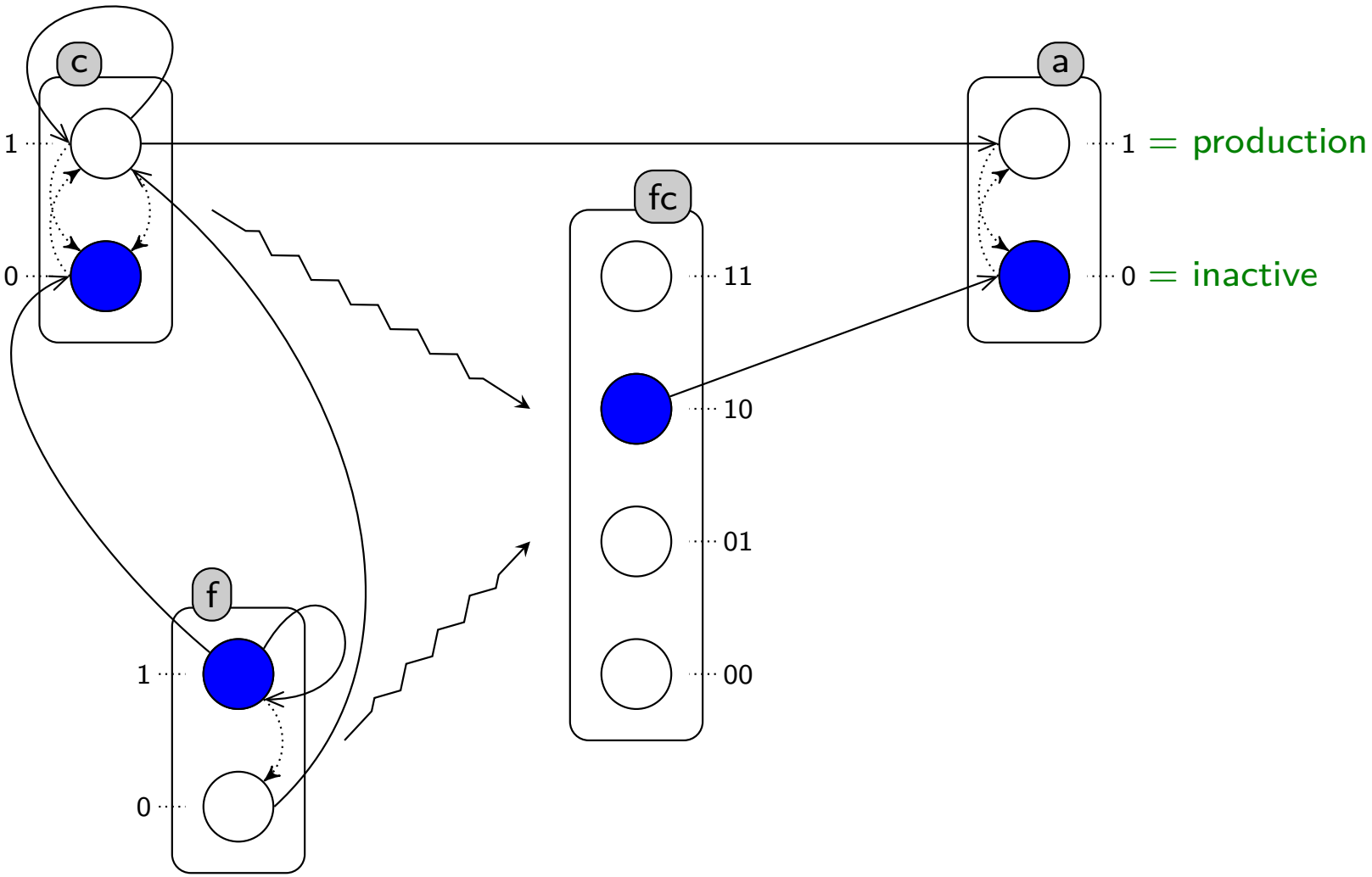
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



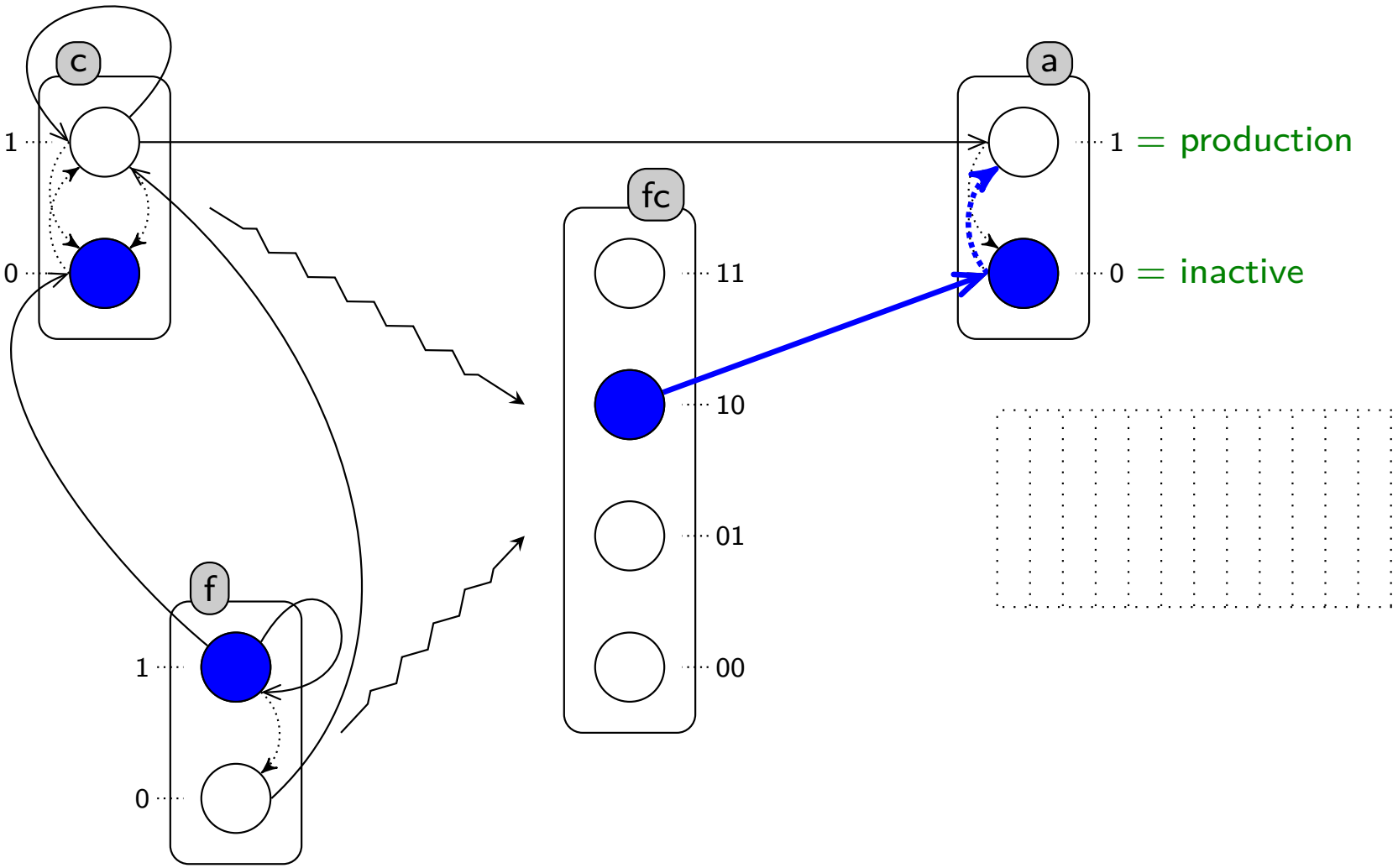
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



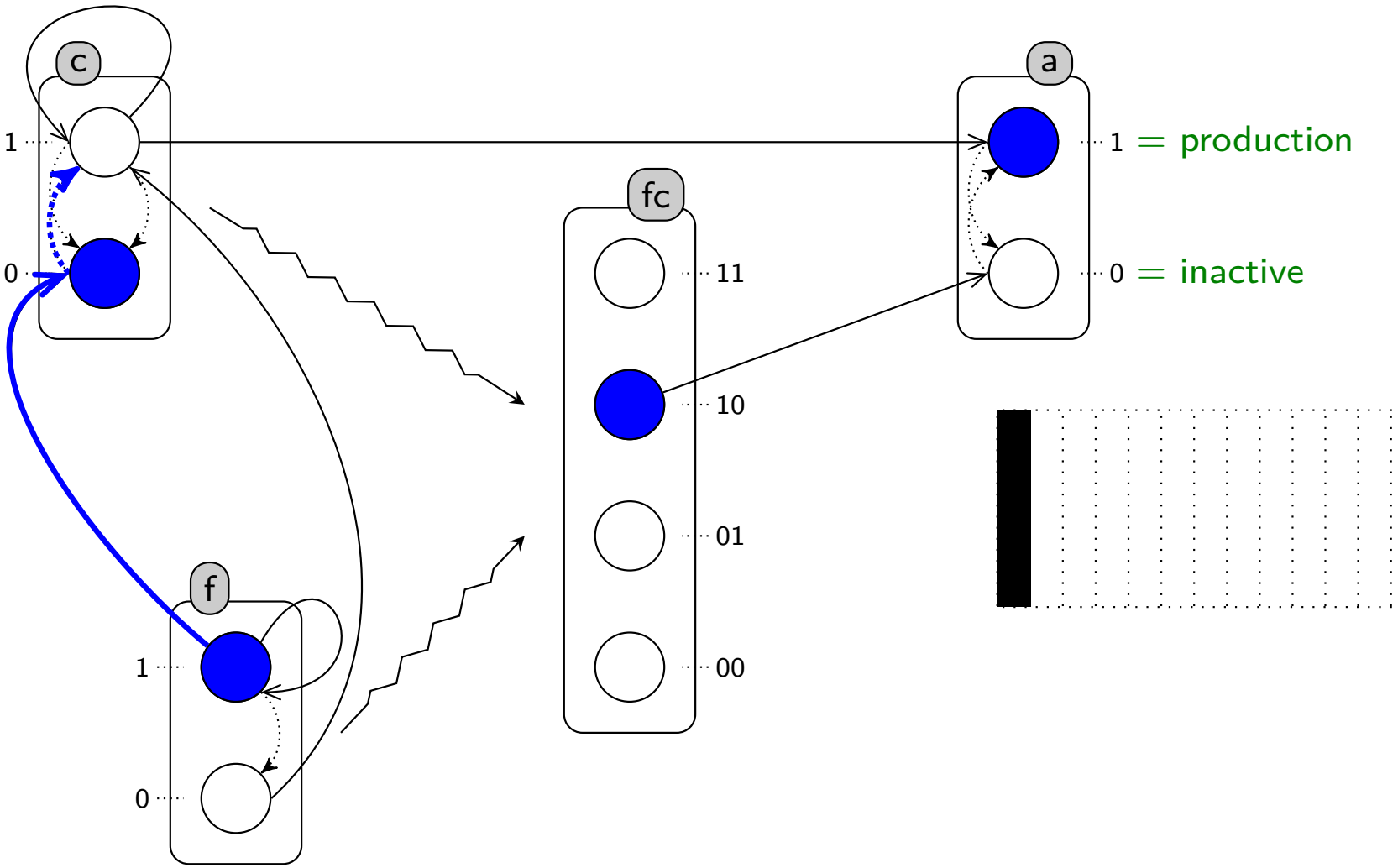
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



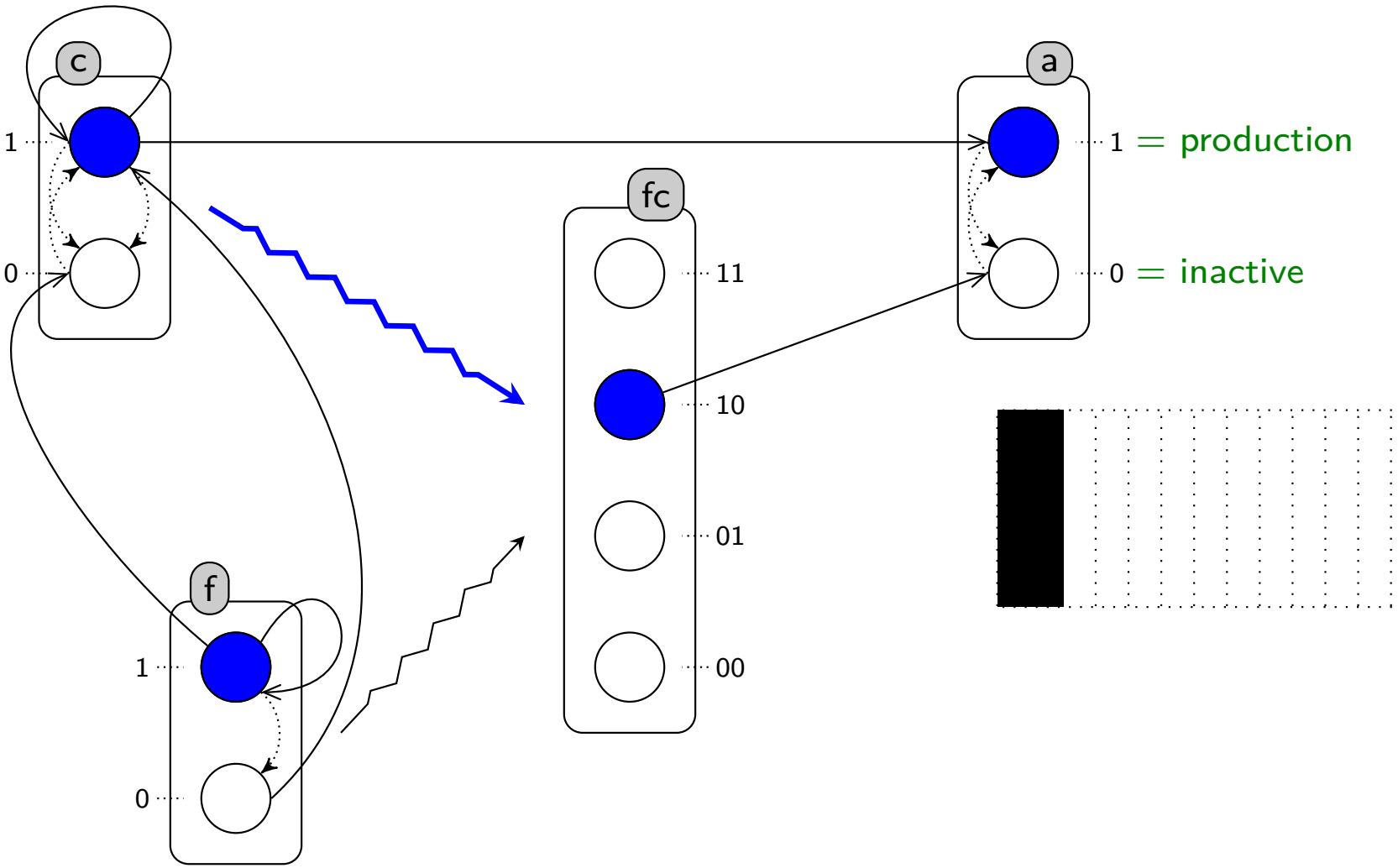
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



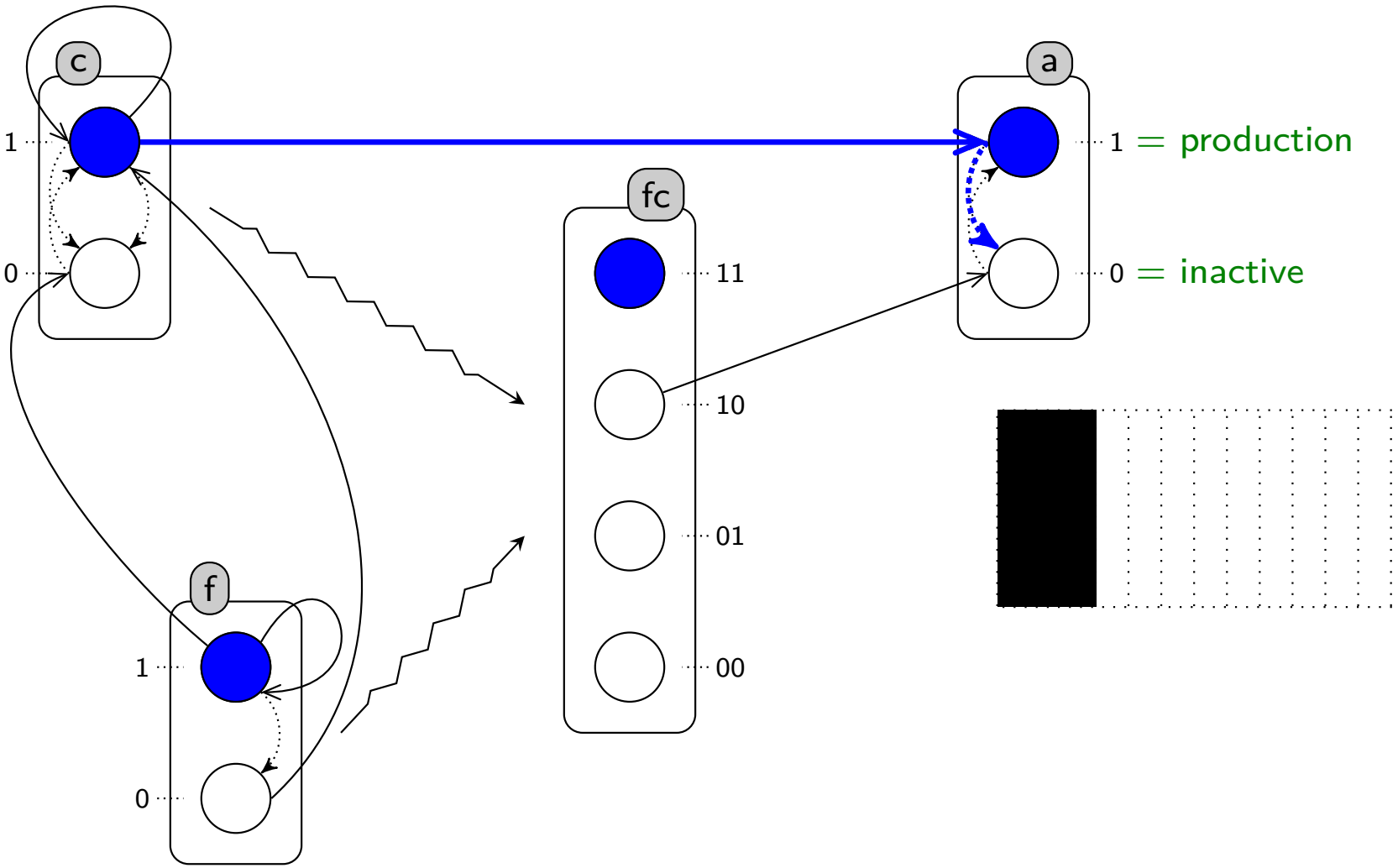
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



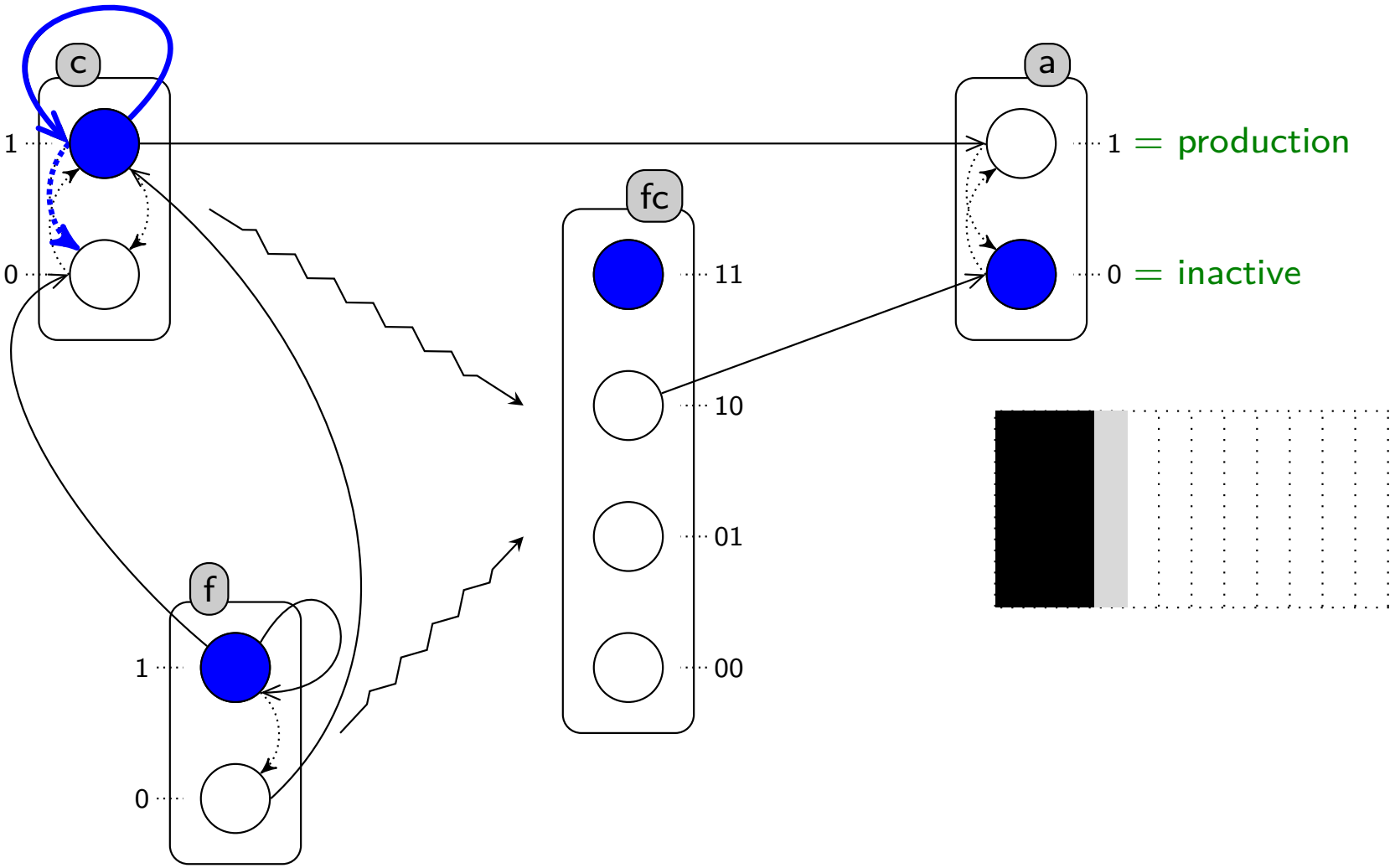
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



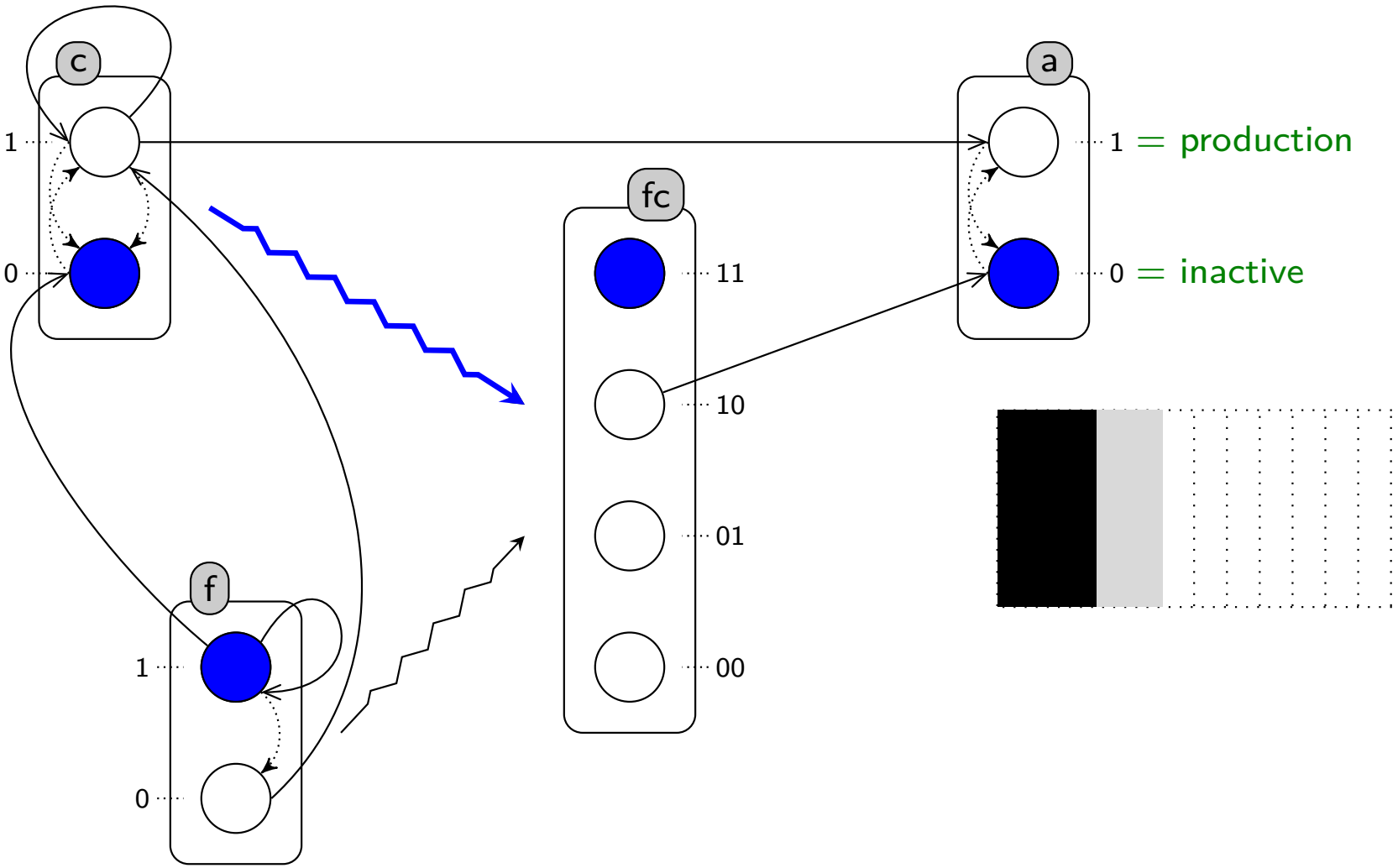
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



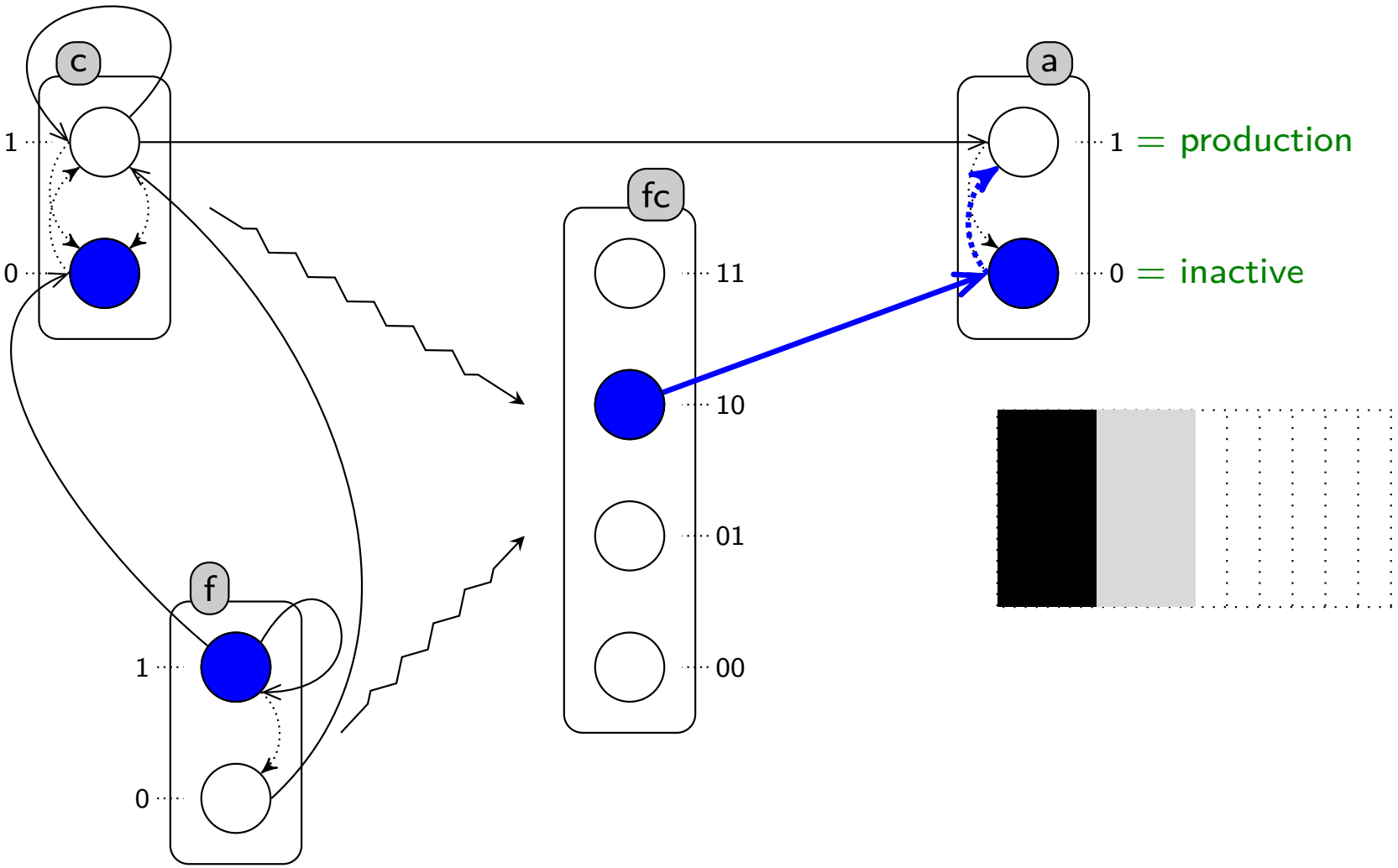
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



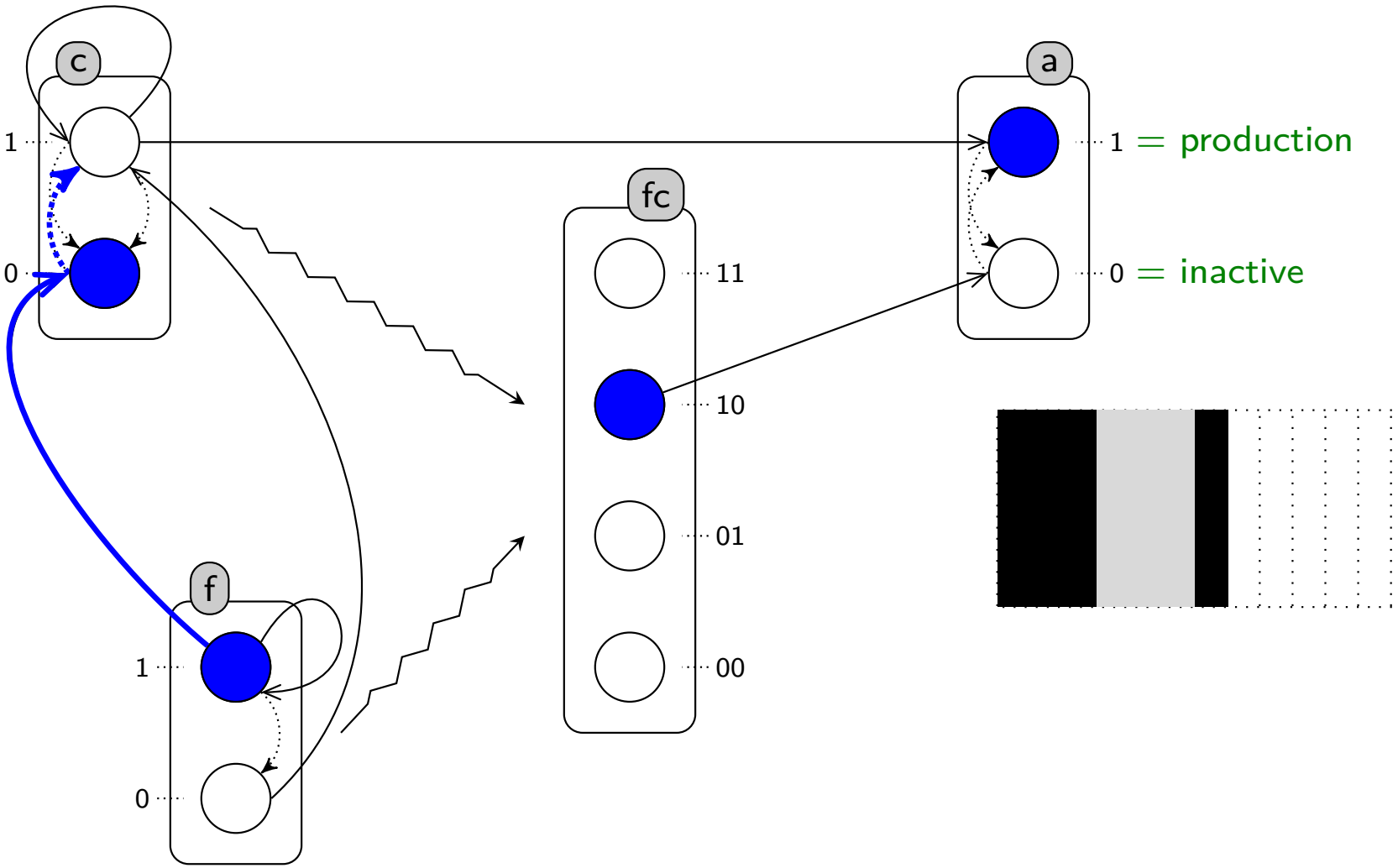
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



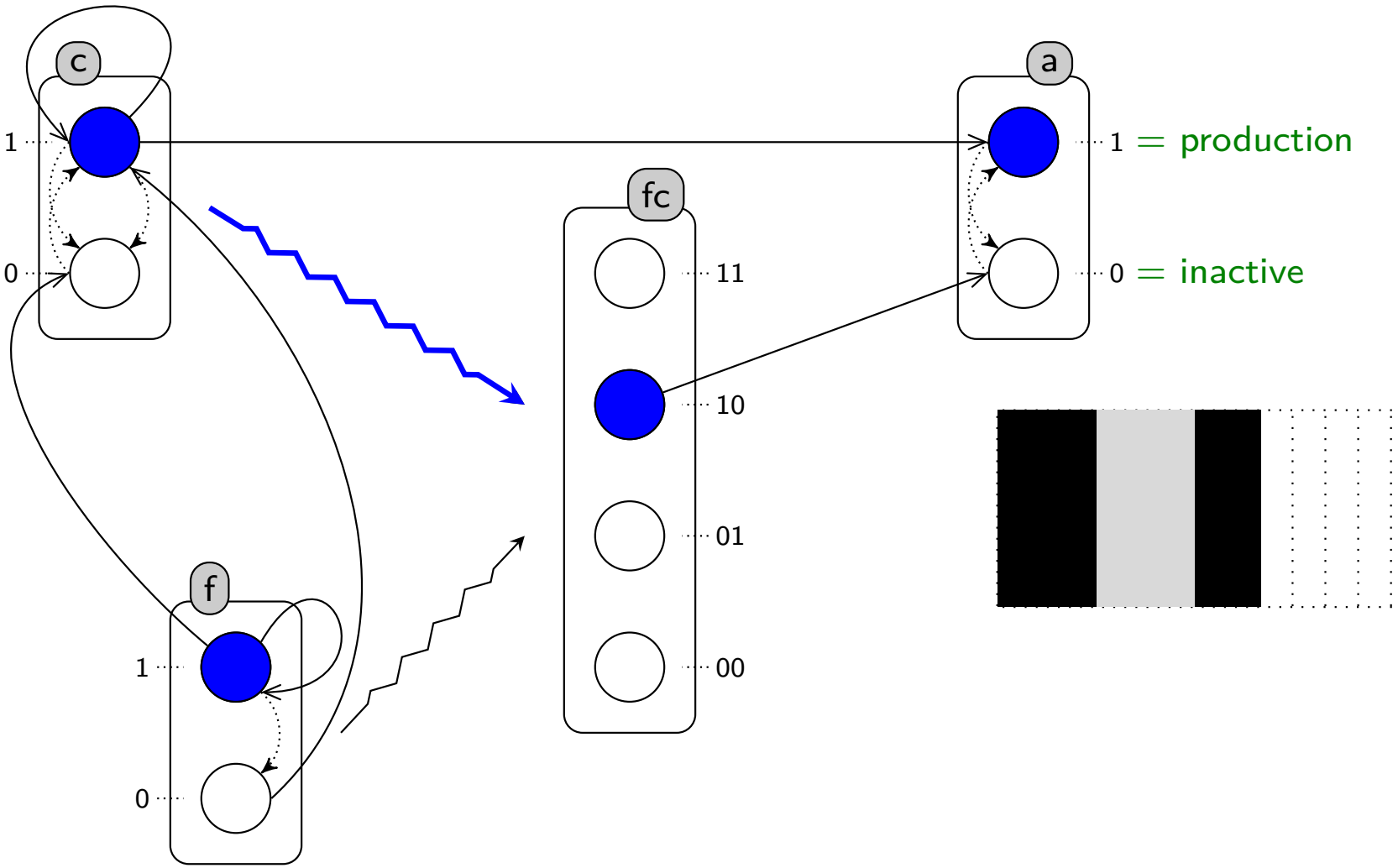
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



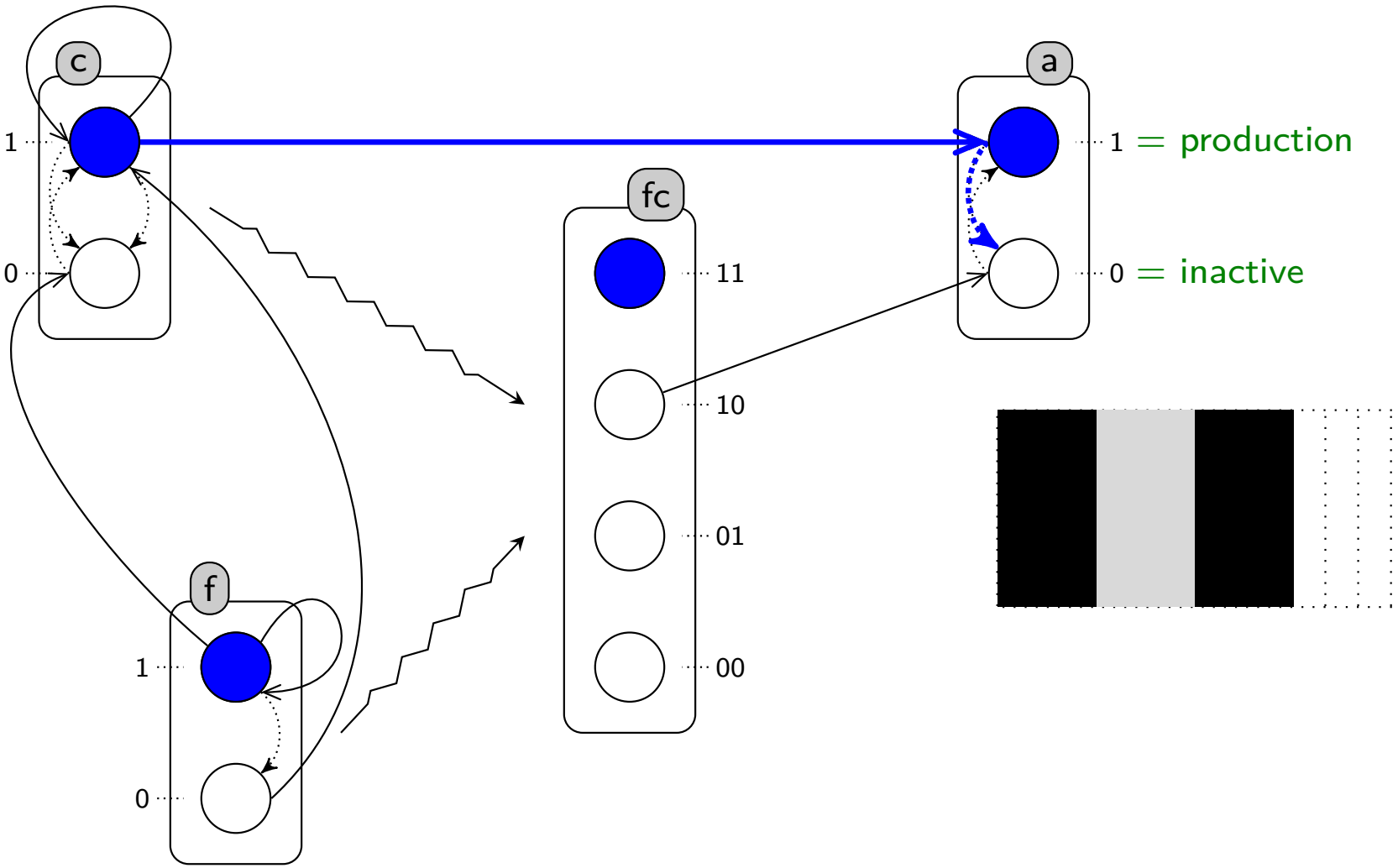
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



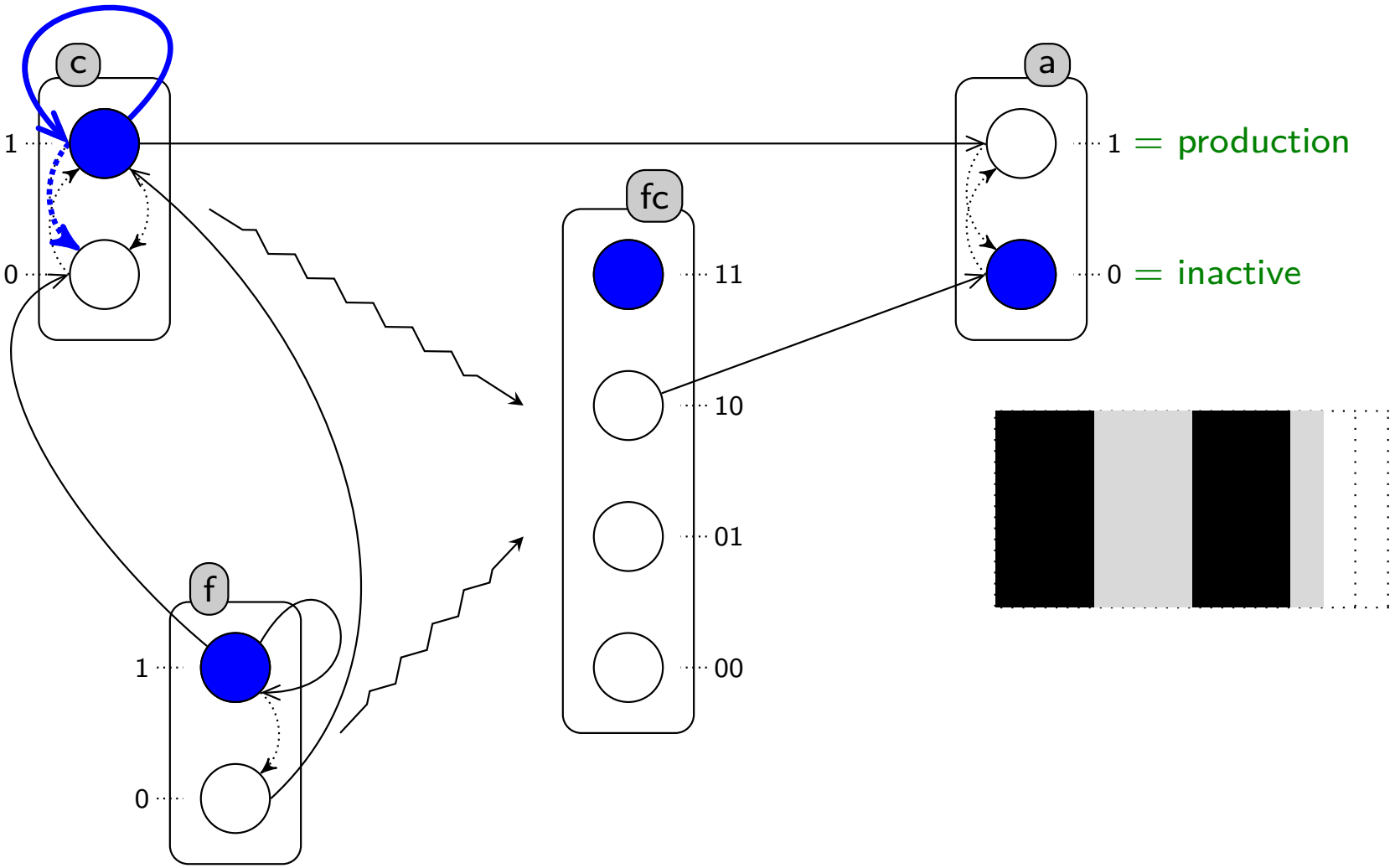
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



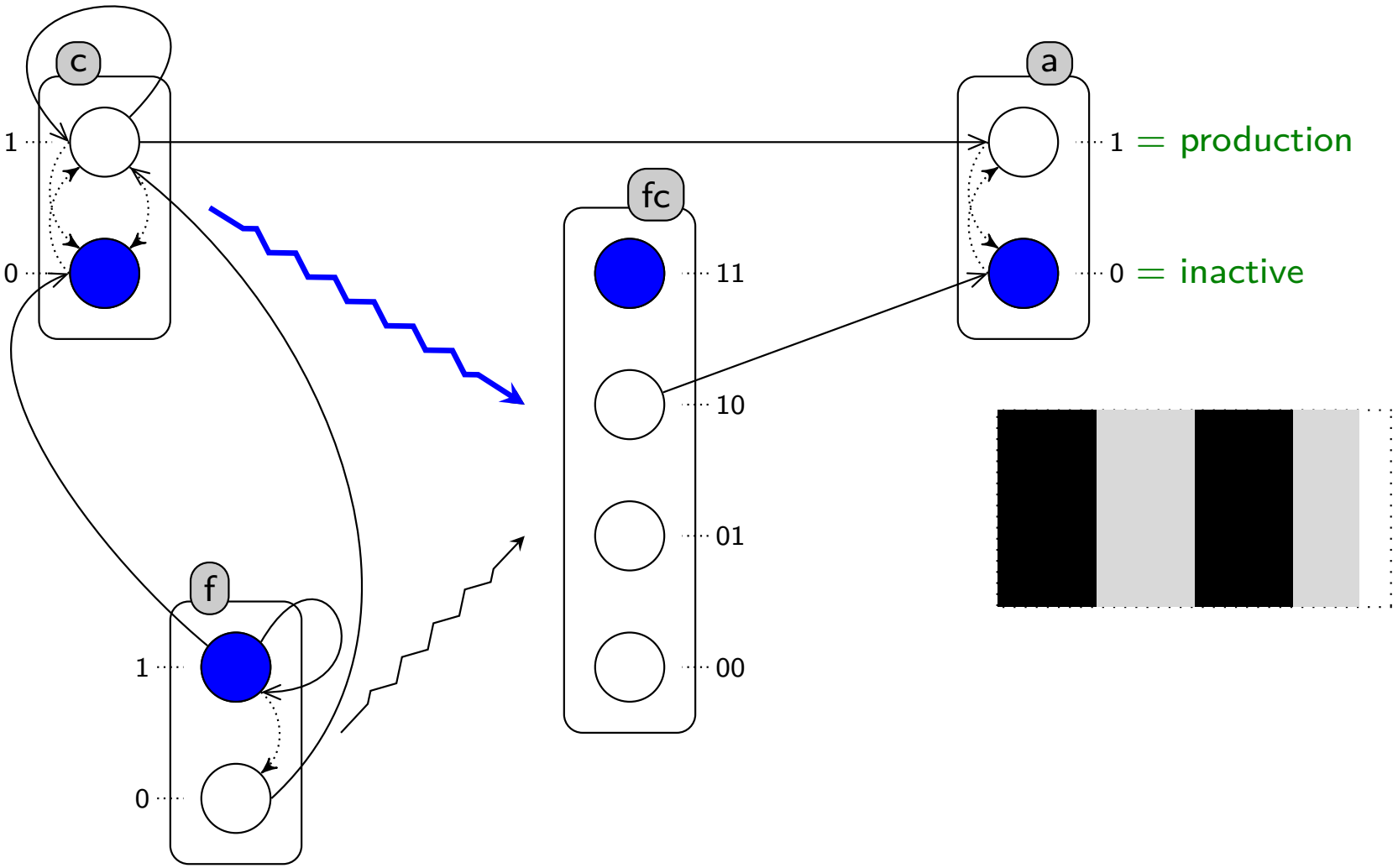
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



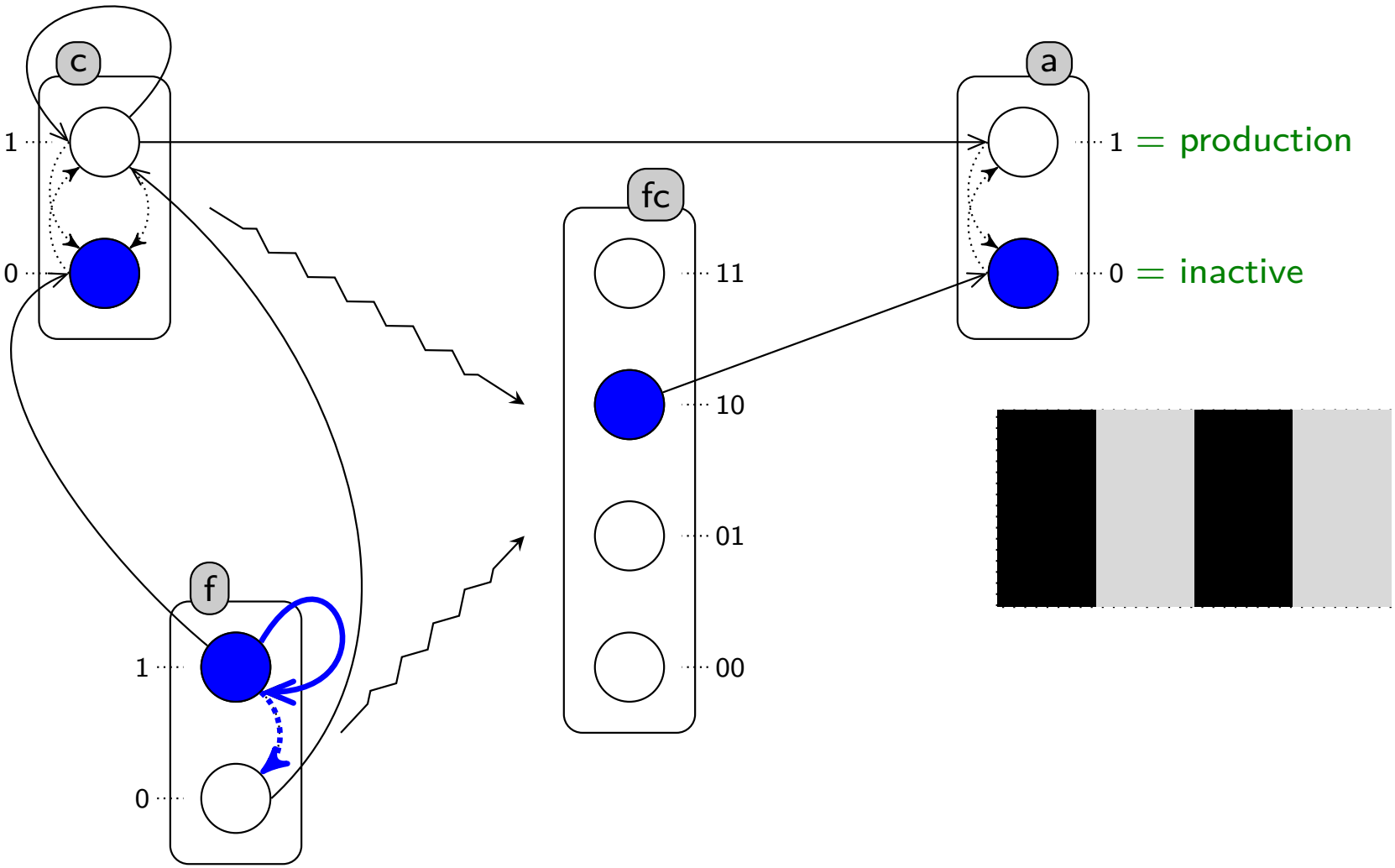
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



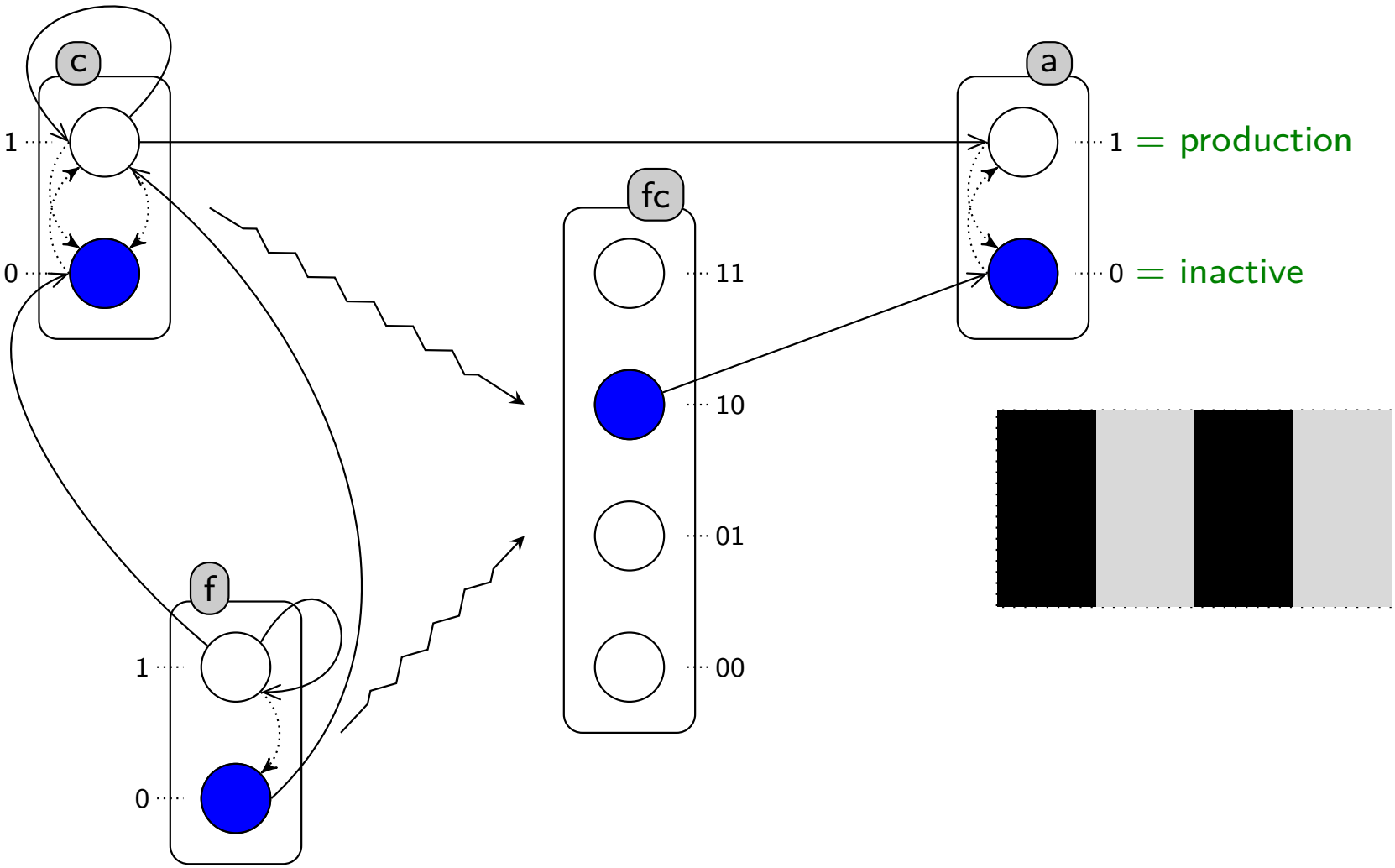
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



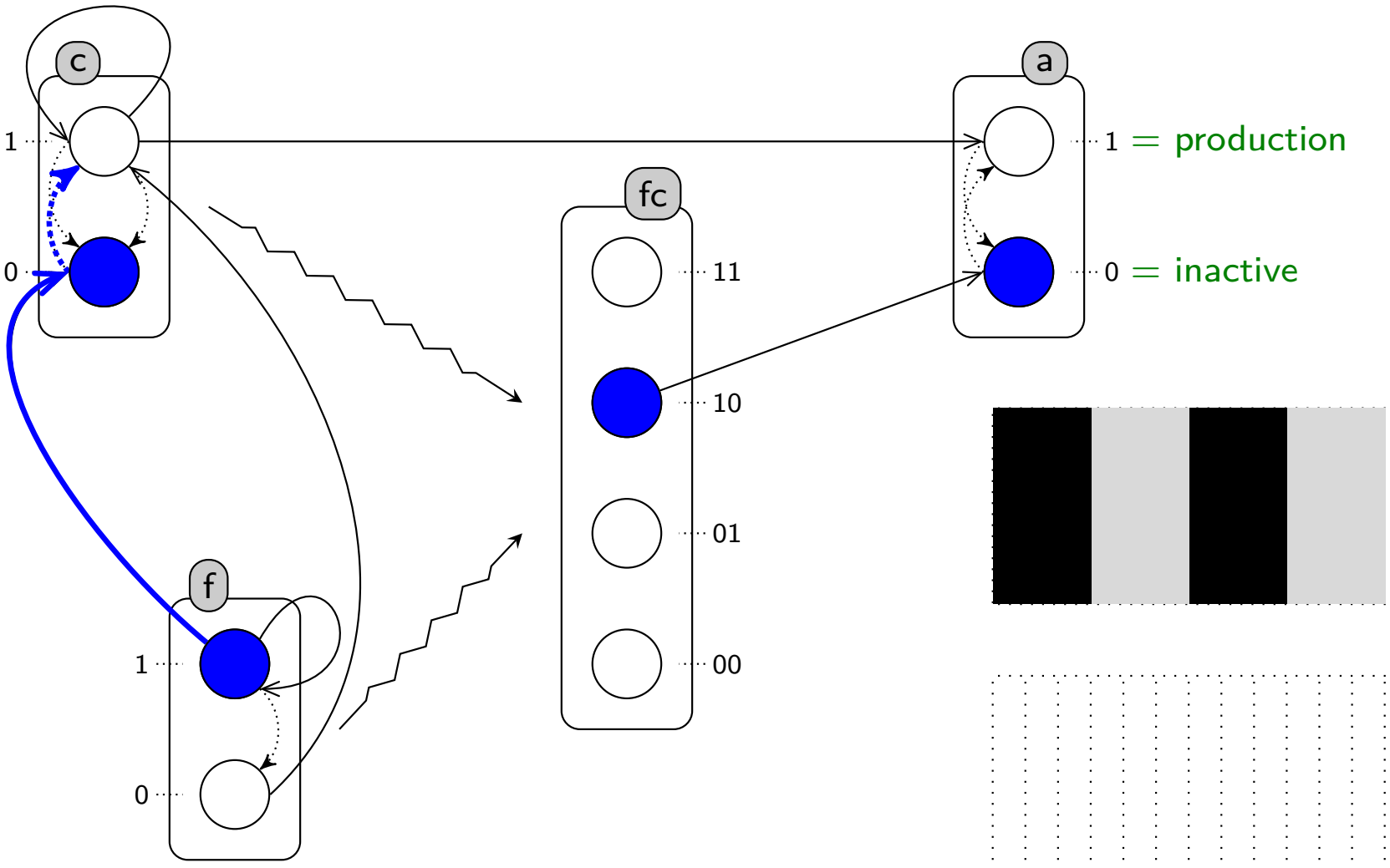
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



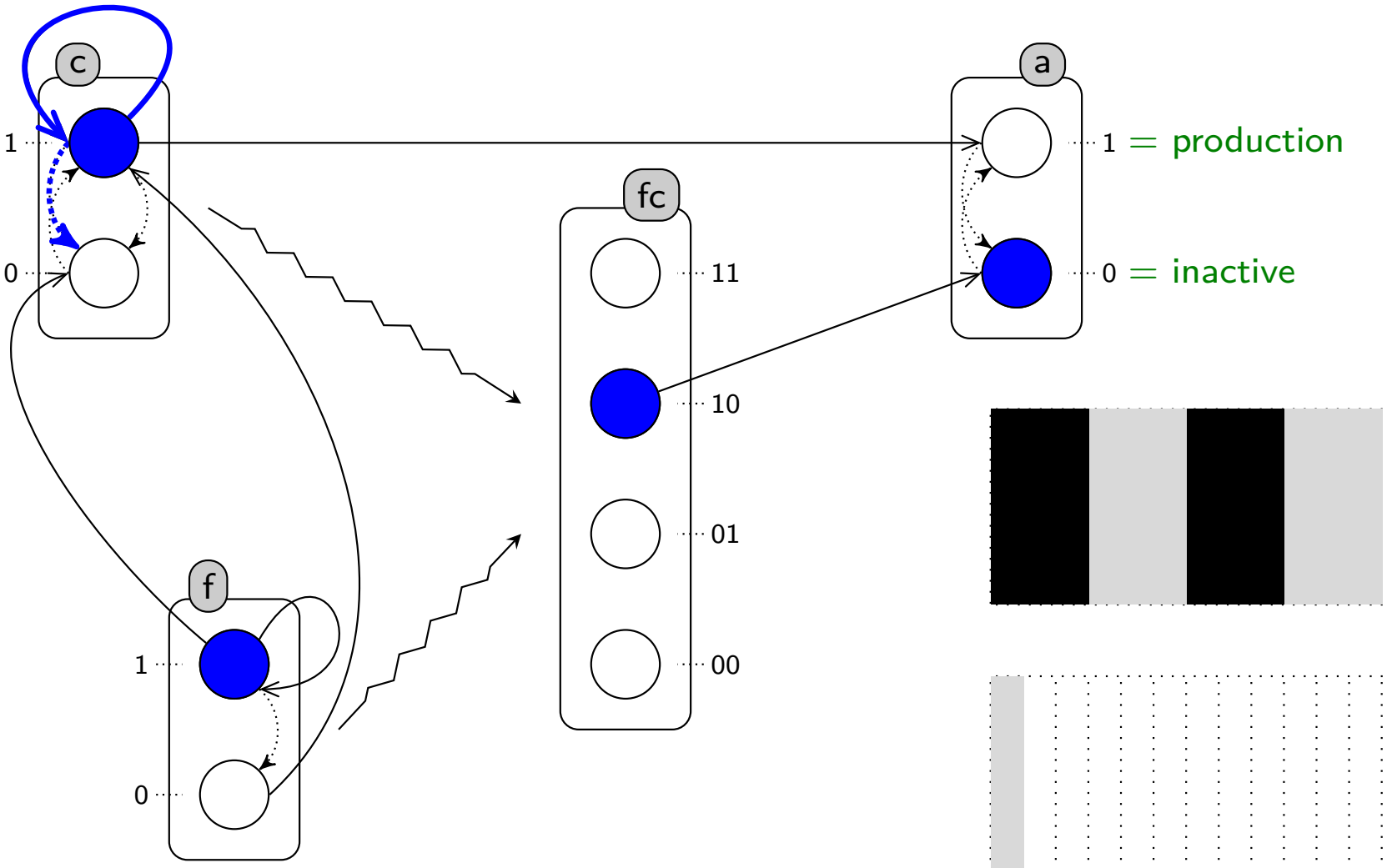
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



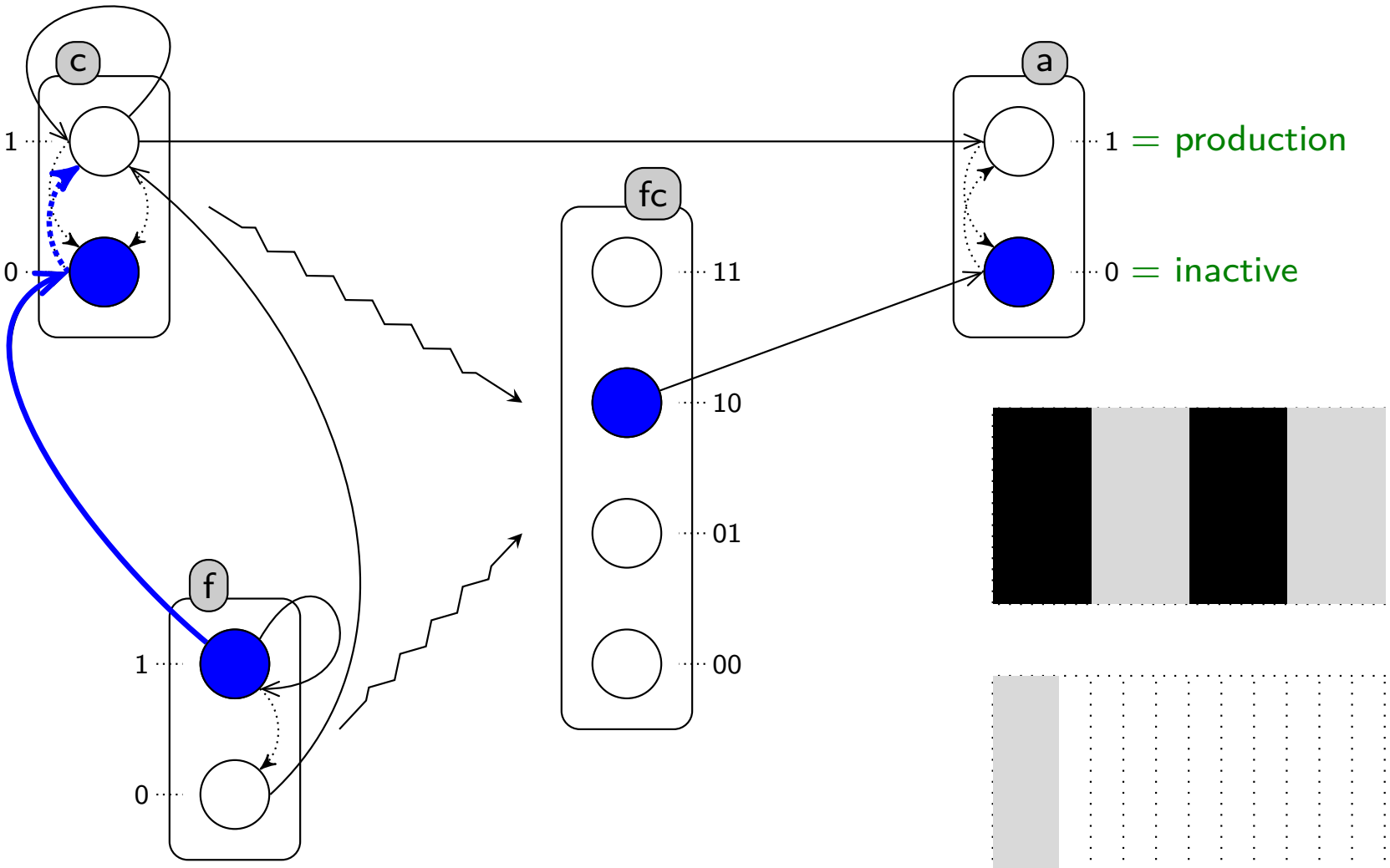
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



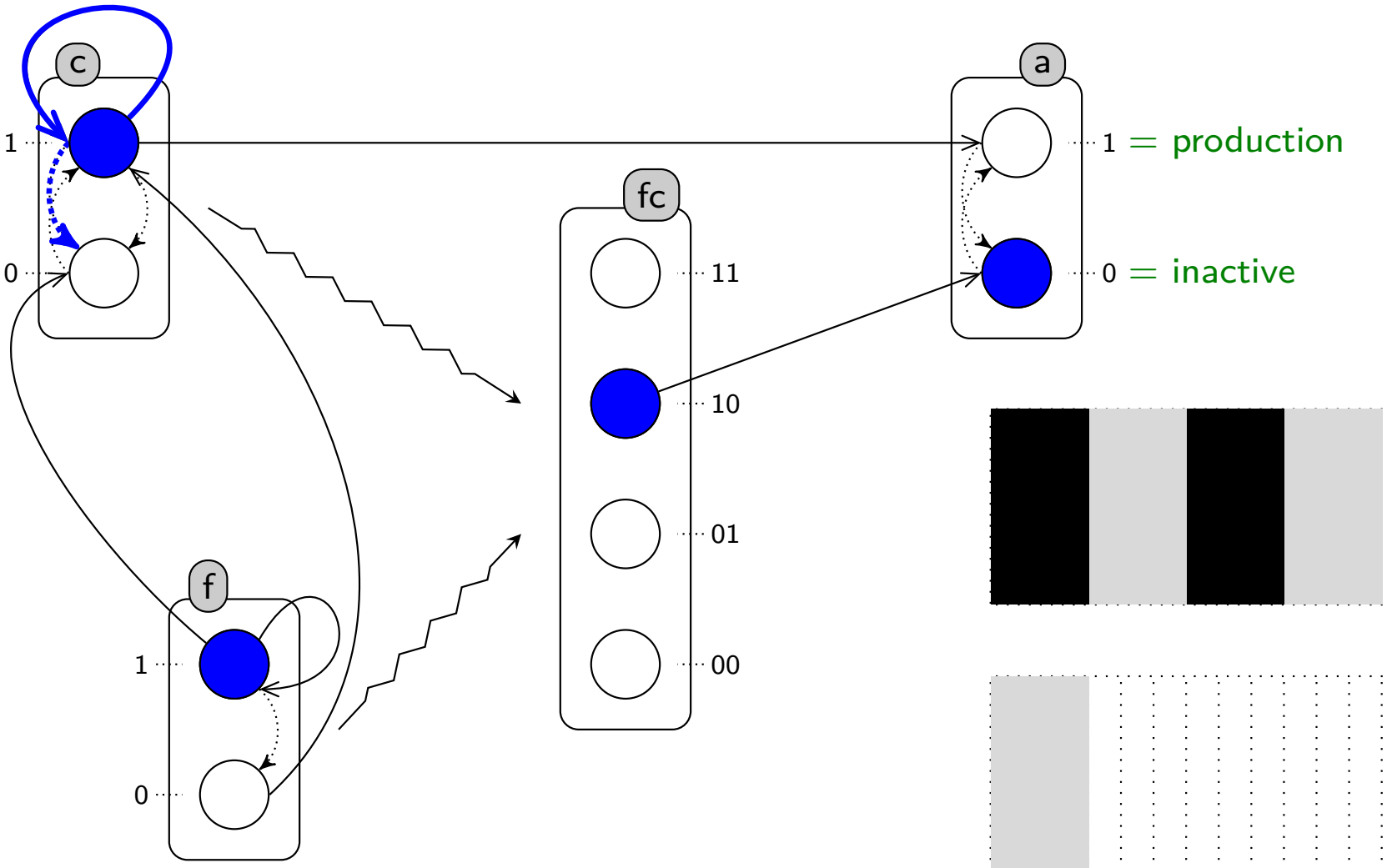
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



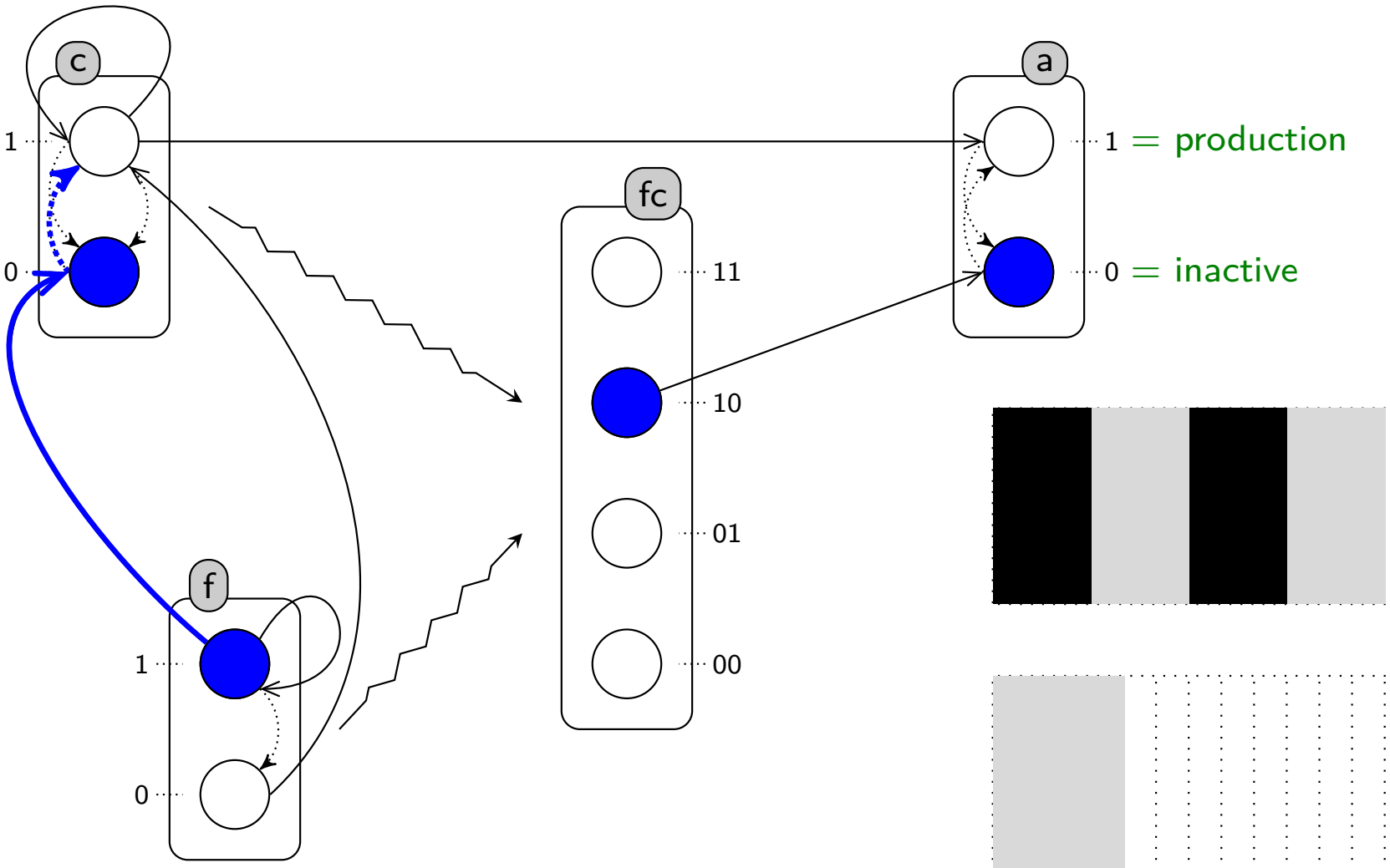
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



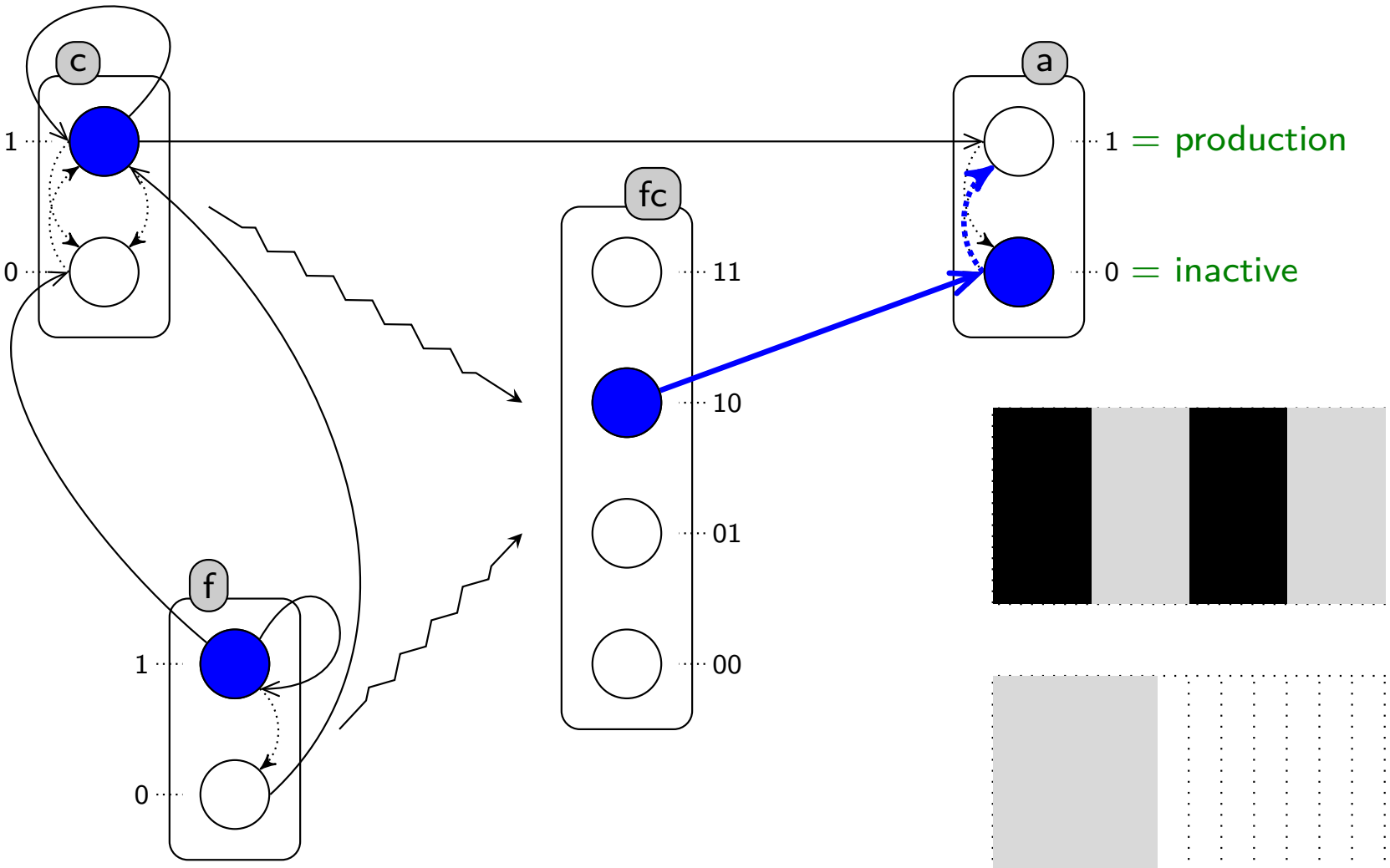
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



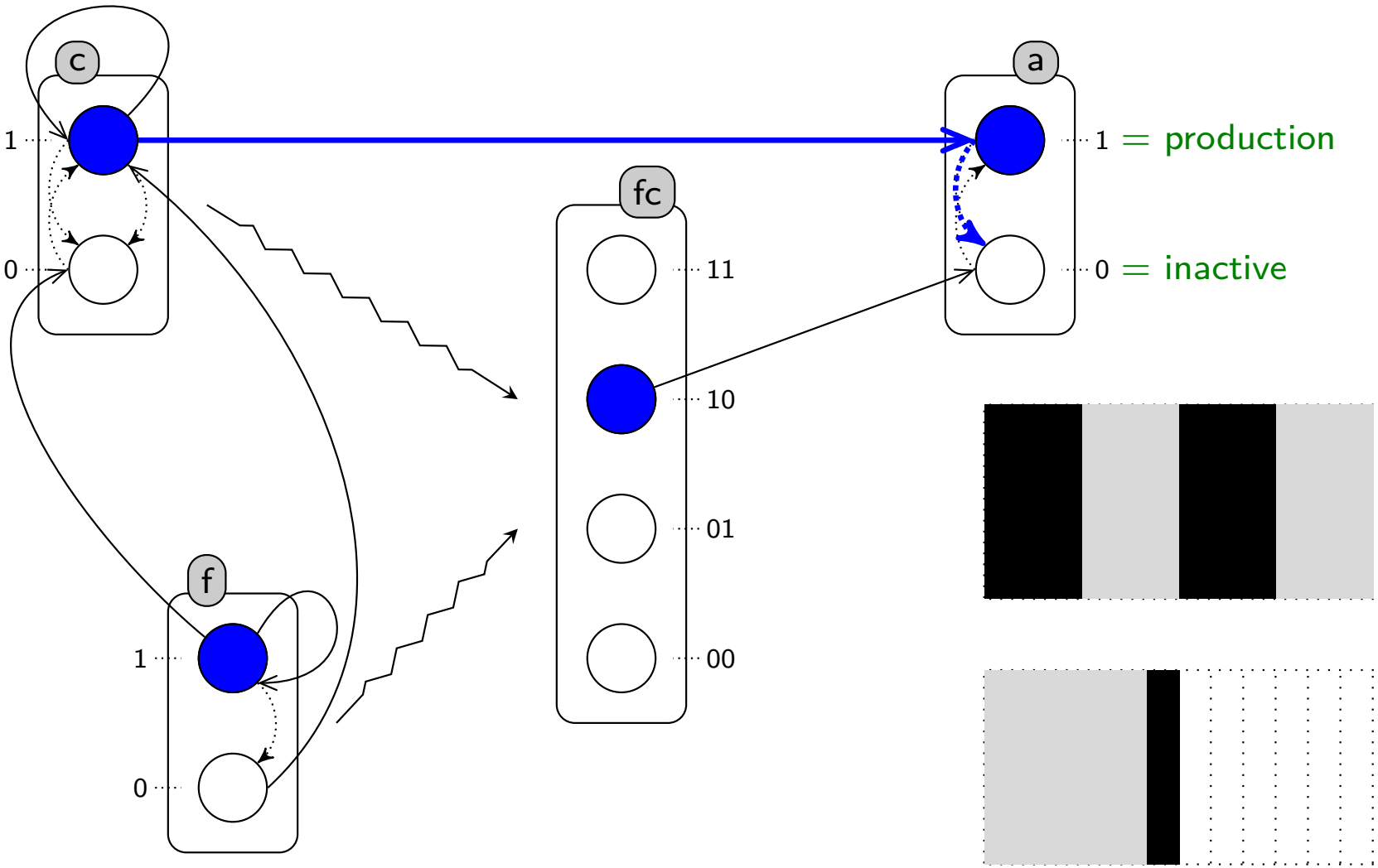
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



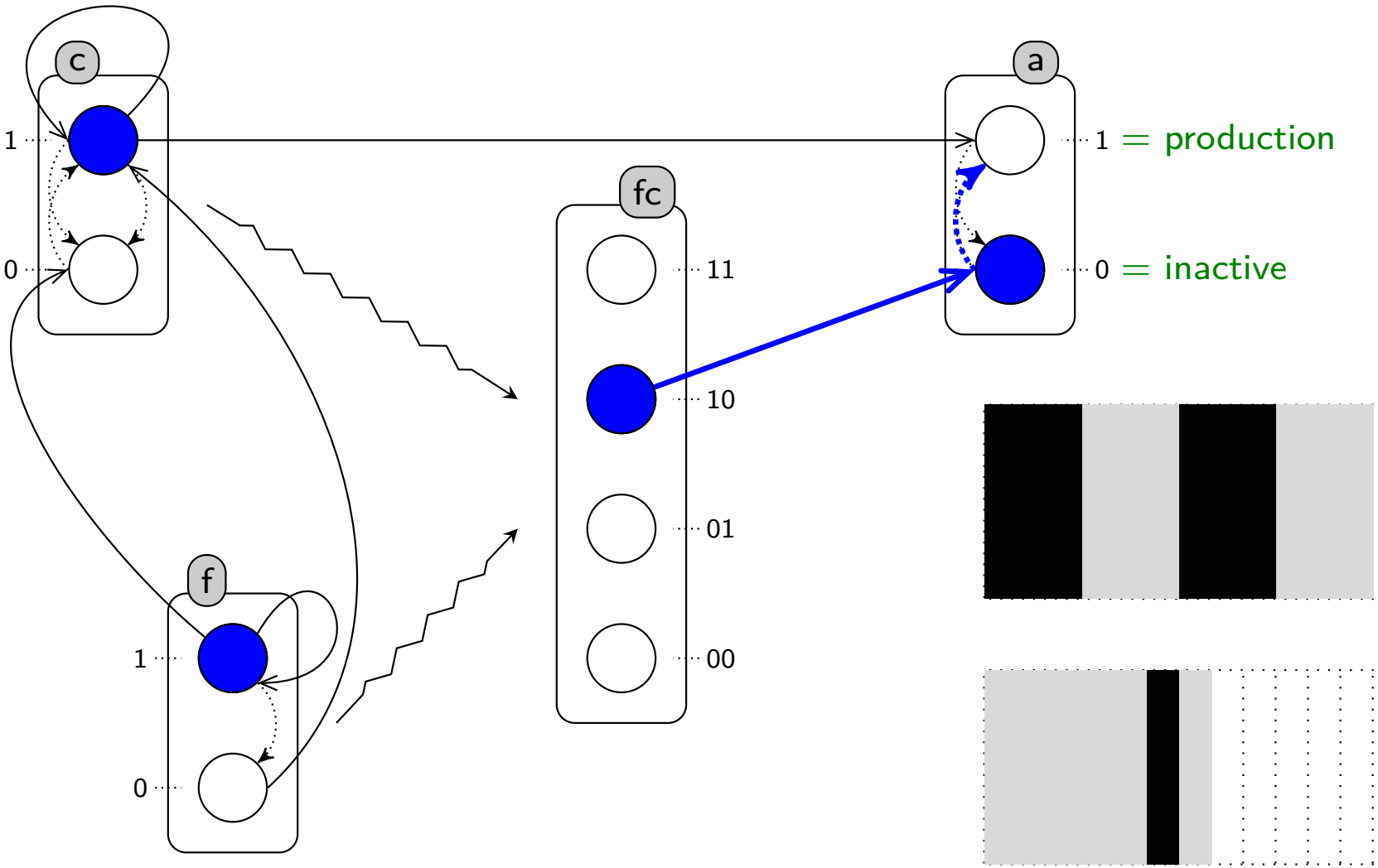
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



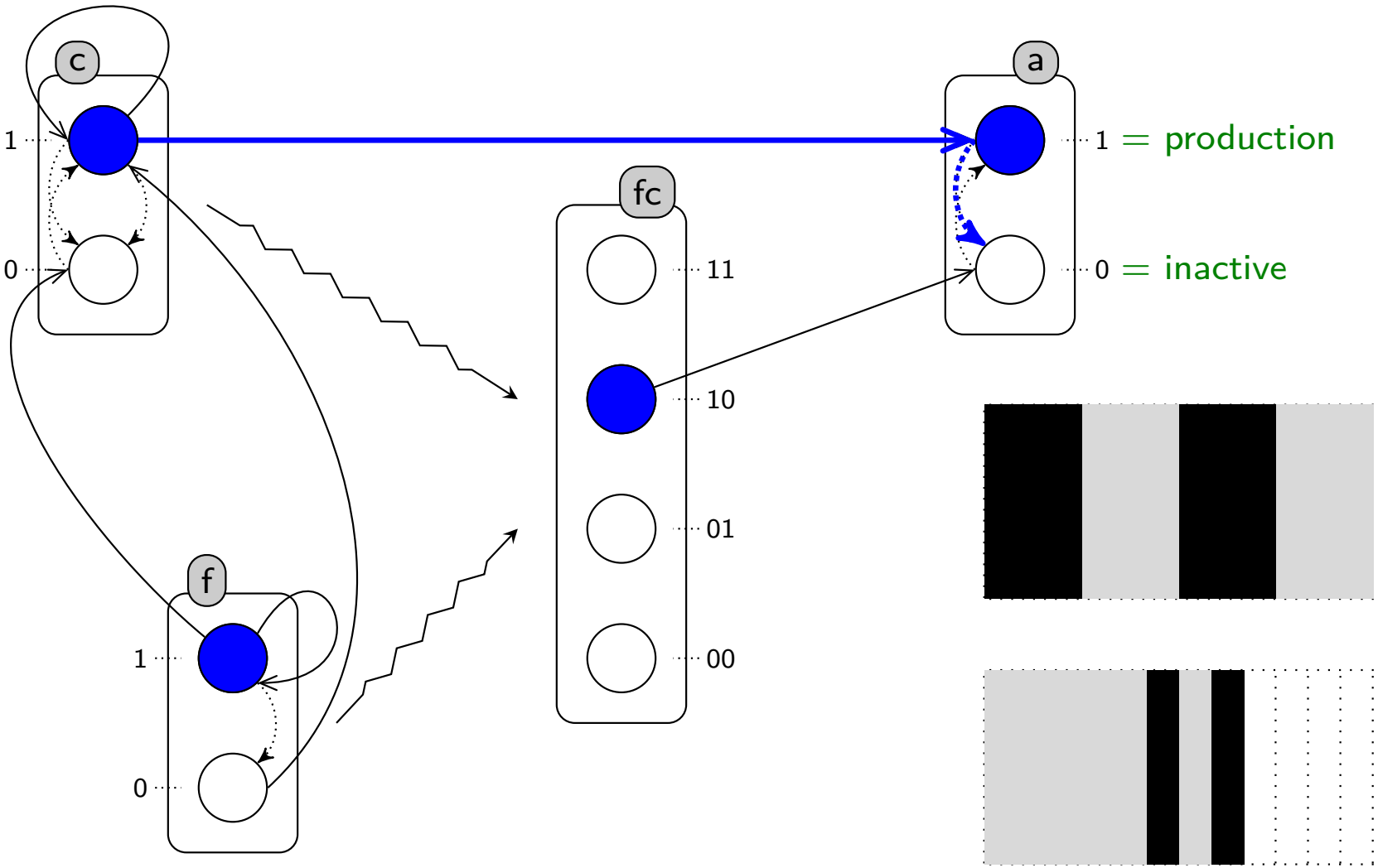
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



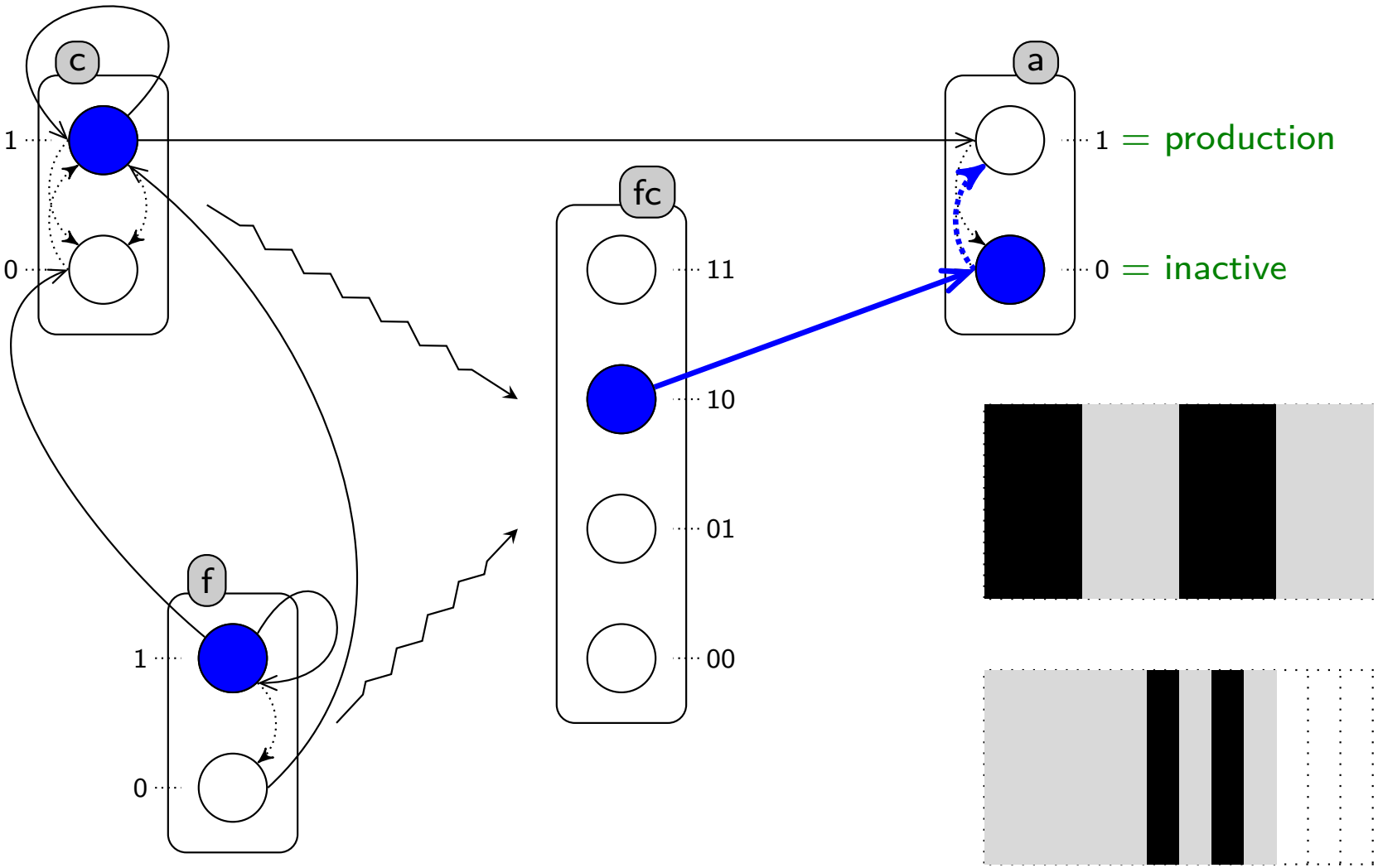
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



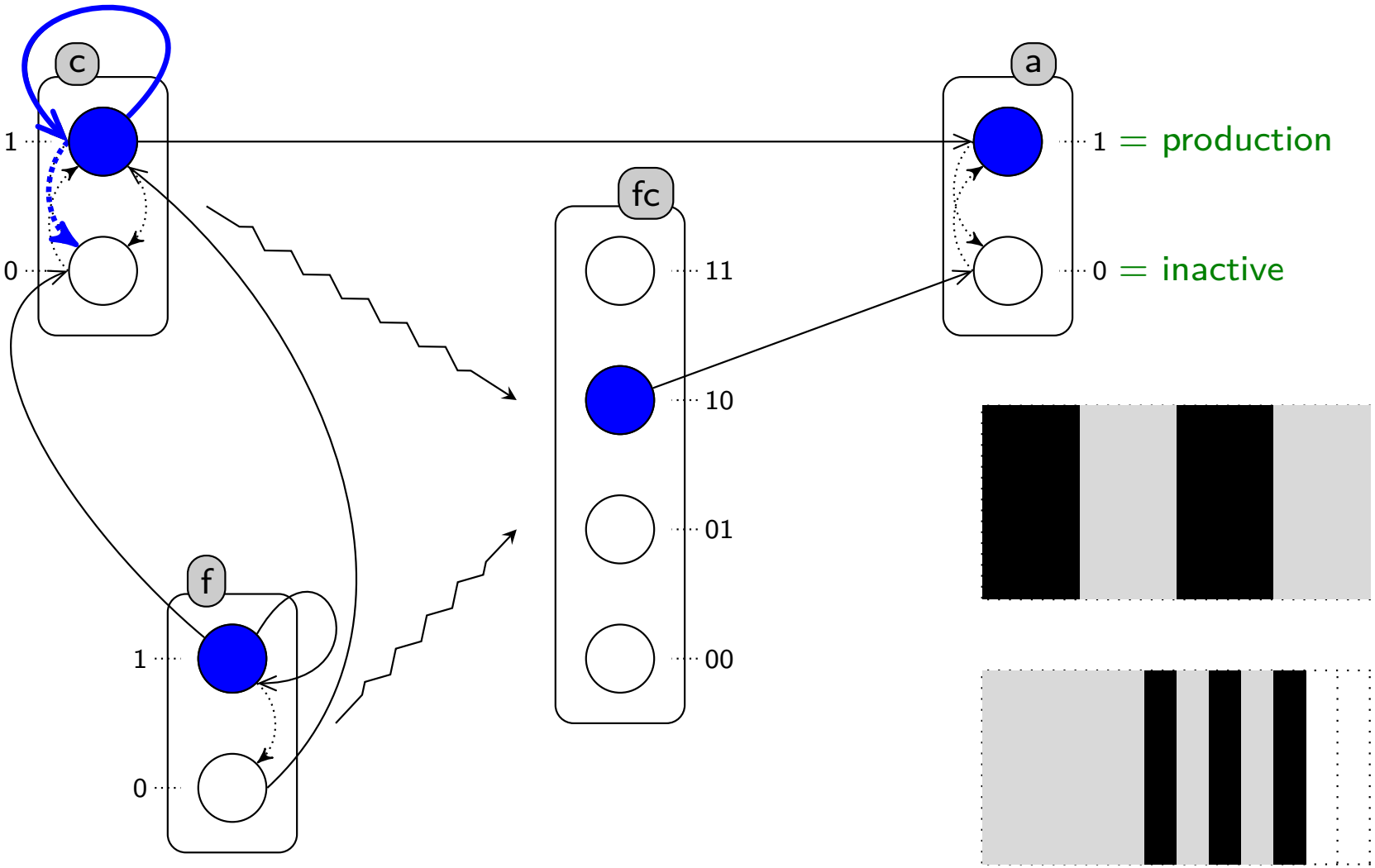
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



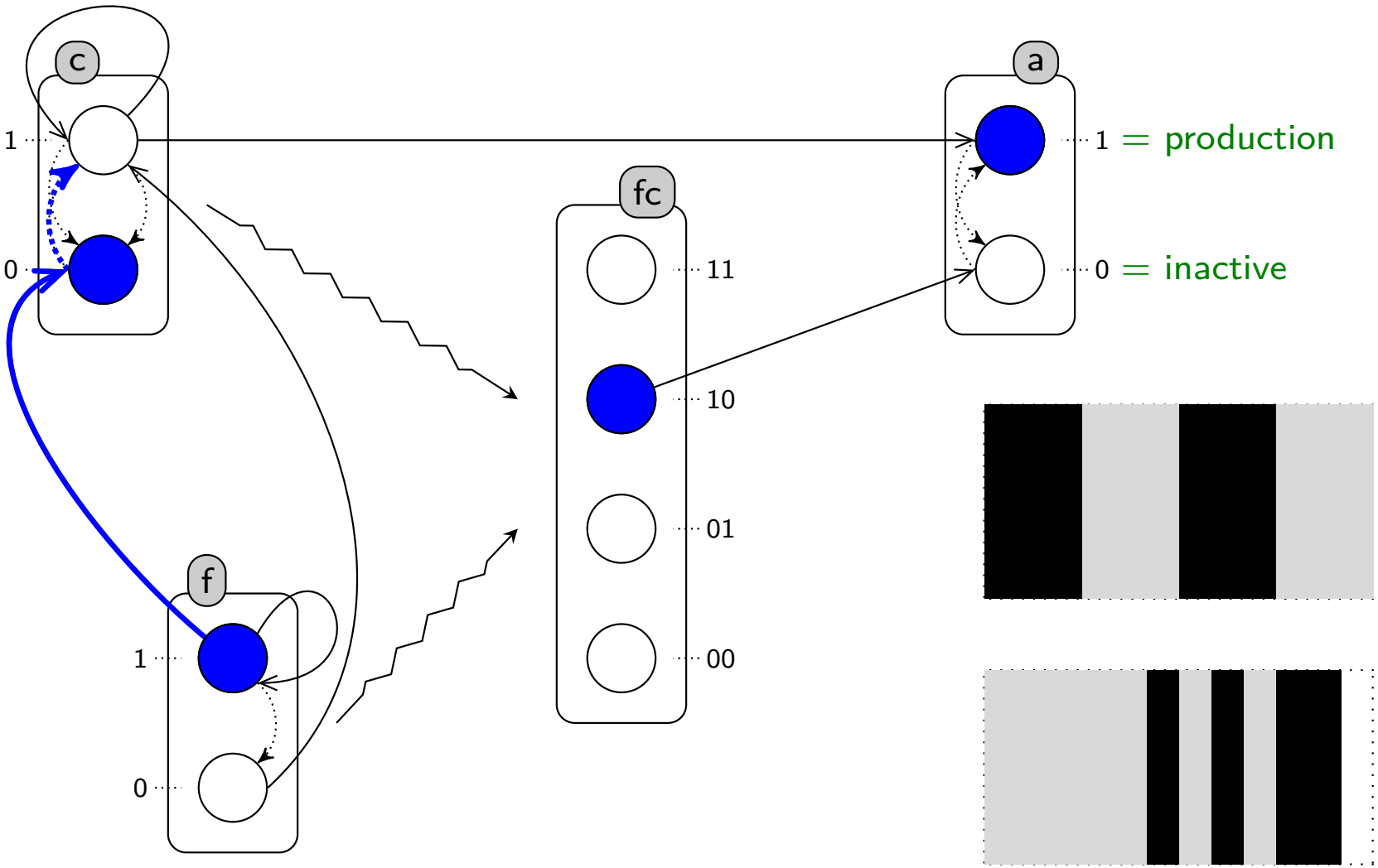
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



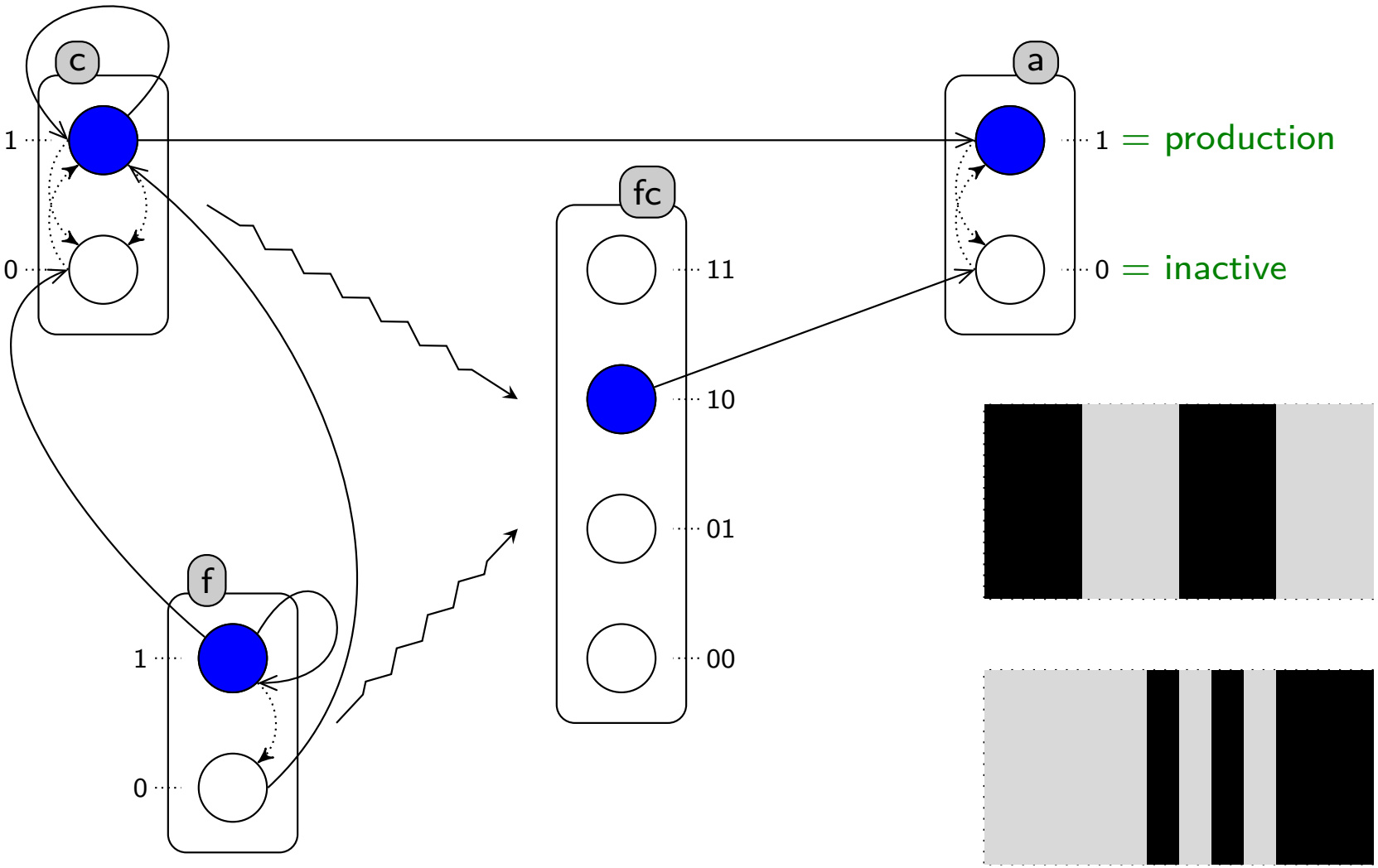
Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]

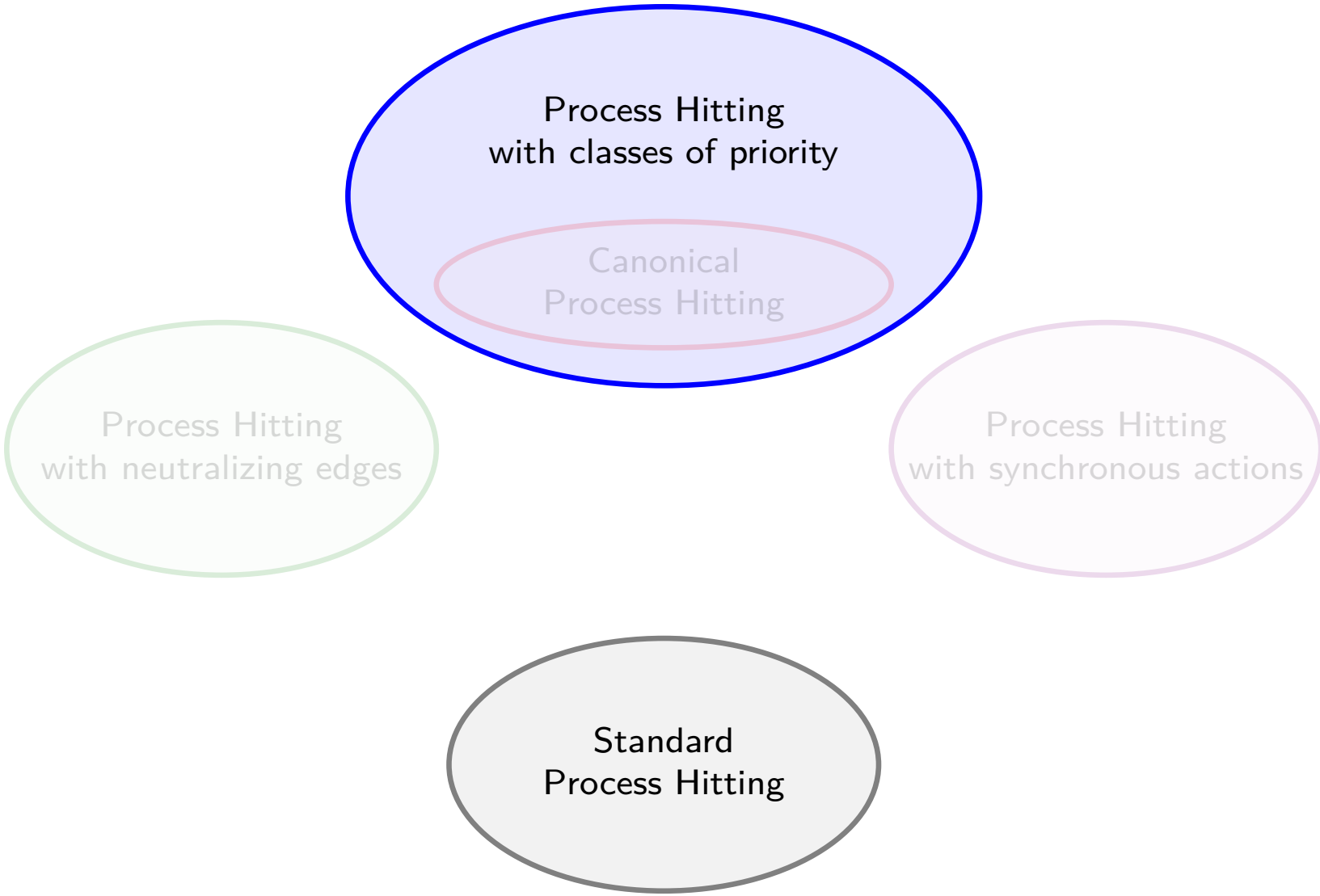


Permissiveness of the Standard Dynamics

Model extracted from [François *et al.* in Molecular Systems Biology, 2007]



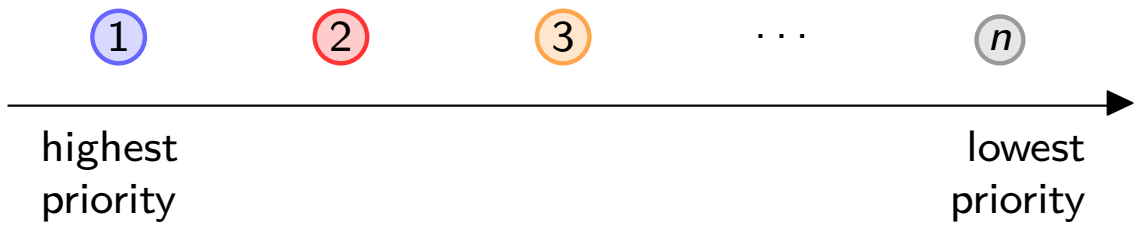
Process Hitting with Classes of Priorities

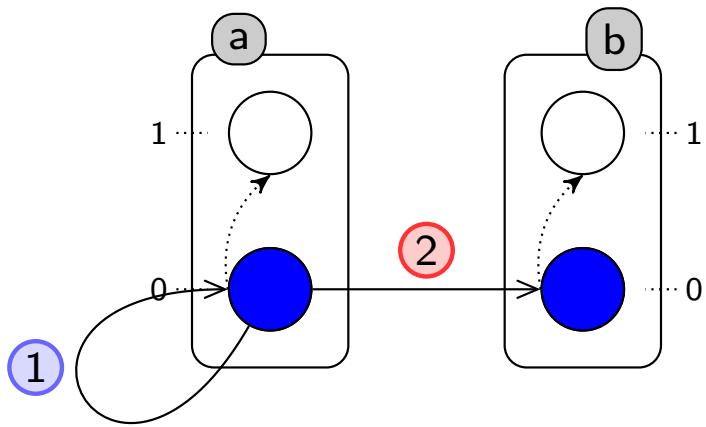


Addition of classes of priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable



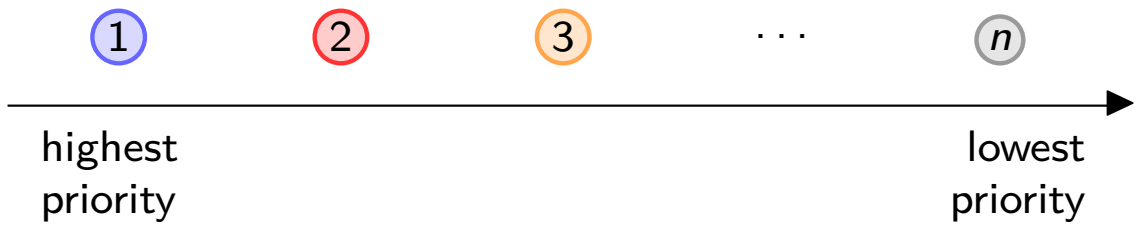


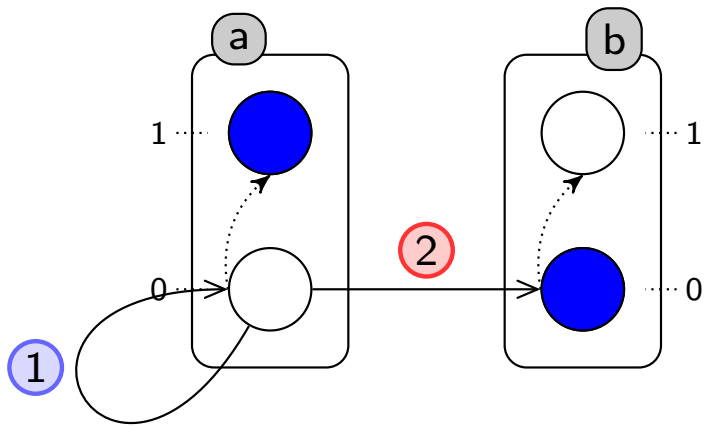
$\rightarrow$ b_1 **cannot** be reached

Addition of classes of priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable

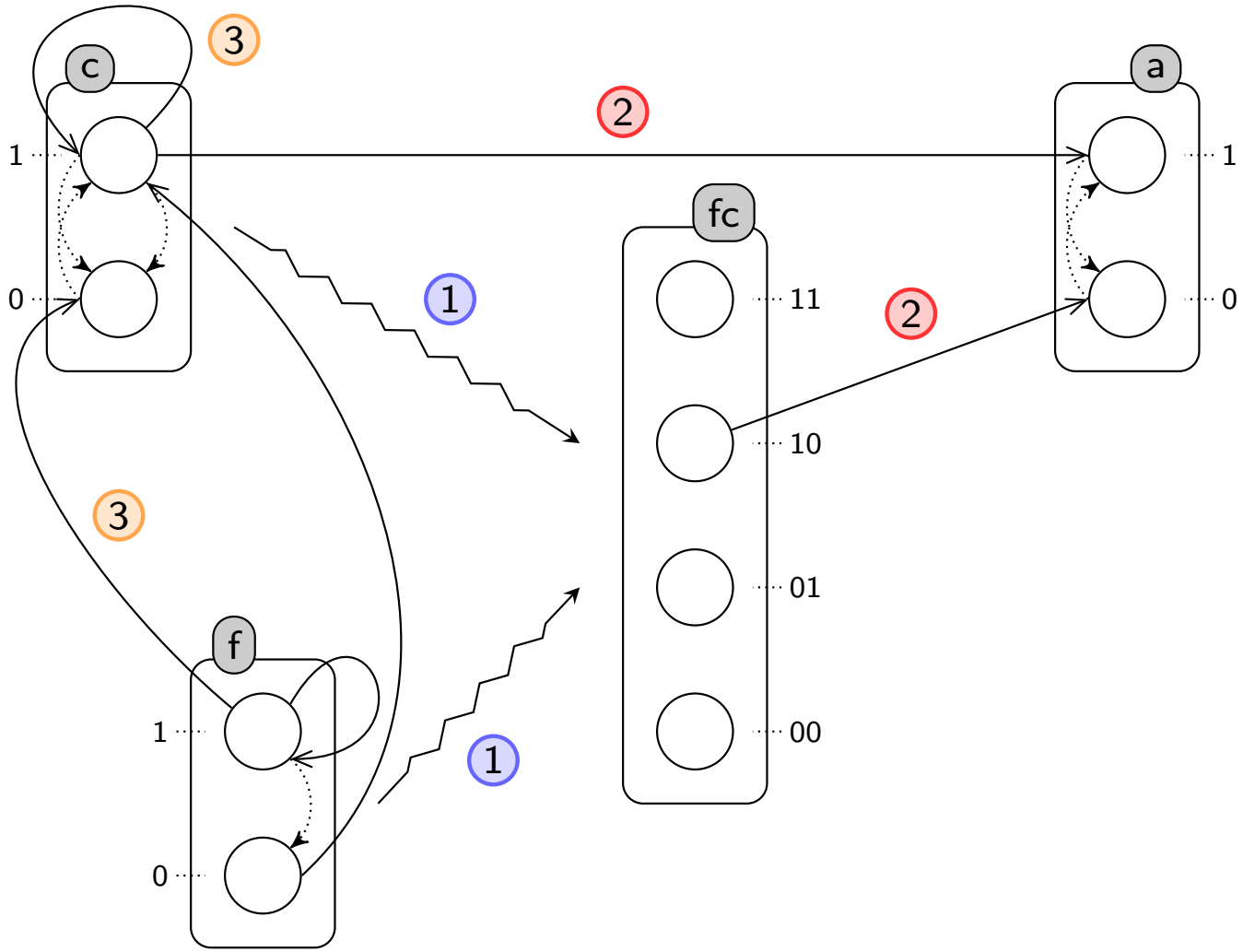




$\rightarrow$ b_1 **cannot** be reached

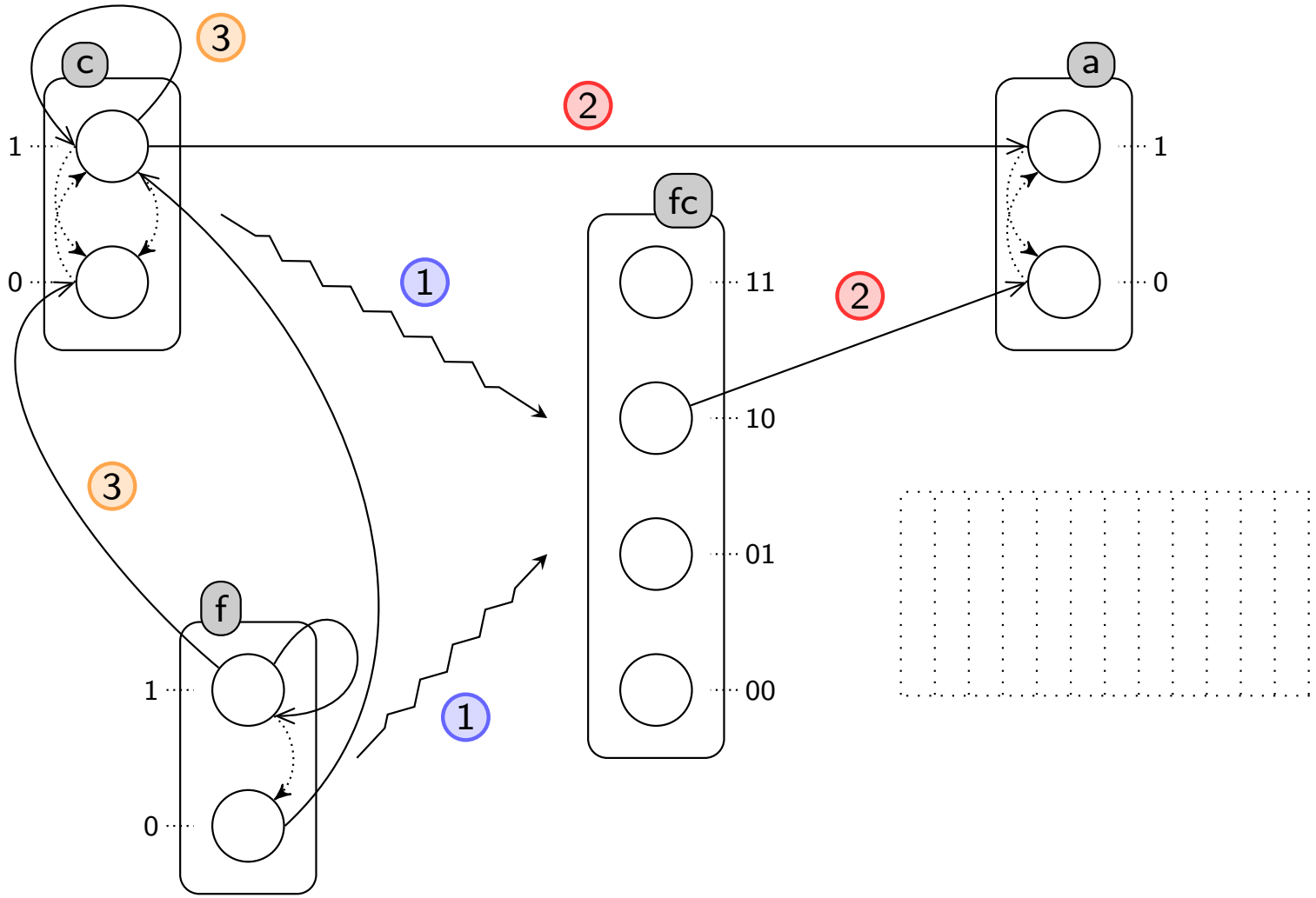
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



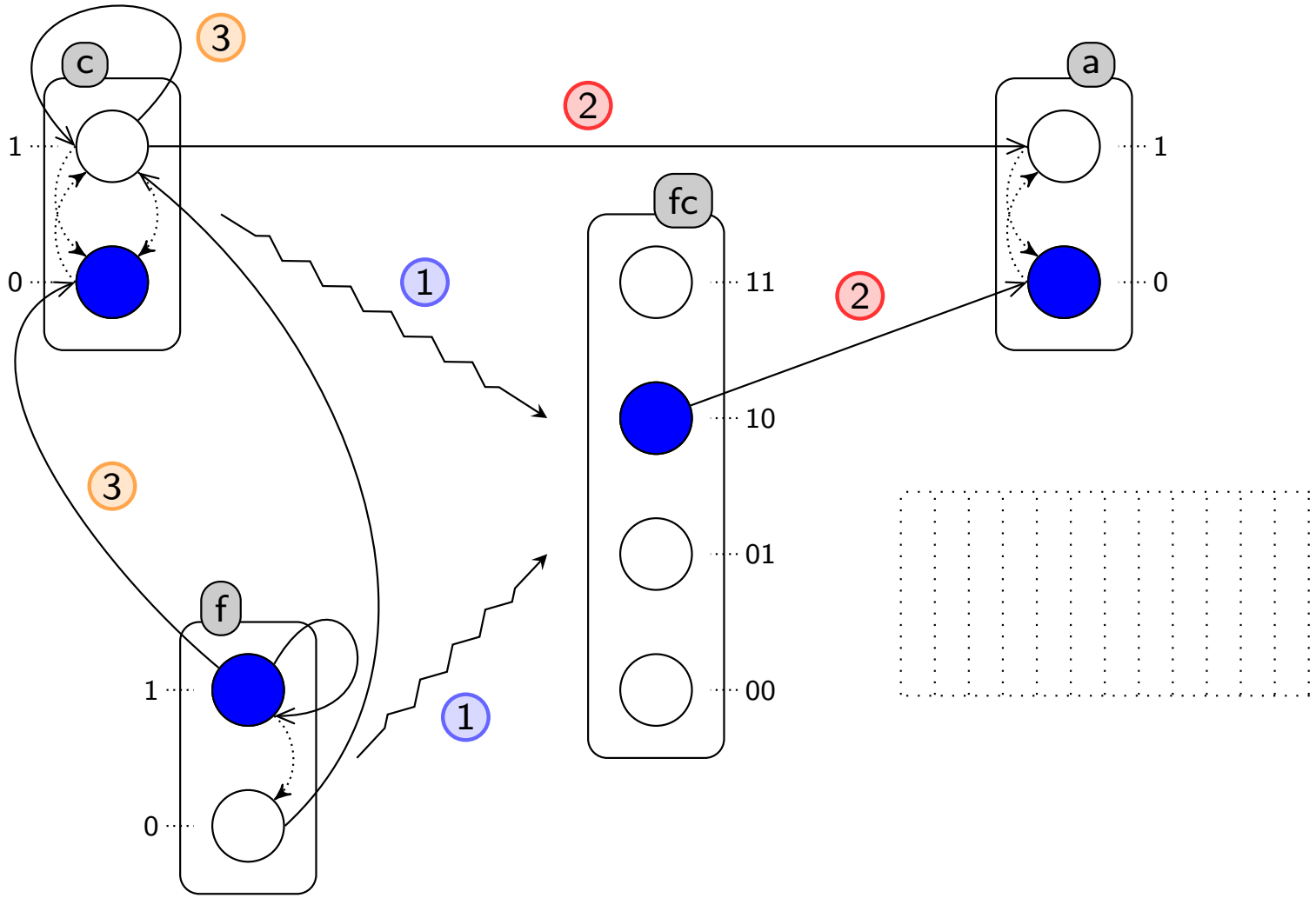
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



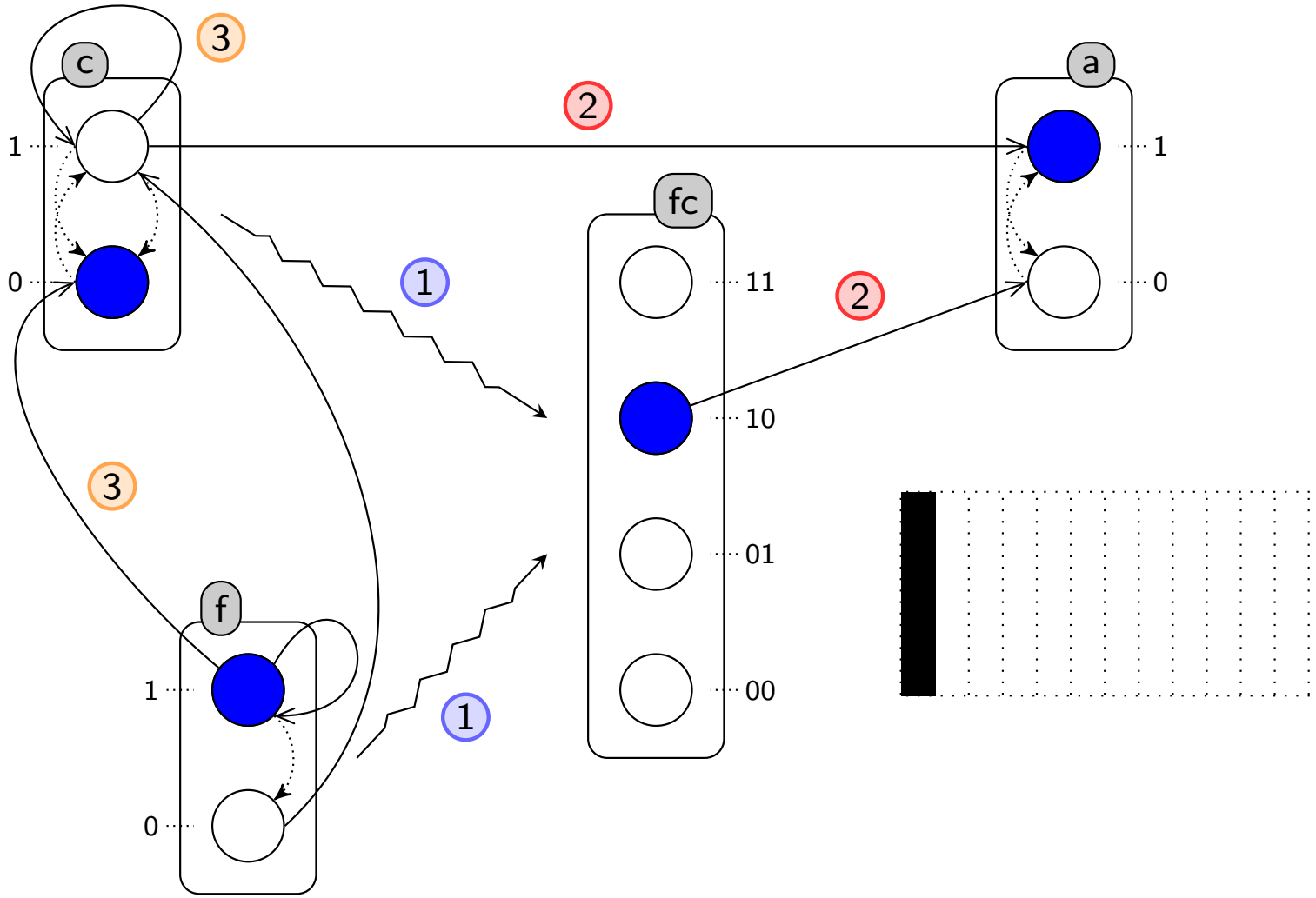
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



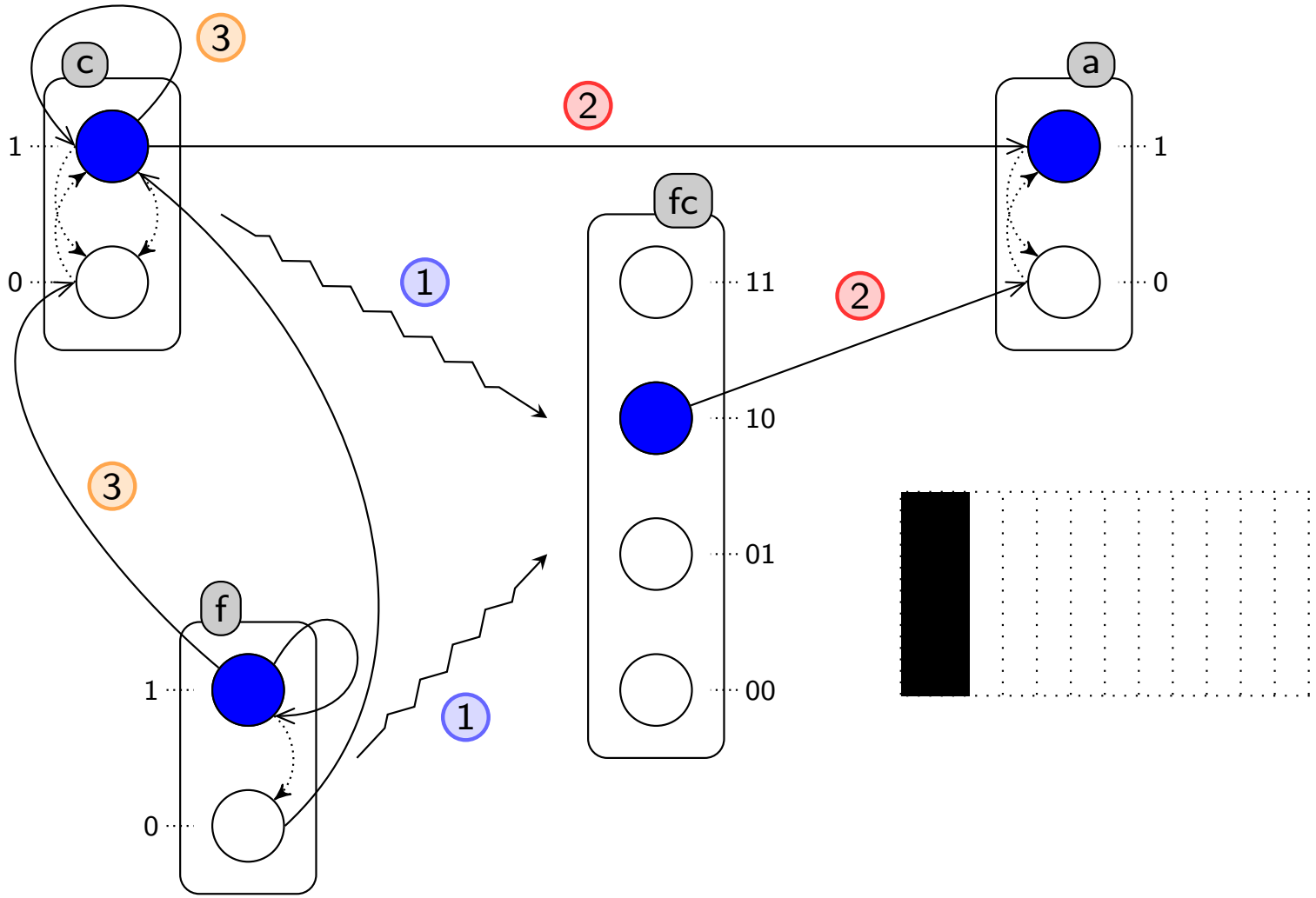
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



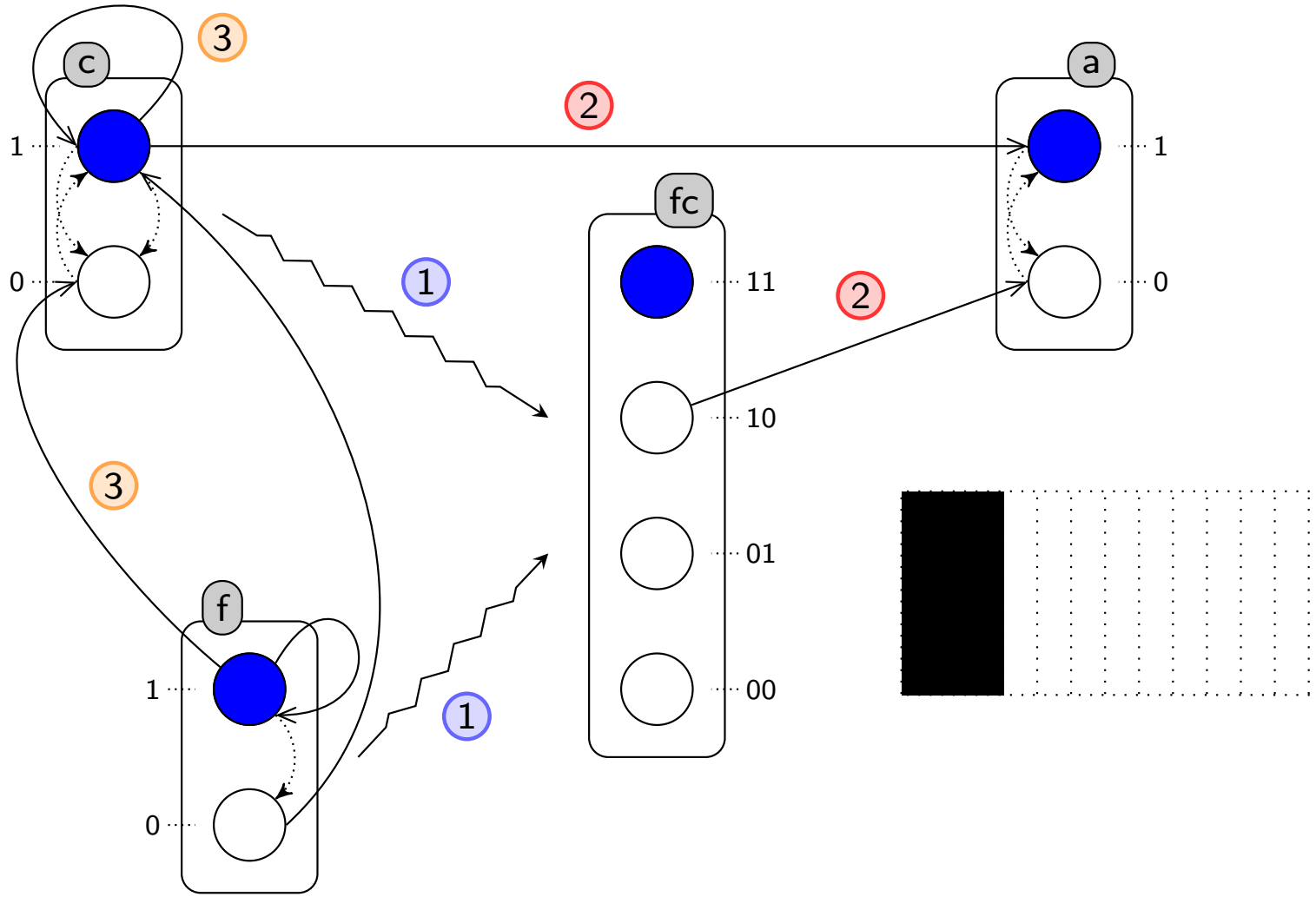
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



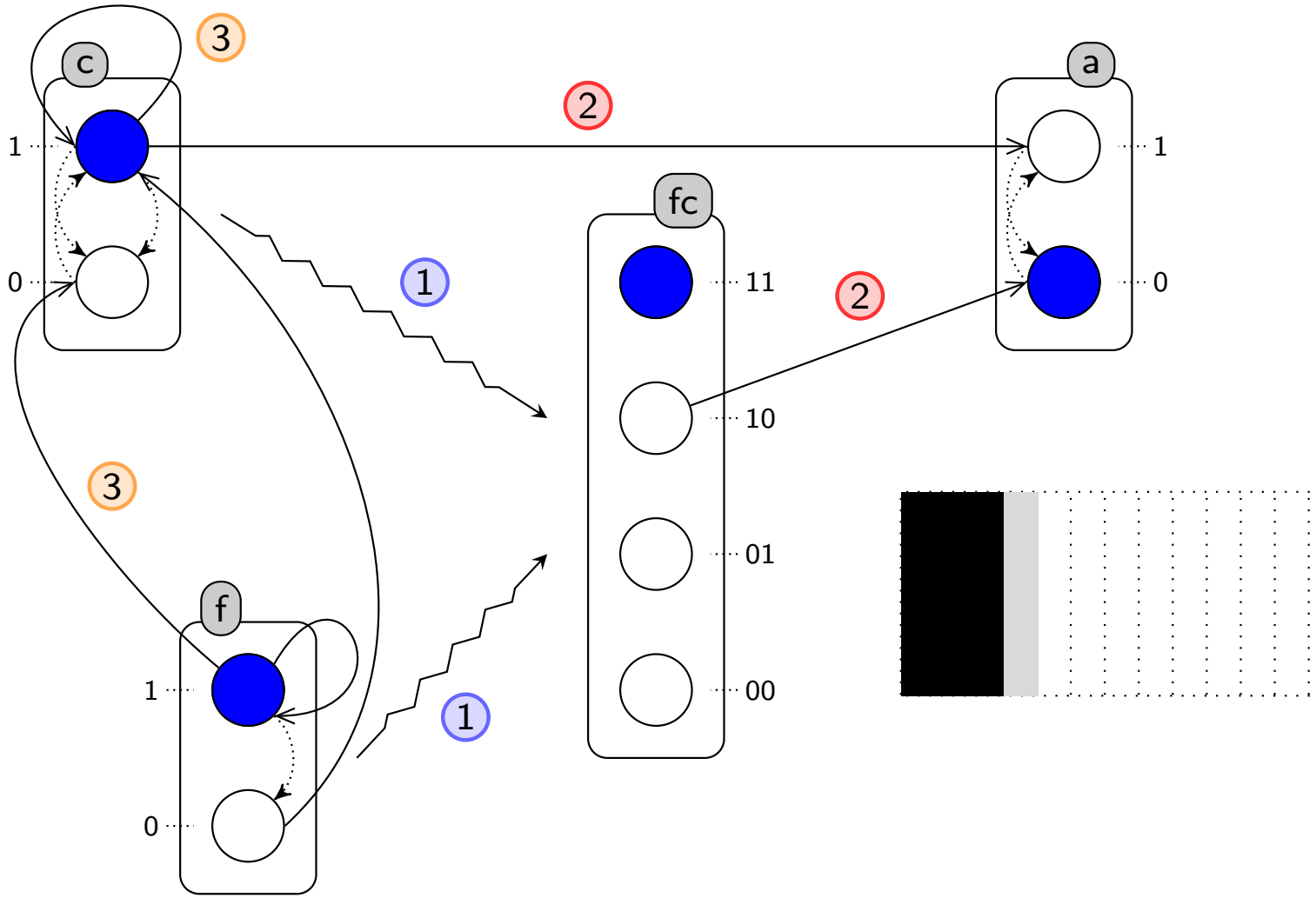
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



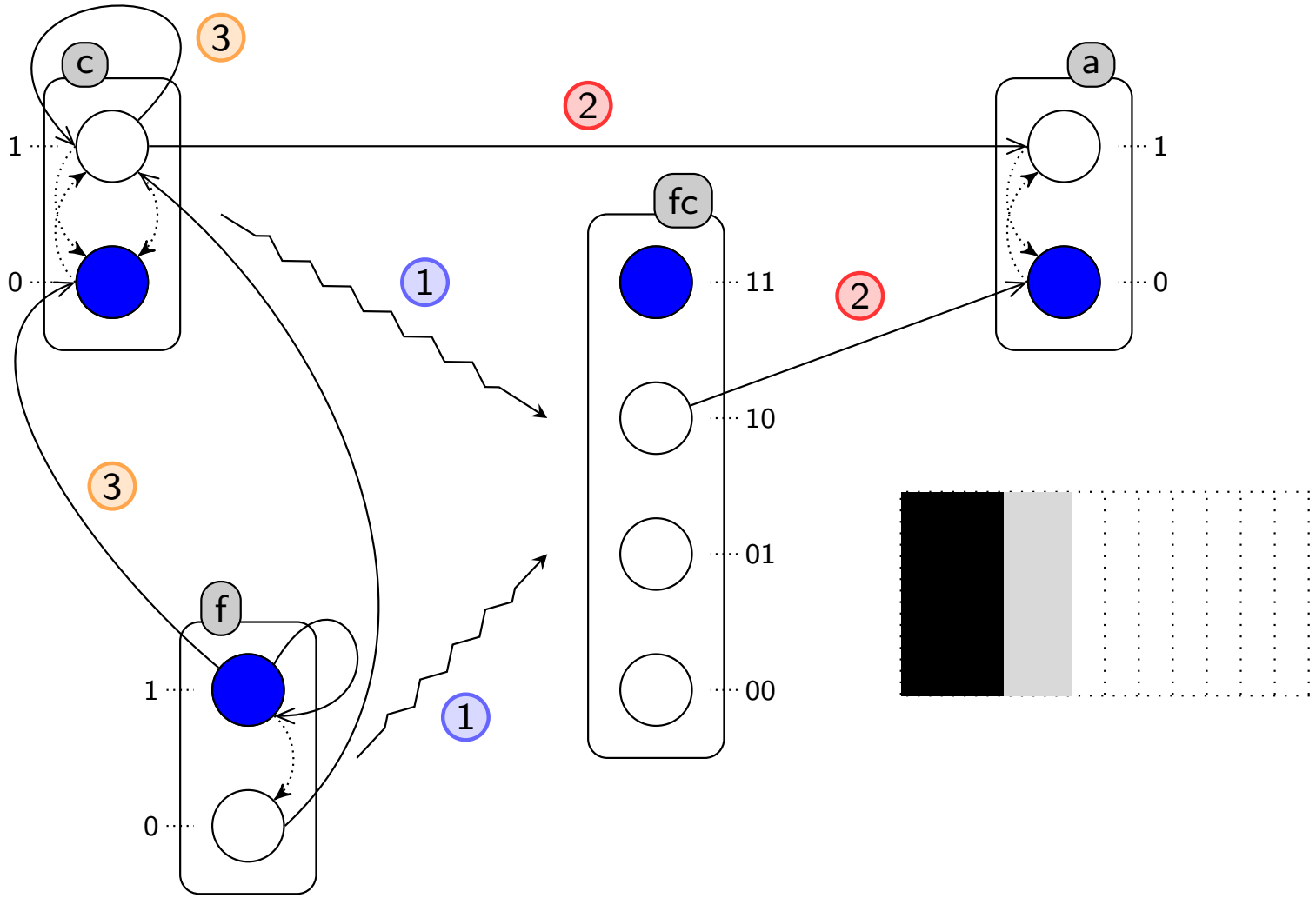
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



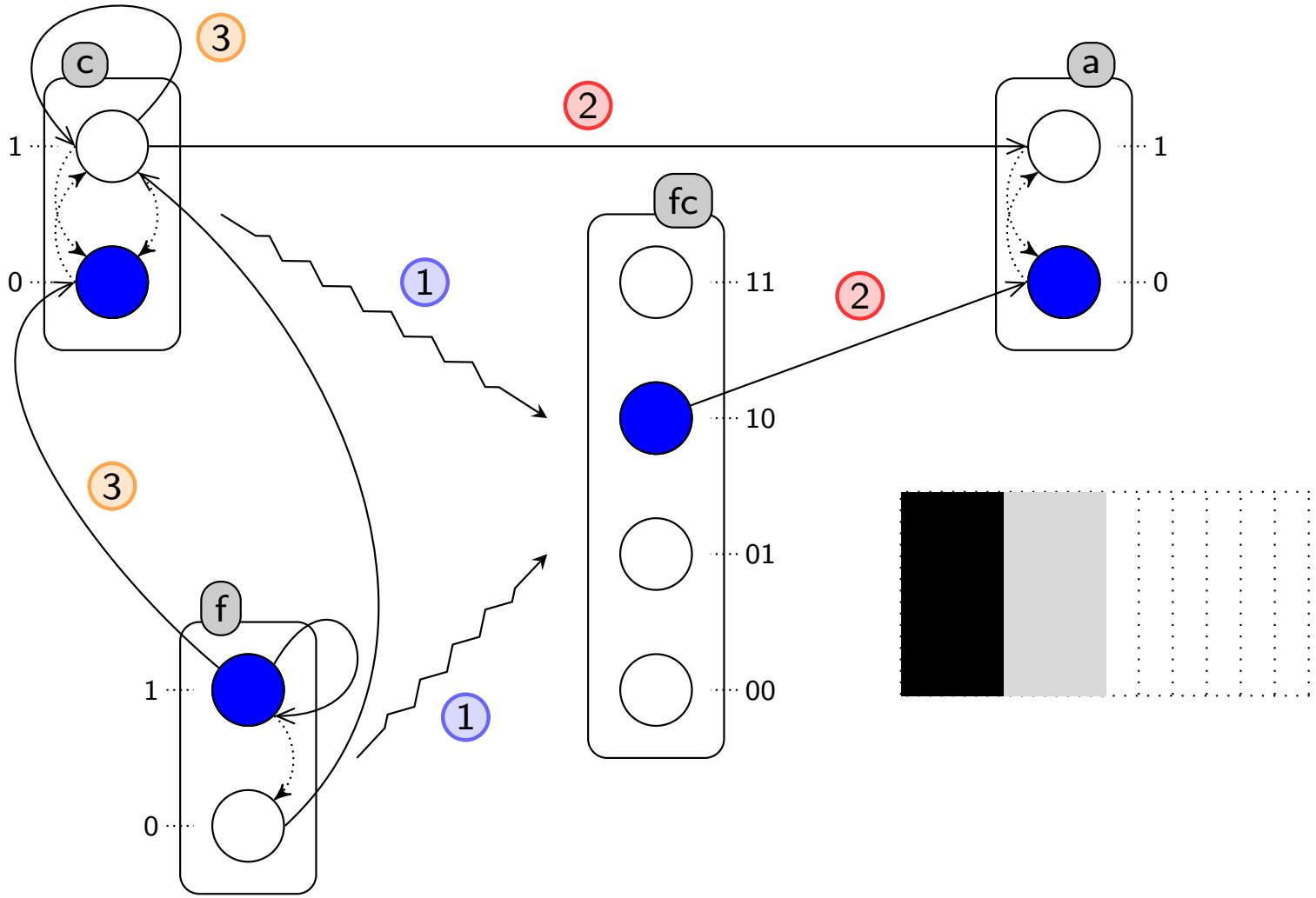
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



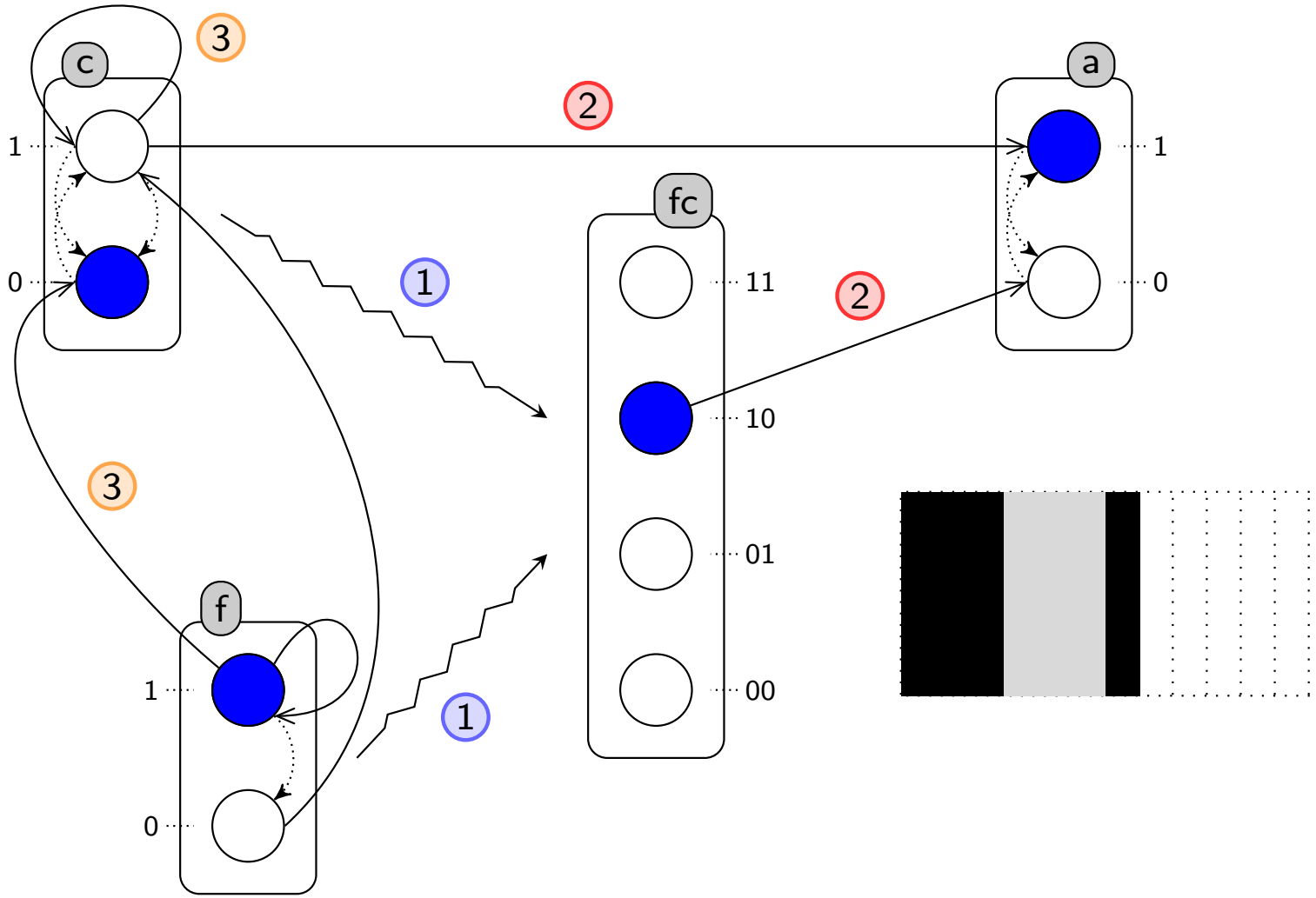
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



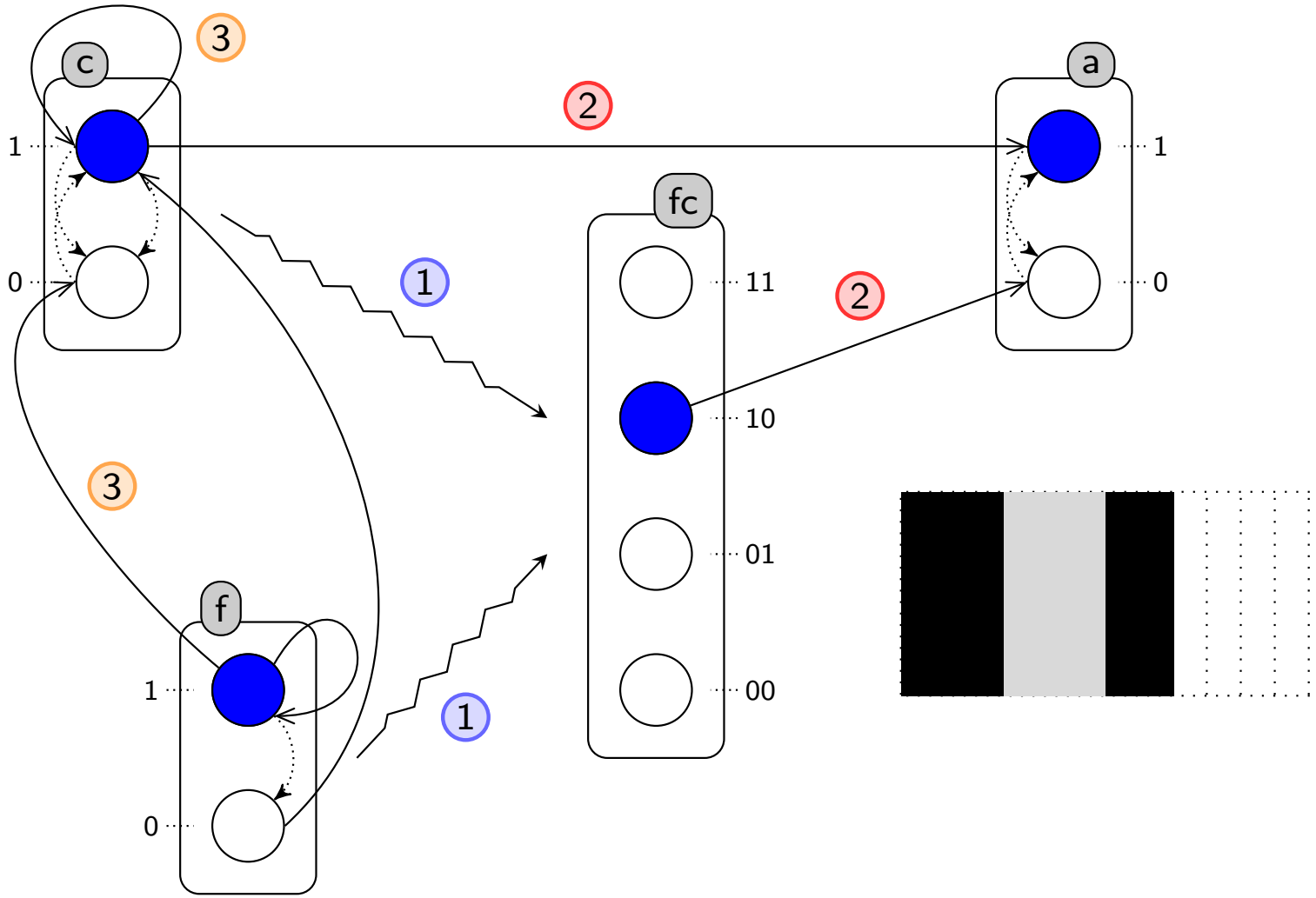
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



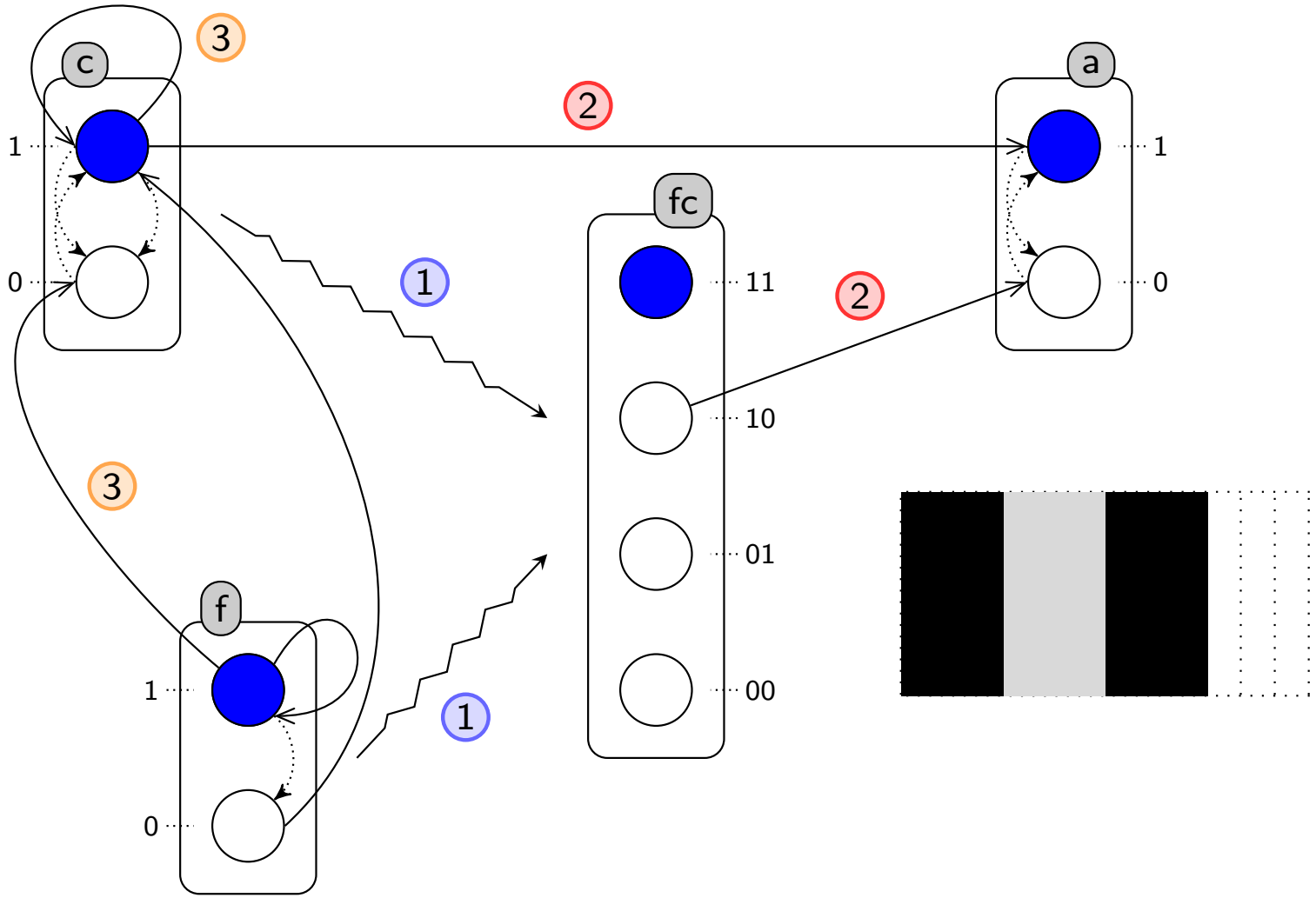
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



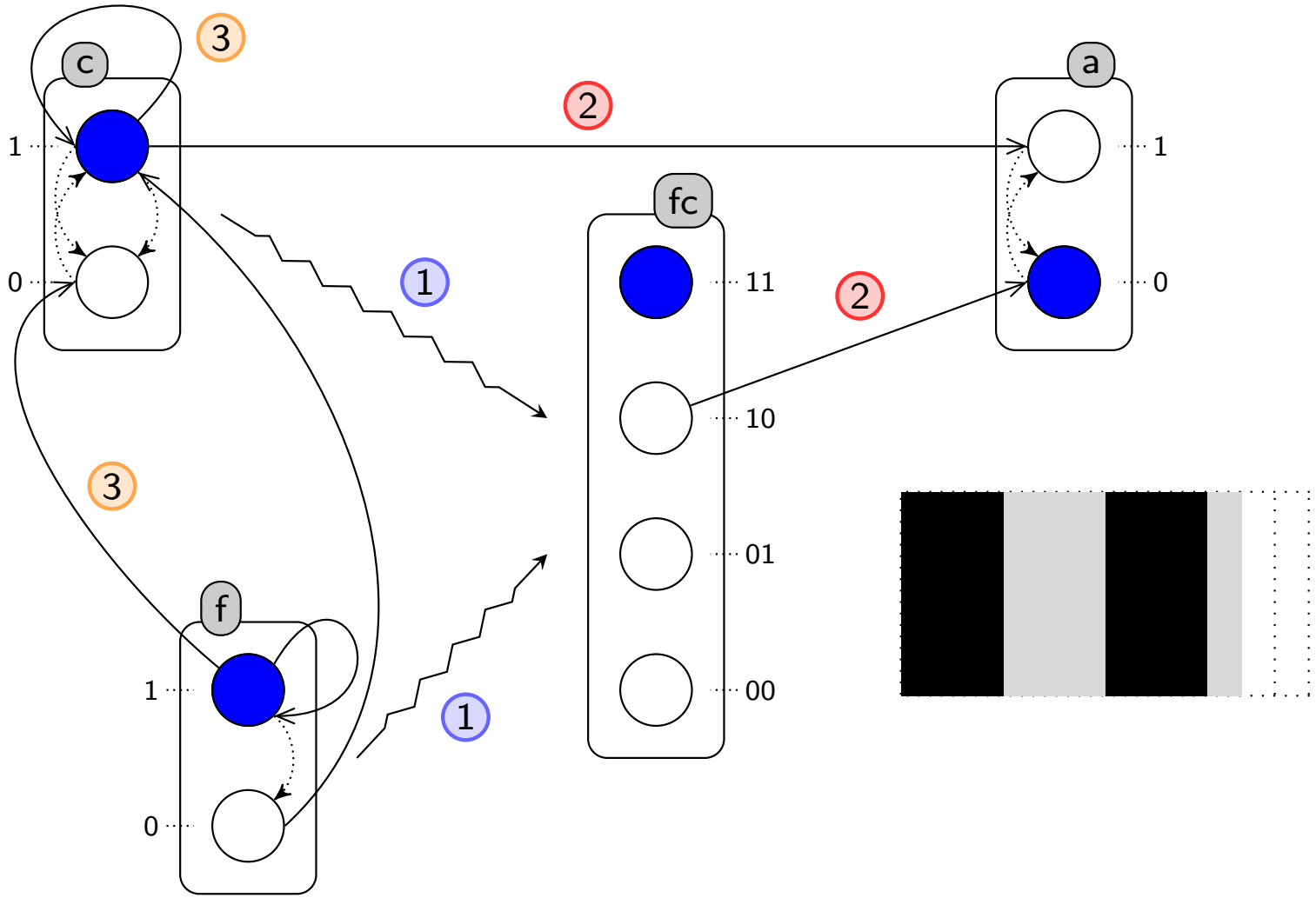
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



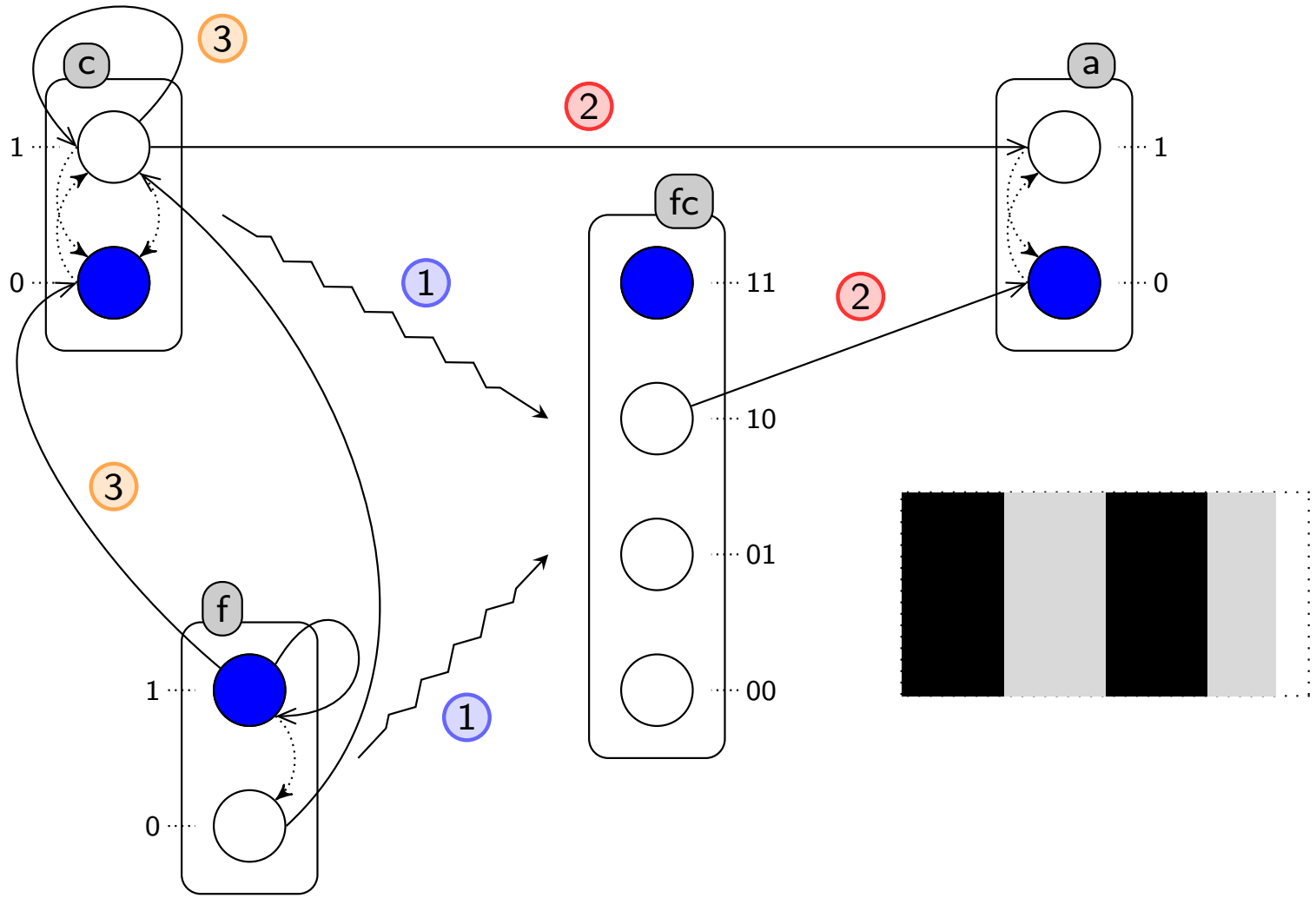
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



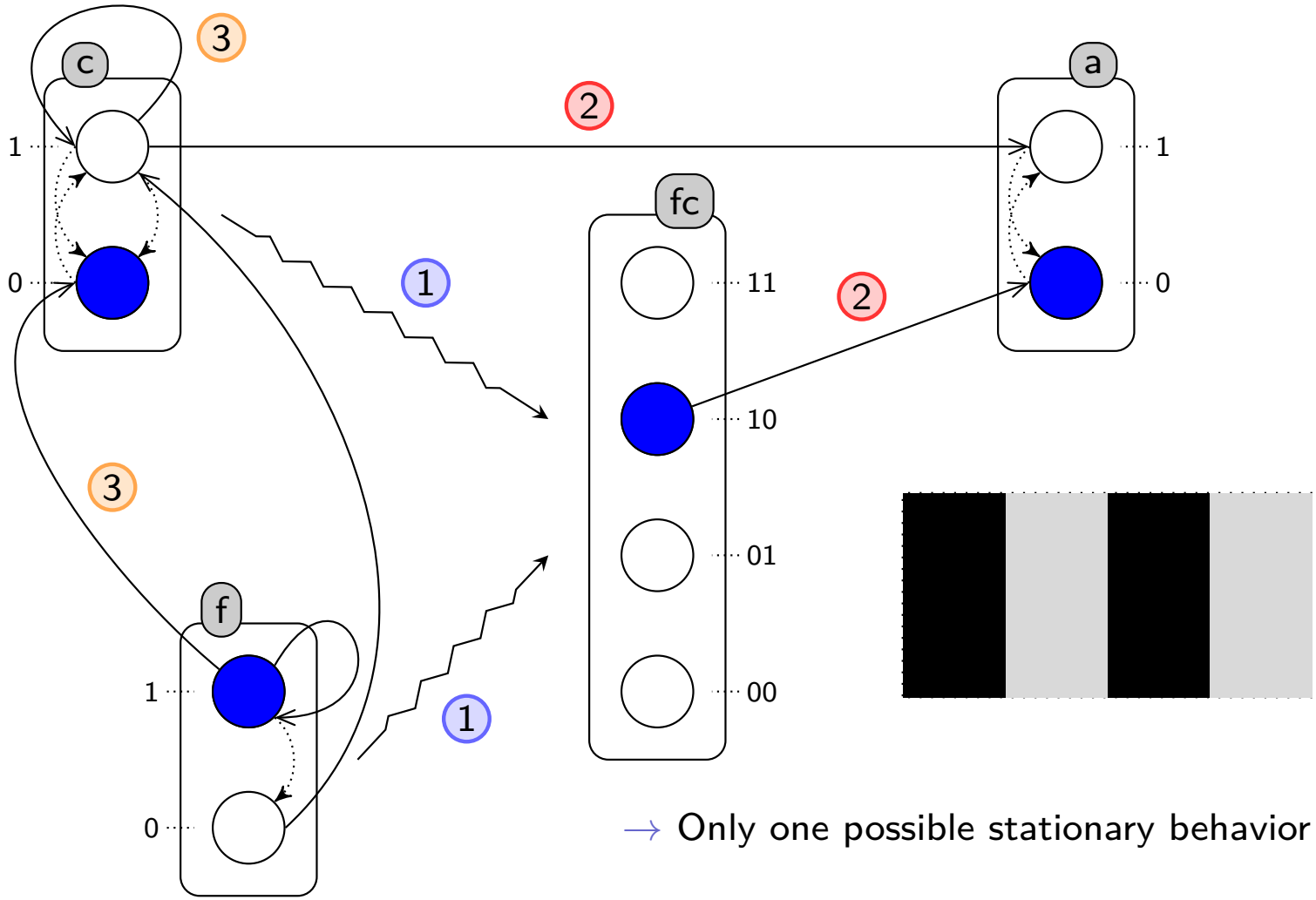
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

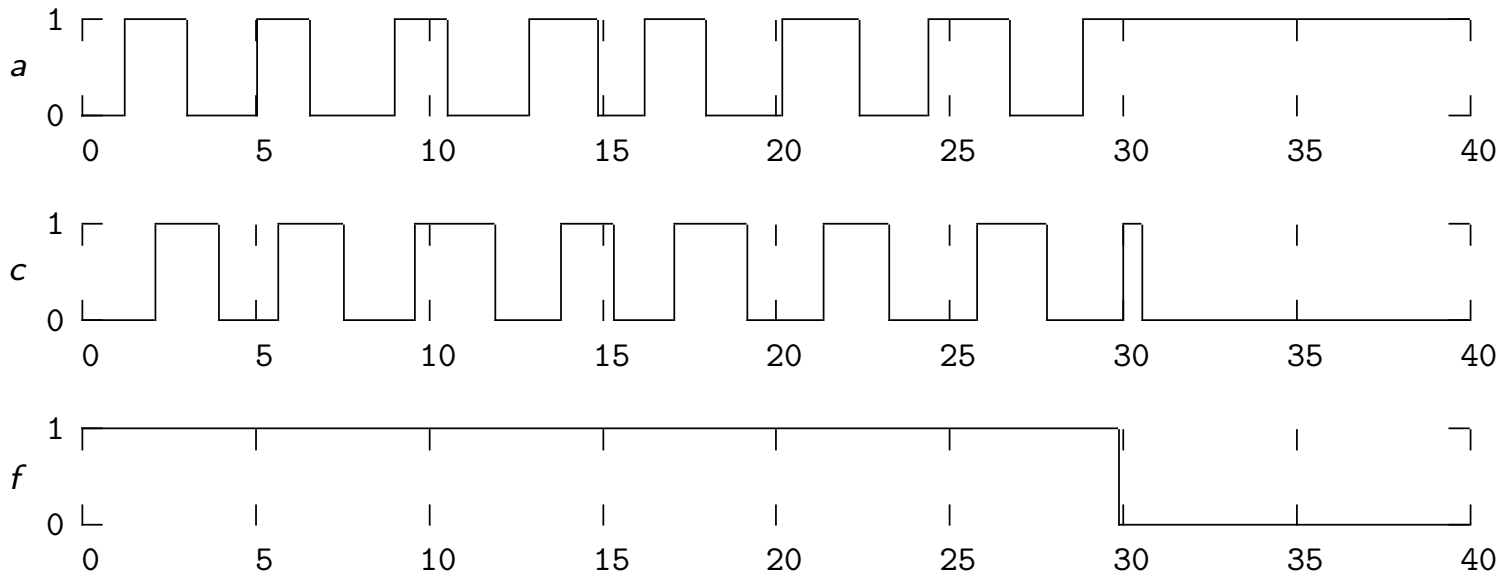


→ Only one possible stationary behavior

Abstraction of Temporal Parameters

[Paulevé (PhD thesis), 2011]

- Simulation with stochastic parameters:

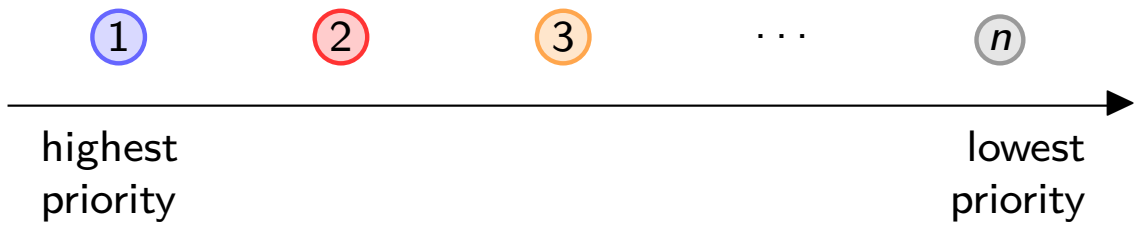


- Other possible analysis: stochastic model checkers (PRISM)
  → But combinatoric explosion: PRISM fails for more than 5 components

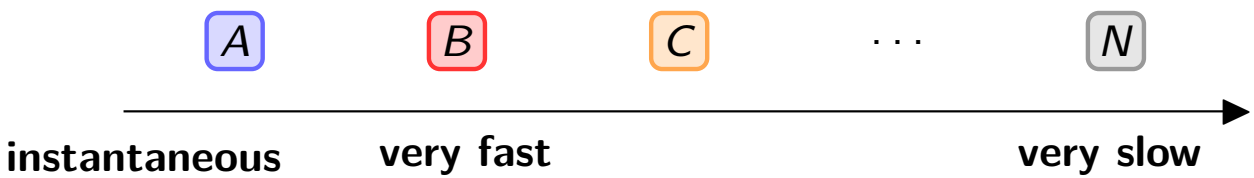
Addition of classes of priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable

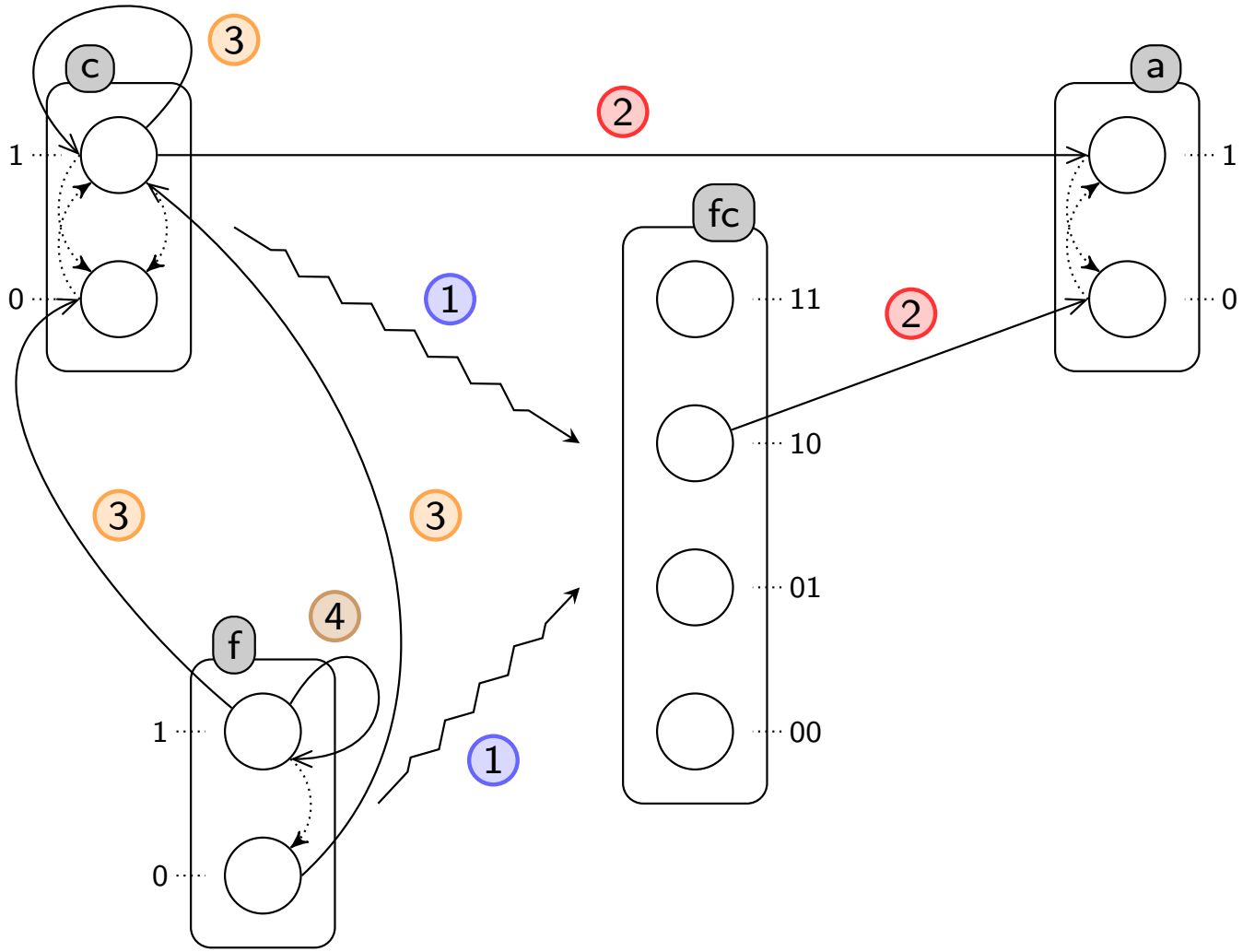


- Allow to model classes of actions with similar speeds or temporal parameters



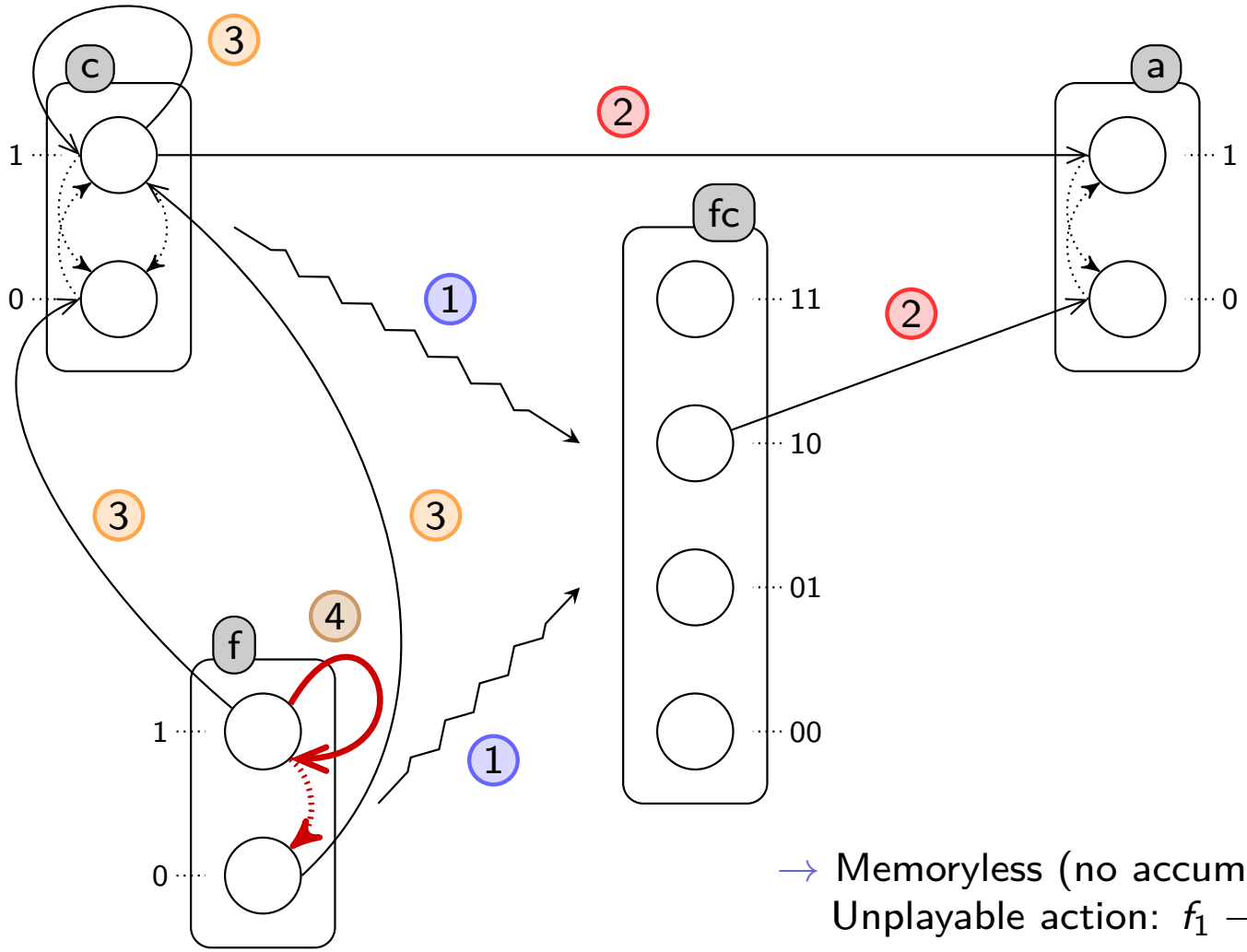
Limitation of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Limitation of the Classes of Priorities

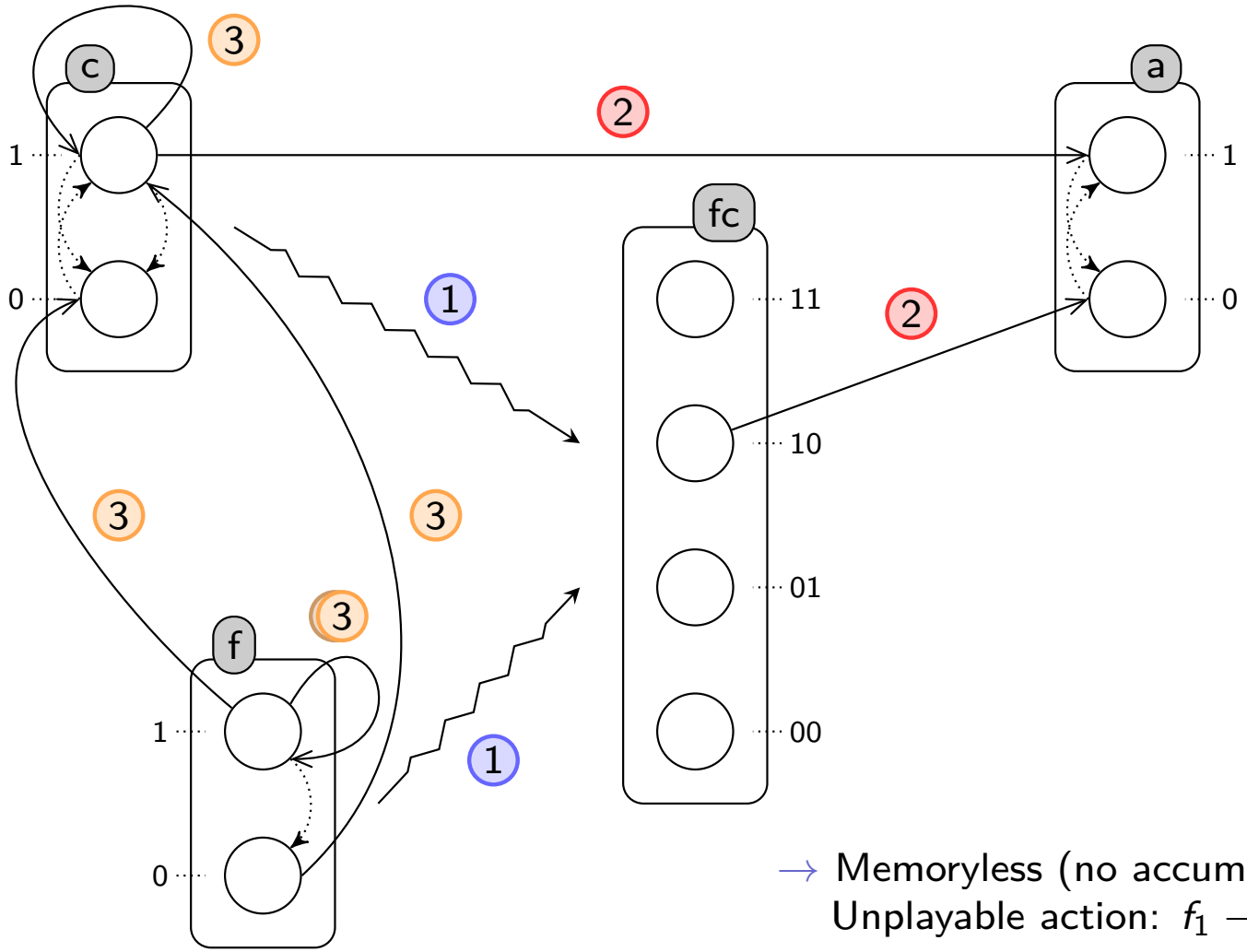
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



→ Memoryless (no accumulation)
Unplayable action: $f_1 \rightarrow f_1 \not\vdash f_0$

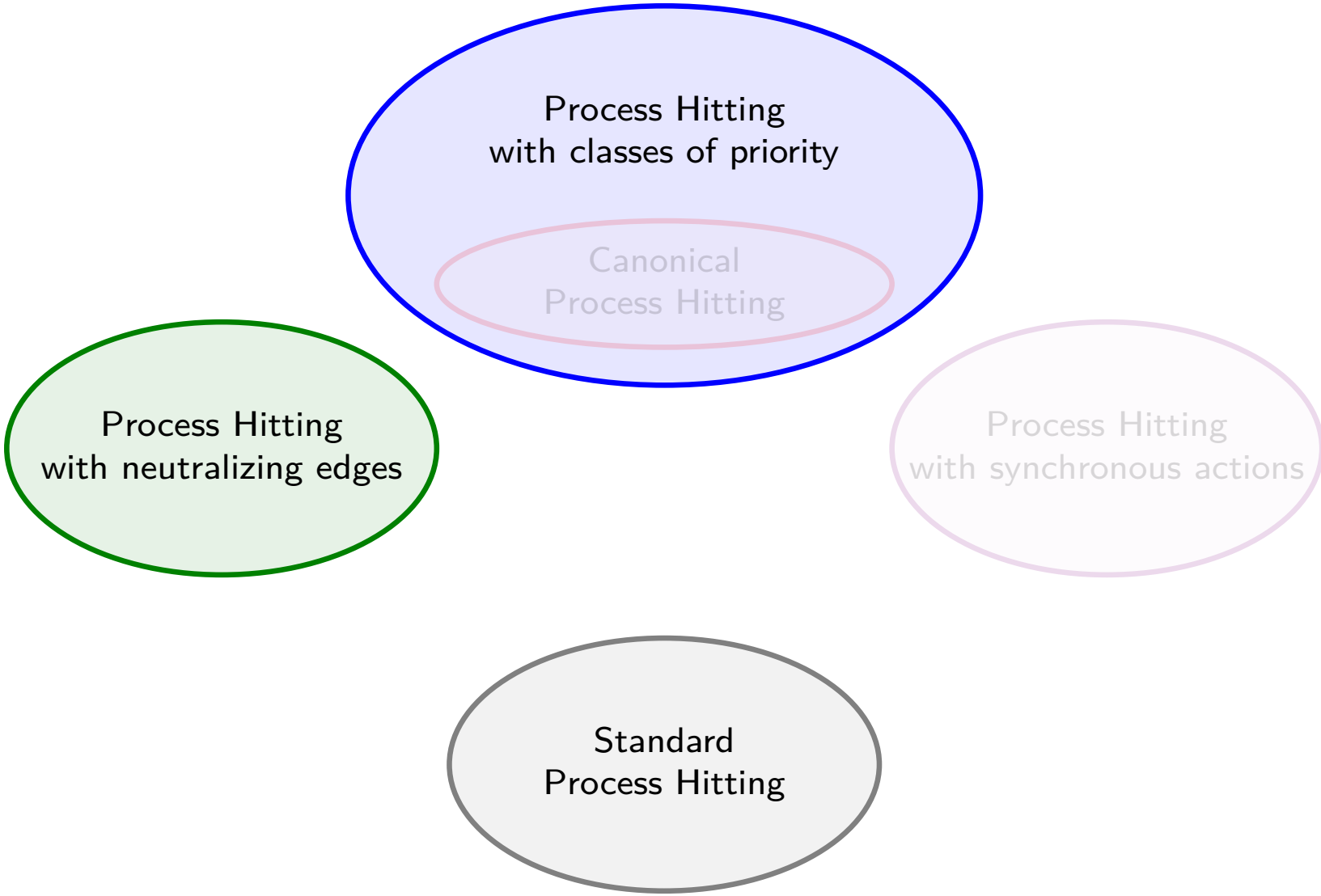
Limitation of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

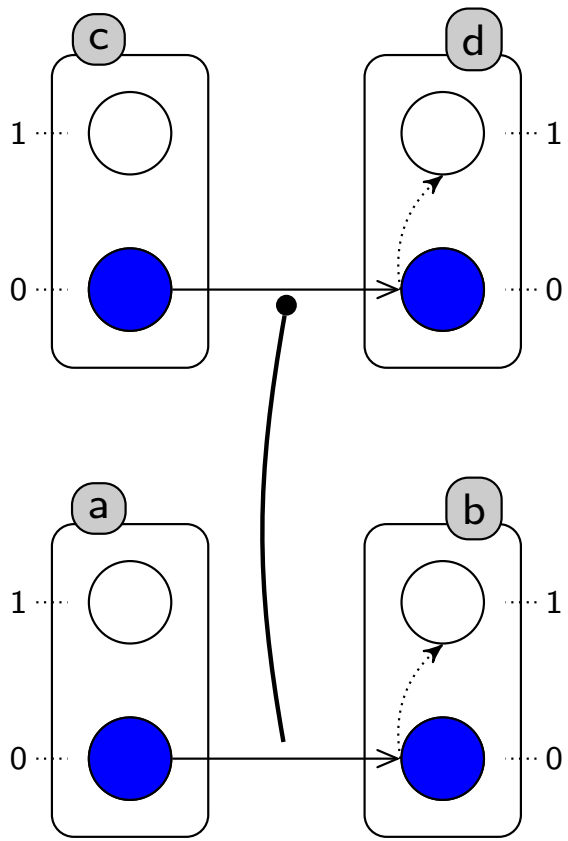


→ Memoryless (no accumulation)
Unplayable action: $f_1 \rightarrow f_1 \not\Uparrow f_0$

Process Hitting with Neutralizing Edges



Addition of Neutralizing Edges



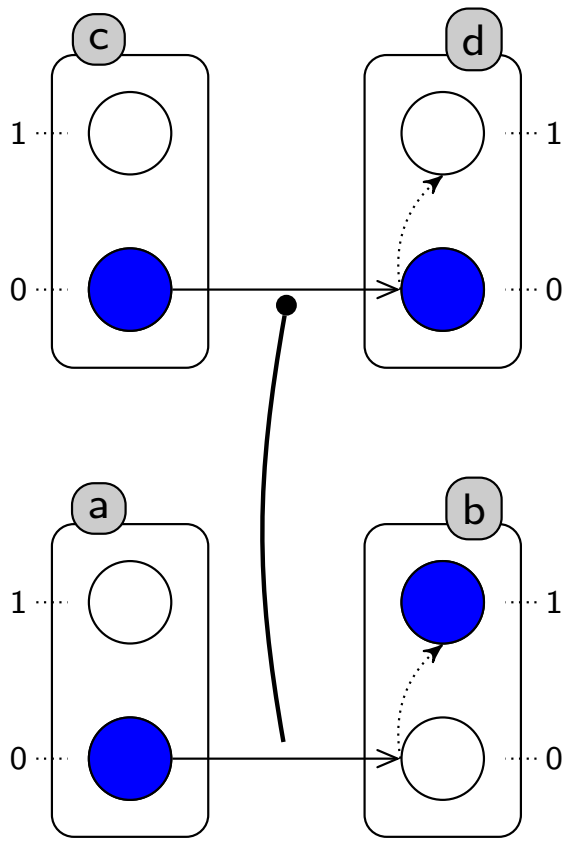
- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to "atomistic priorities"

$c_0 \rightarrow d_0 \overset{\uparrow}{\Gamma} d_1$ cannot be plays **while**

$a_0 \rightarrow b_0 \overset{\uparrow}{\Gamma} b_1$ is playable

$\rightarrow d_1$ is **always** reached after b_1

Addition of Neutralizing Edges



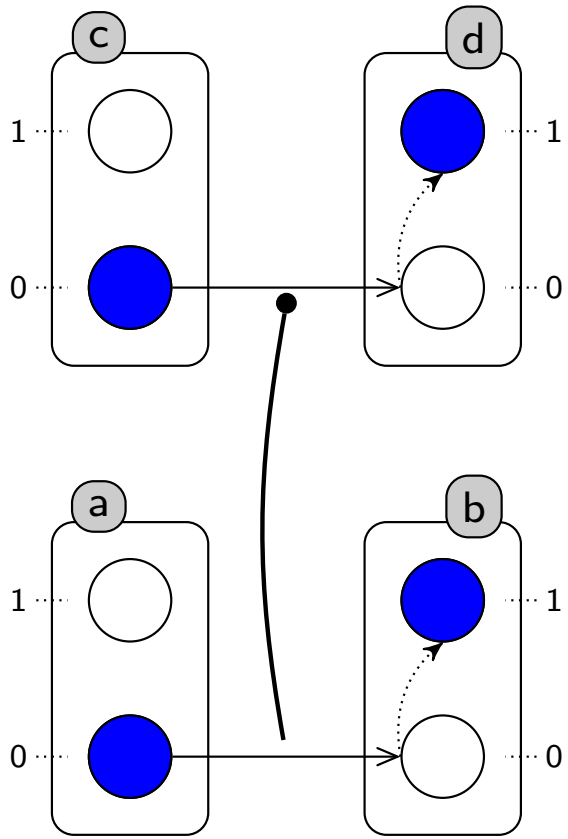
- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to "atomistic priorities"

$c_0 \rightarrow d_0 \not\uparrow d_1$ cannot be plays **while**

$a_0 \rightarrow b_0 \not\uparrow b_1$ is playable

$\rightarrow d_1$ is **always** reached after b_1

Addition of Neutralizing Edges



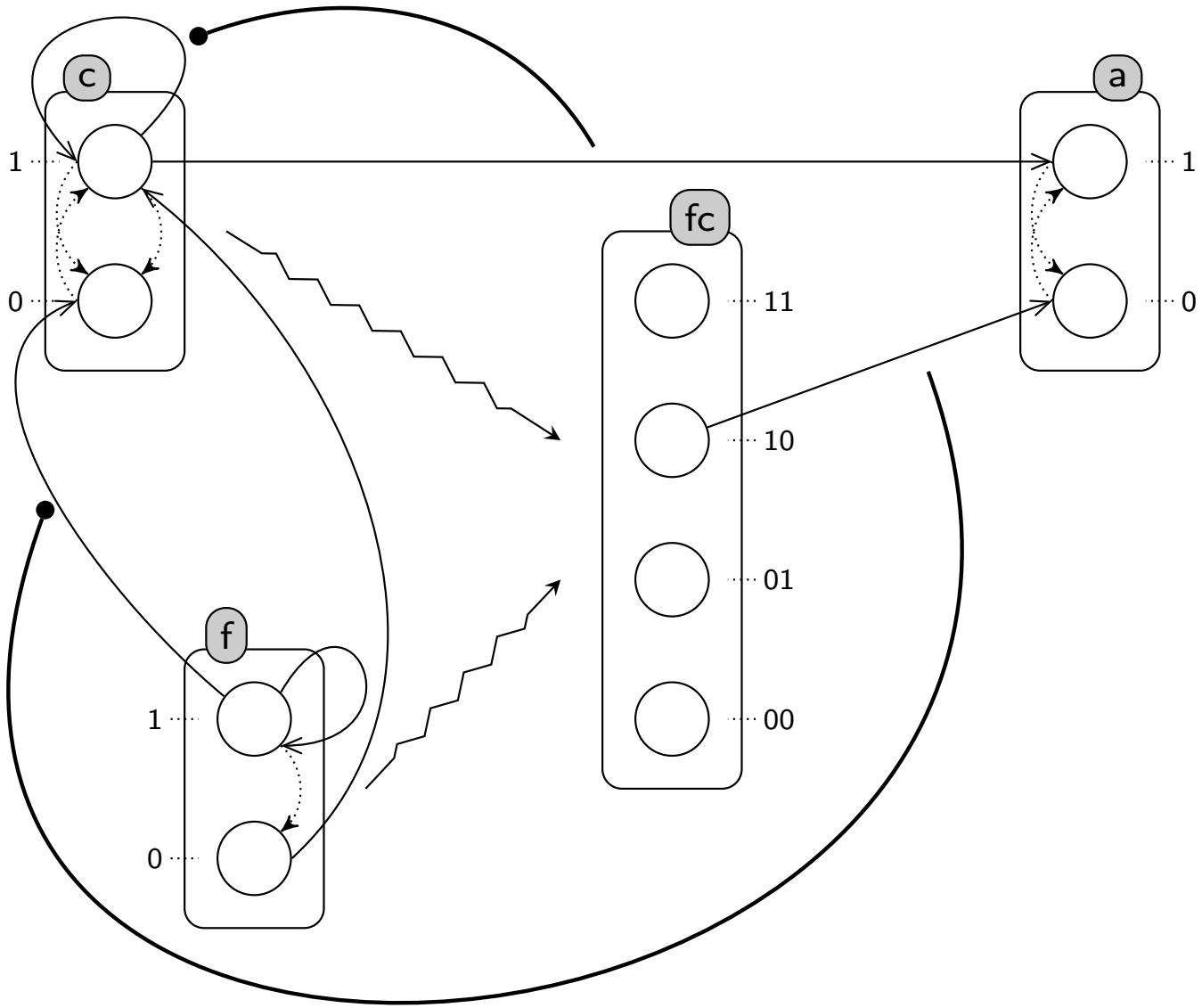
- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to "atomistic priorities"

$c_0 \rightarrow d_0 \Rsh d_1$ cannot be plays **while**

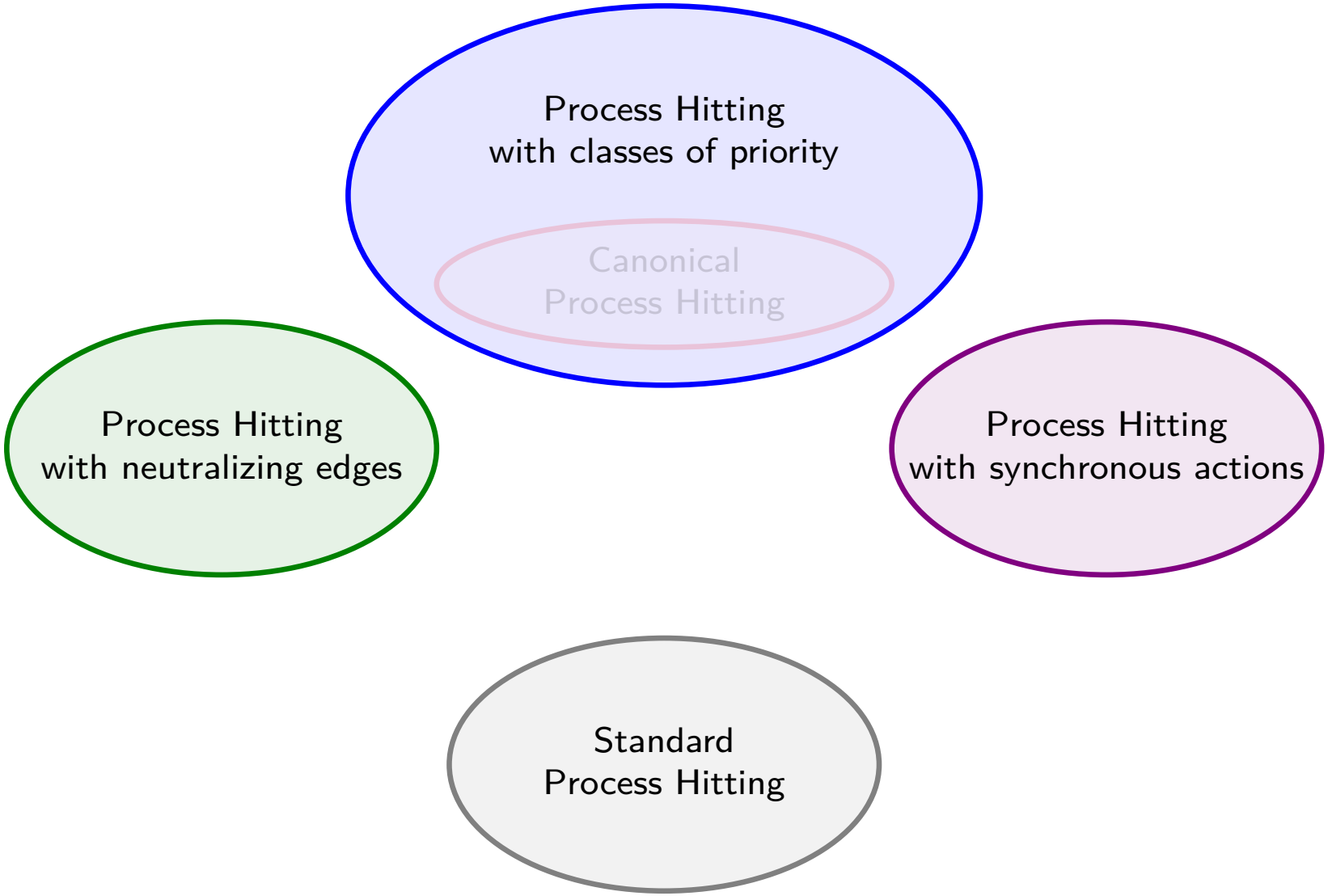
$a_0 \rightarrow b_0 \Rsh b_1$ is playable

$\rightarrow d_1$ is **always** reached after b_1

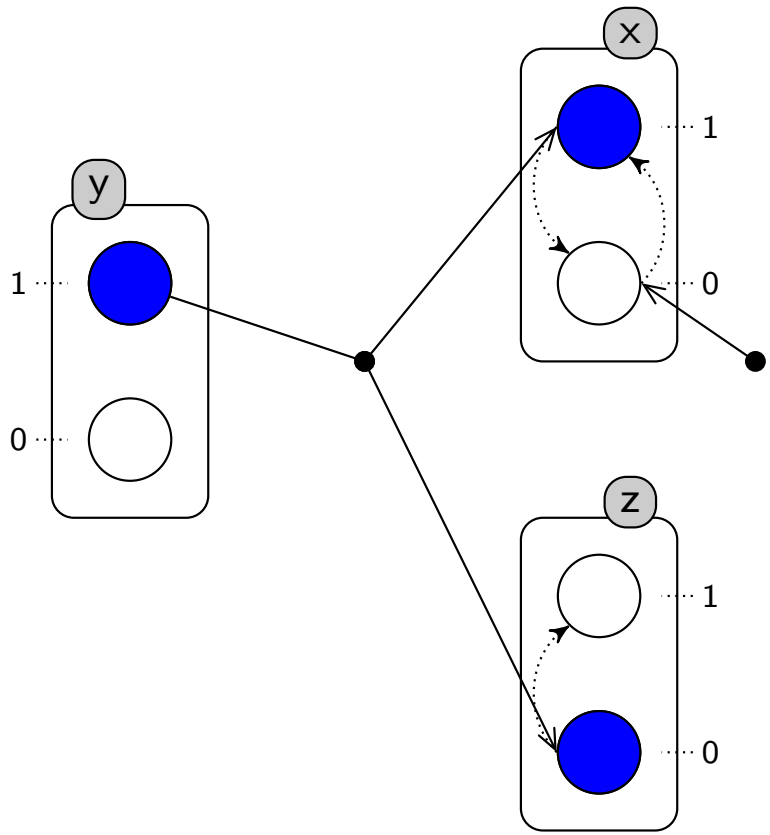
Use of Neutralizing Edges



Process Hitting with Synchronous Actions



Addition of Synchronous Actions



- Synchronizations between actions:
  - All catalysts must be present
  - Reactants are consumed all together
  - Simultaneous creation of the products
- Representation of biochemical equations:

$$X \xrightarrow{Y} Z$$

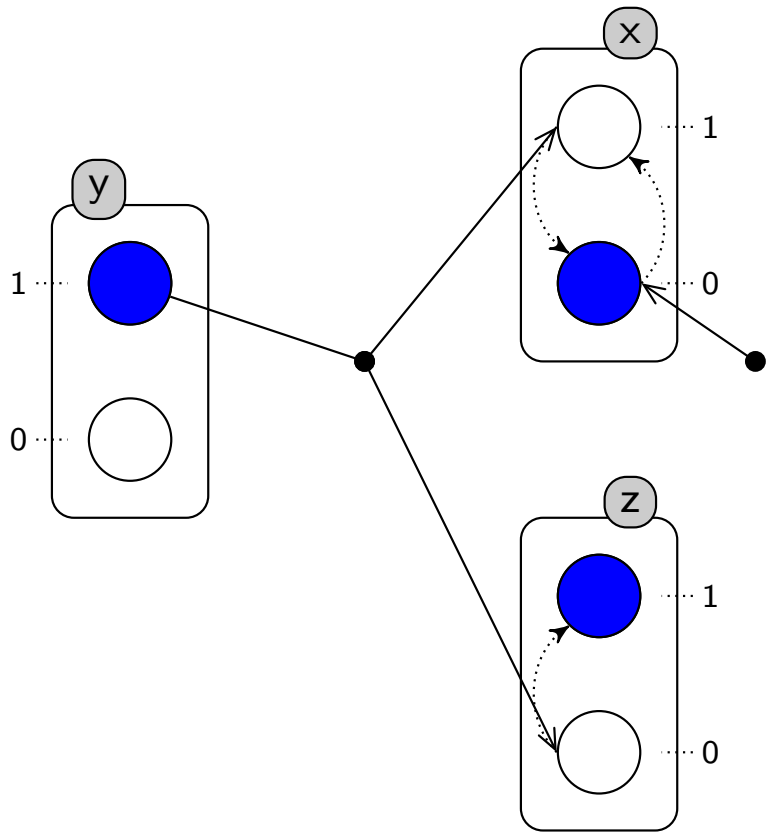
under the form:

$$h_2 = \{x_1, y_1, z_0\} \rightarrowtail \{x_0, z_1\}$$

All processes of A
must be present to play $A \rightarrowtail B$

After the play of $A \rightarrowtail B$,
all processes of B are present

Addition of Synchronous Actions



- Synchronizations between actions:
  - All catalysts must be present
  - Reactants are consumed all together
  - Simultaneous creation of the products
- Representation of biochemical equations:

$$X \xrightarrow{Y} Z$$

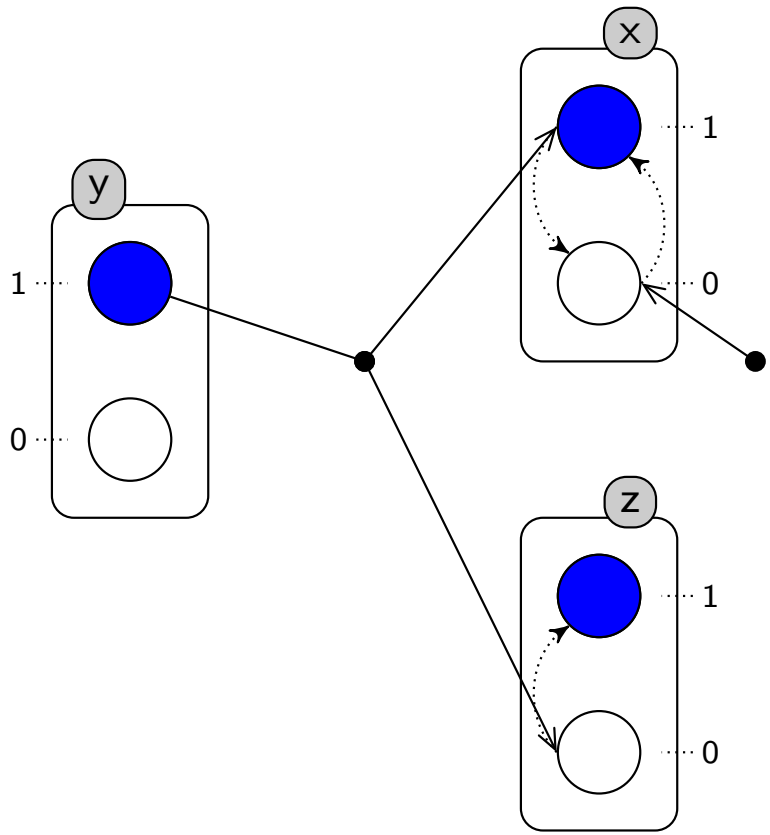
under the form:

$$h_2 = \{x_1, y_1, z_0\} \rightarrowtail \{x_0, z_1\}$$

All processes of A
must be present to play $A \rightarrowtail B$

After the play of $A \rightarrowtail B$,
all processes of B are present

Addition of Synchronous Actions



- Synchronizations between actions:
  - All catalysts must be present
  - Reactants are consumed all together
  - Simultaneous creation of the products
- Representation of biochemical equations:

$$X \xrightarrow{Y} Z$$

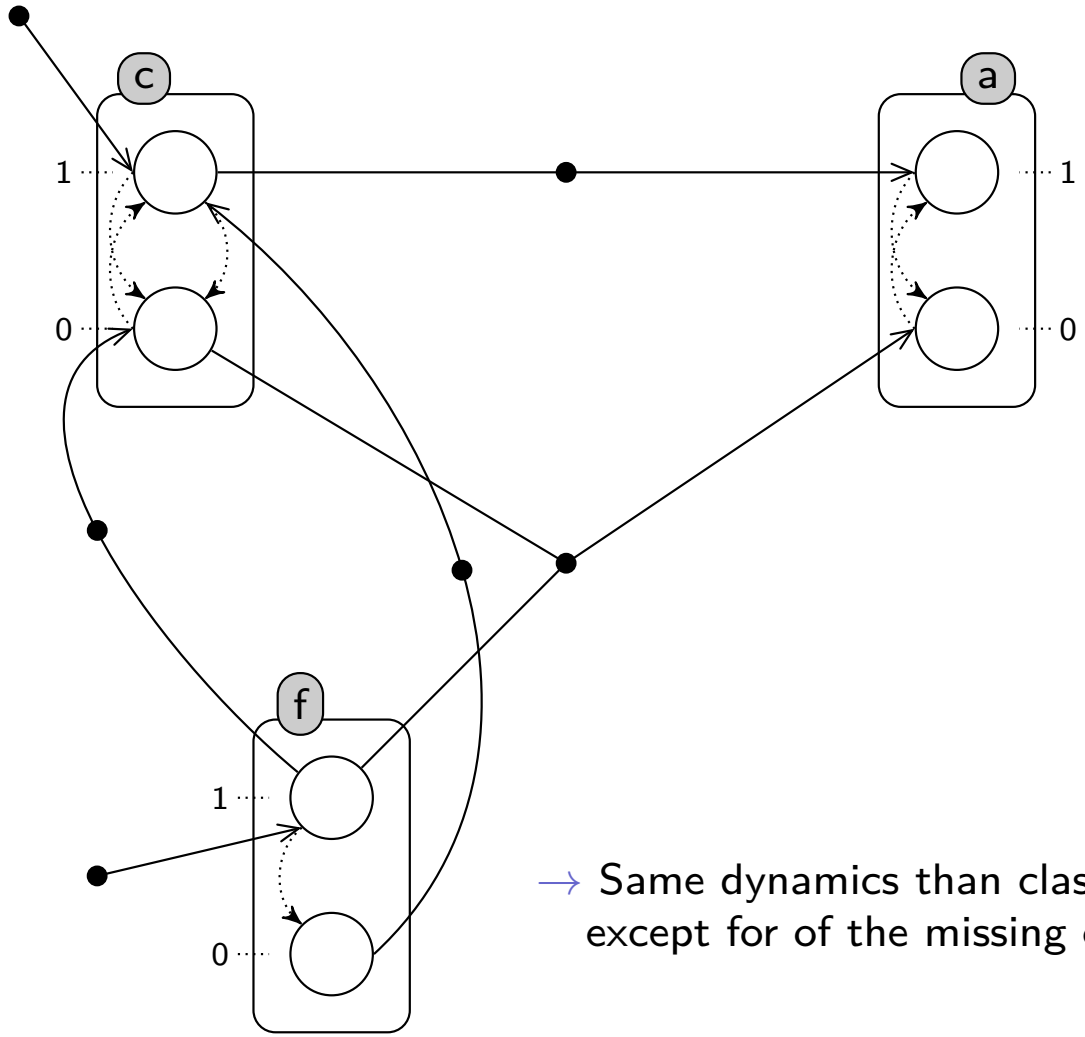
under the form:

$$h_2 = \{x_1, y_1, z_0\} \rightarrowtail \{x_0, z_1\}$$

All processes of A
must be present to play $A \rightarrowtail B$

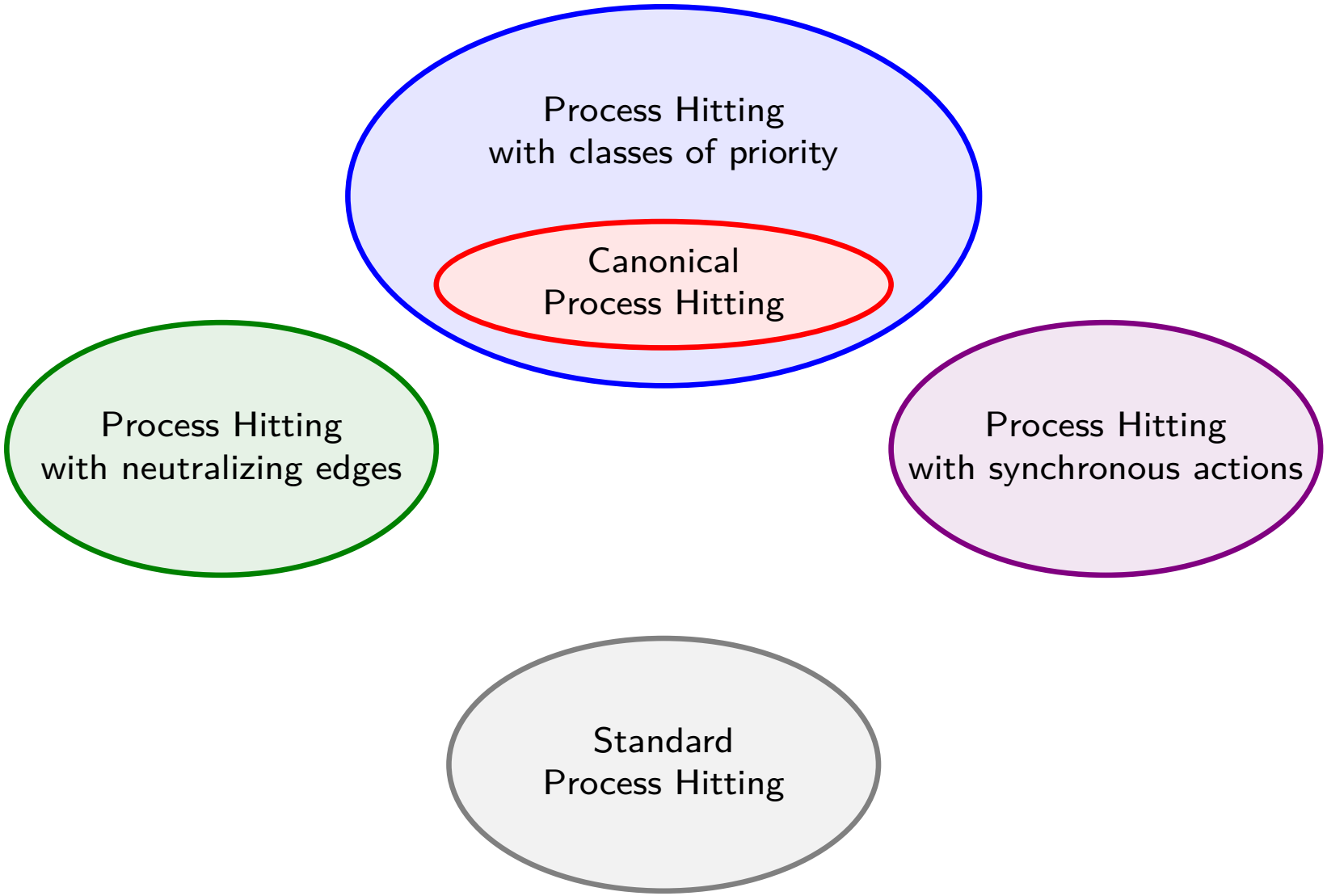
After the play of $A \rightarrowtail B$,
all processes of B are present

Use of Synchronous Actions



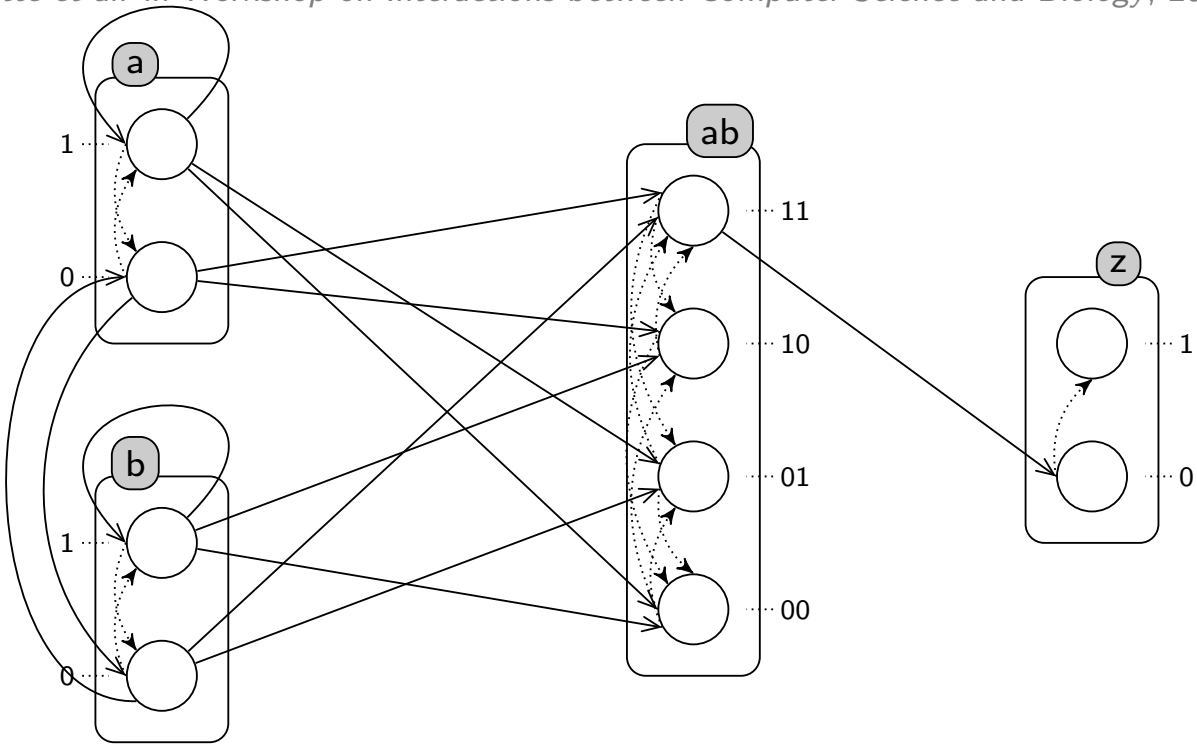
→ Same dynamics than classes of priorities, except for of the missing cooperative sort

Canonical Process Hitting



Temporal Shift in Cooperative Sorts

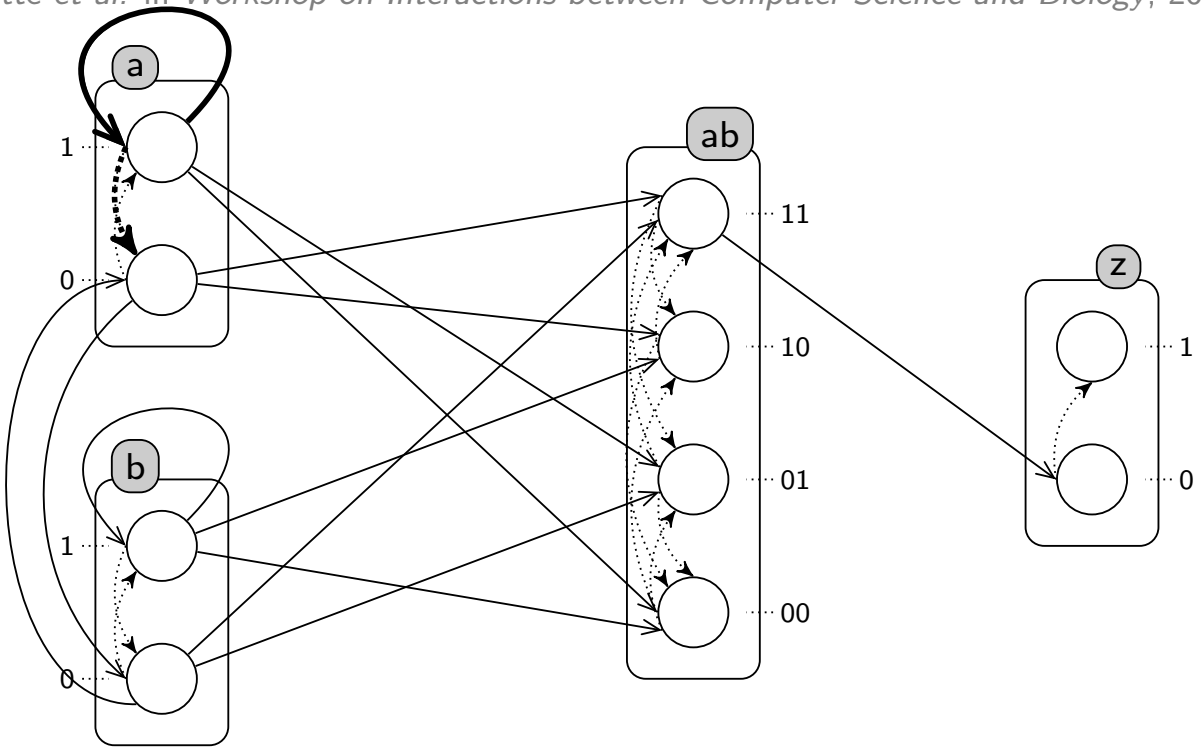
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Drawback: the cooperative sorts are too "loose" (temporal shift)

Temporal Shift in Cooperative Sorts

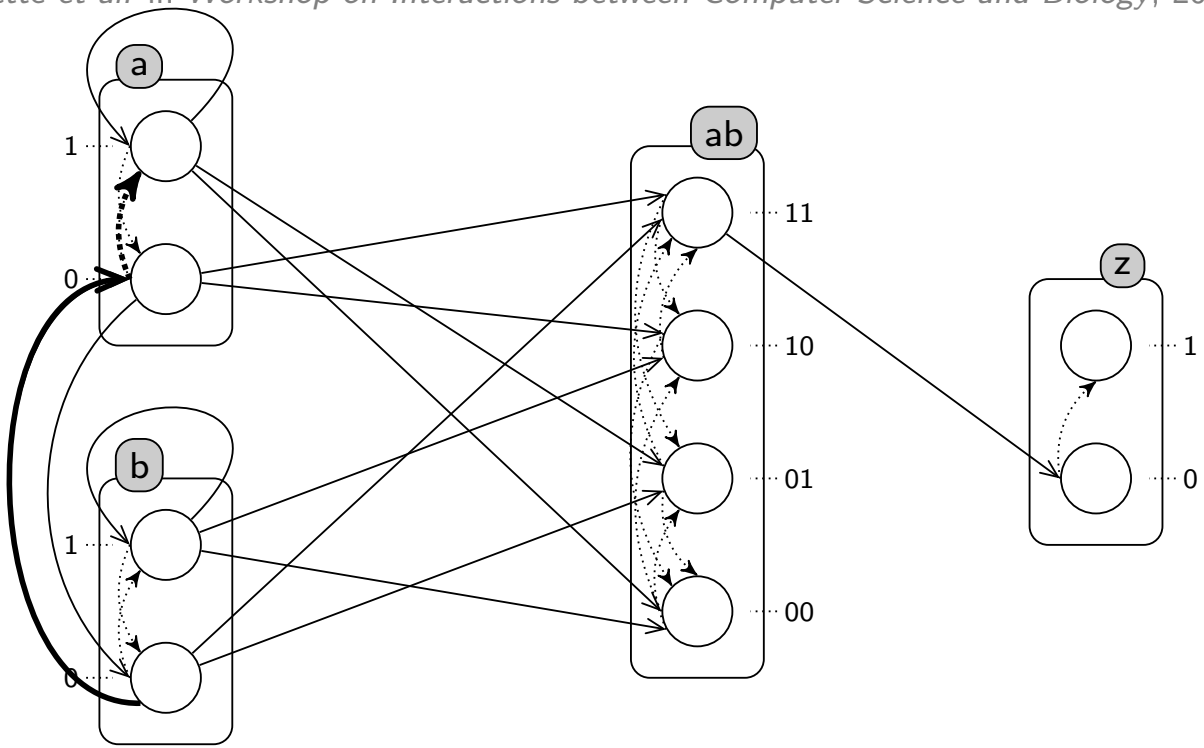
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Drawback: the cooperative sorts are too "loose" (temporal shift)

Temporal Shift in Cooperative Sorts

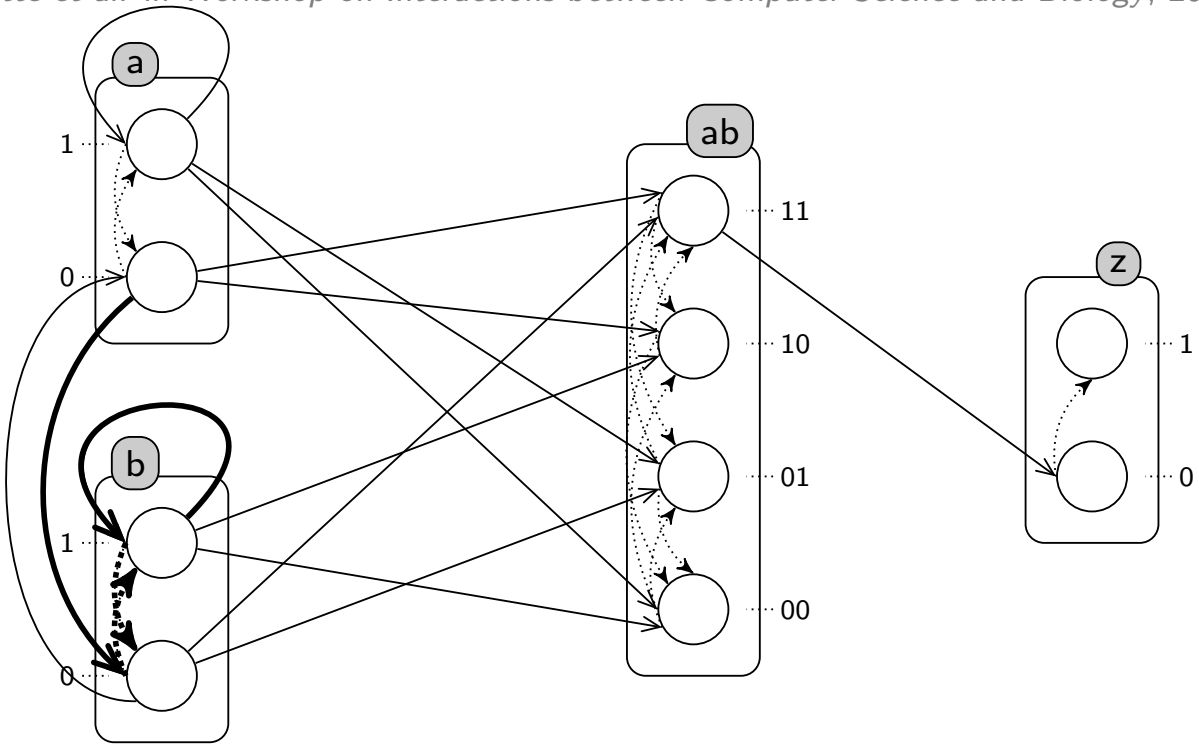
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Drawback: the cooperative sorts are too "loose" (temporal shift)

Temporal Shift in Cooperative Sorts

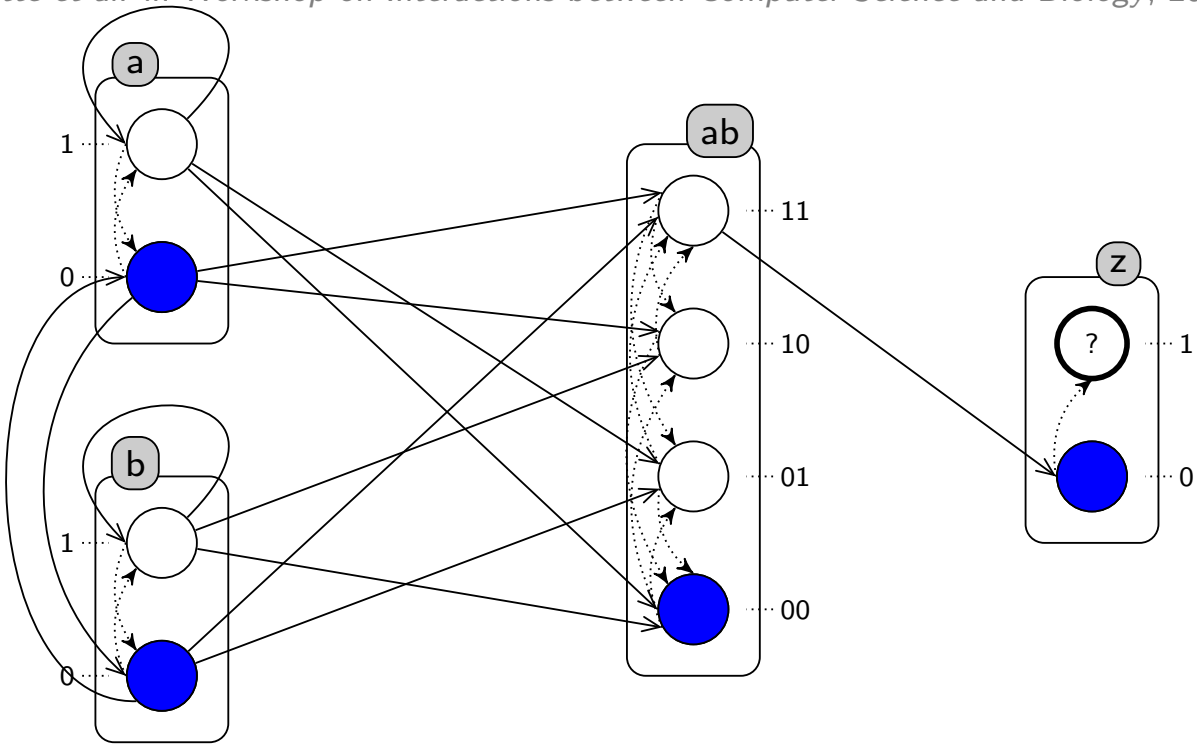
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Drawback: the cooperative sorts are too "loose" (temporal shift)

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

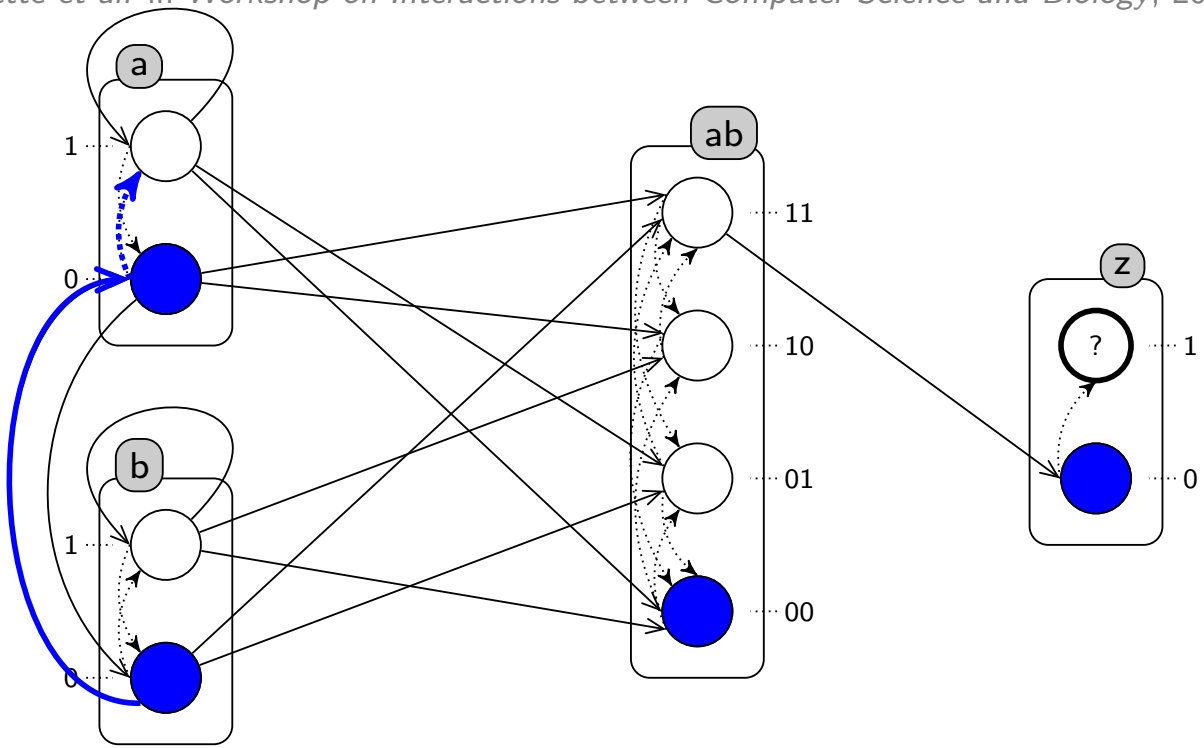


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

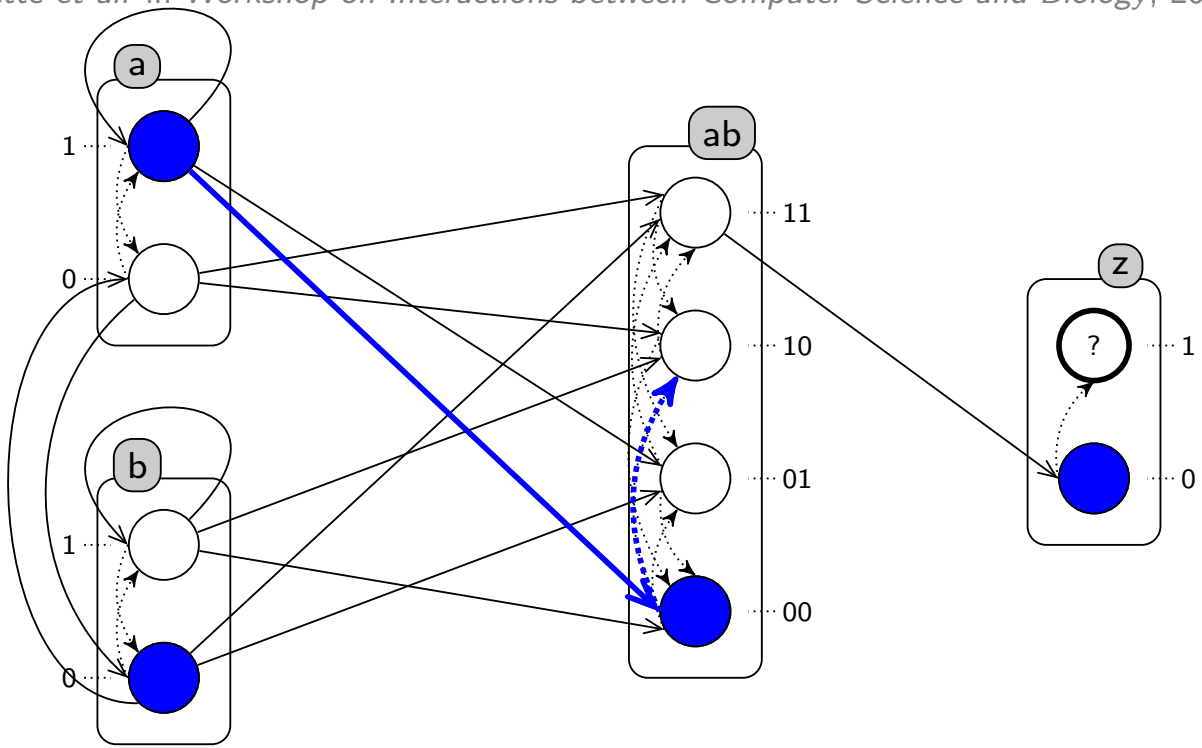


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

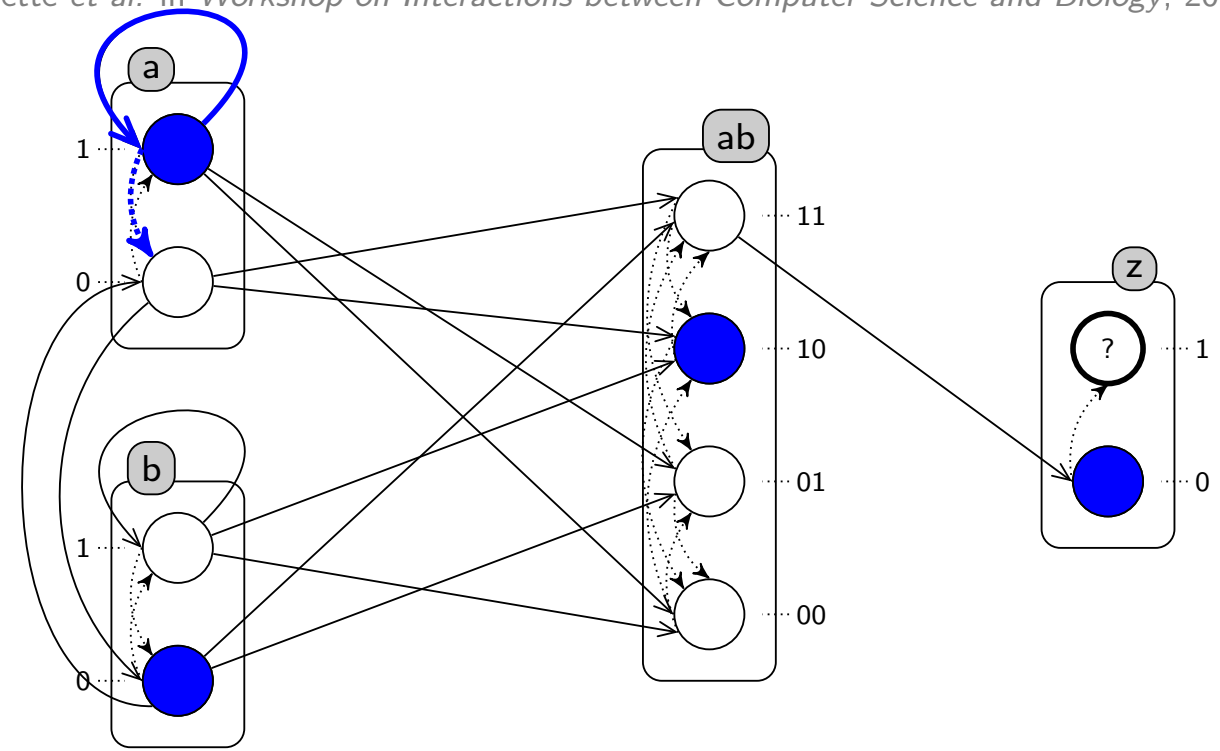


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

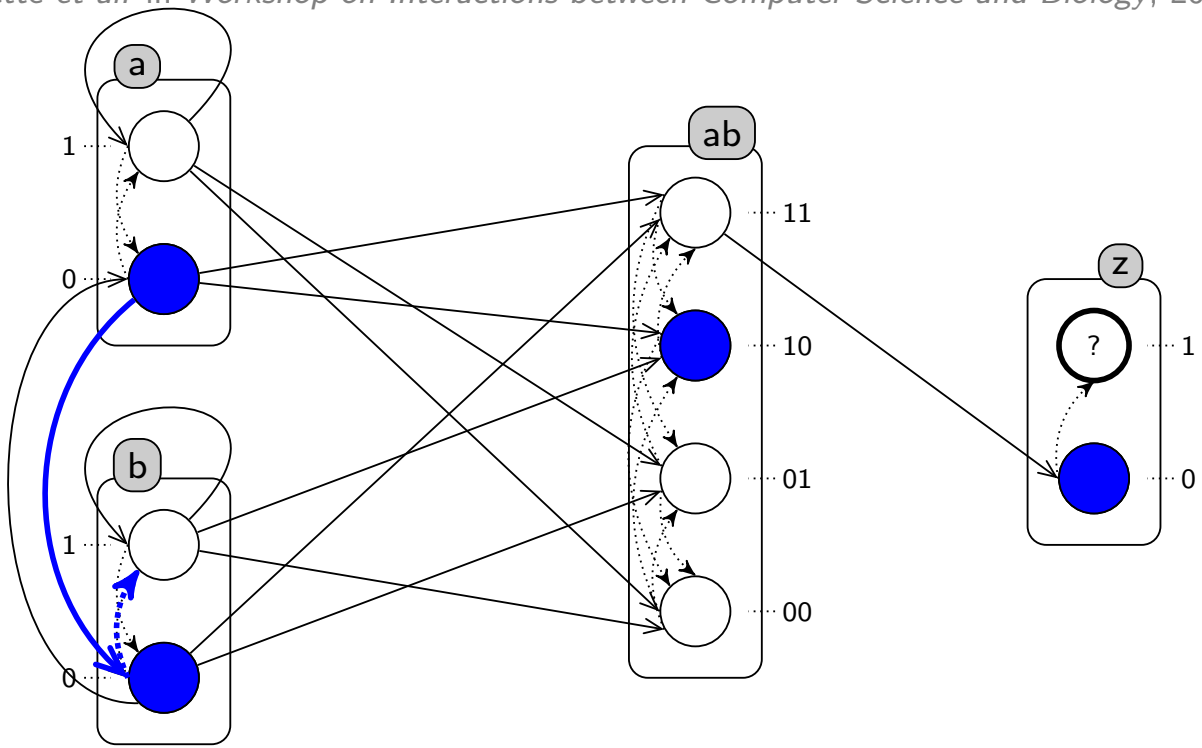


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

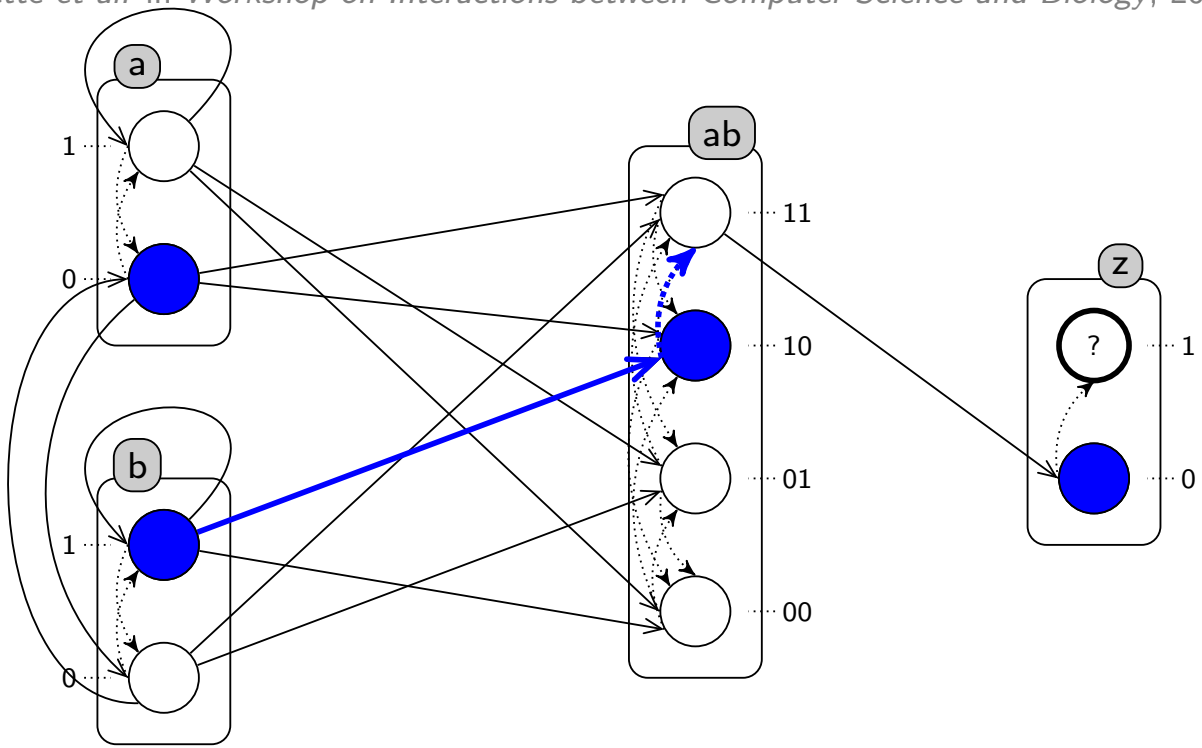


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

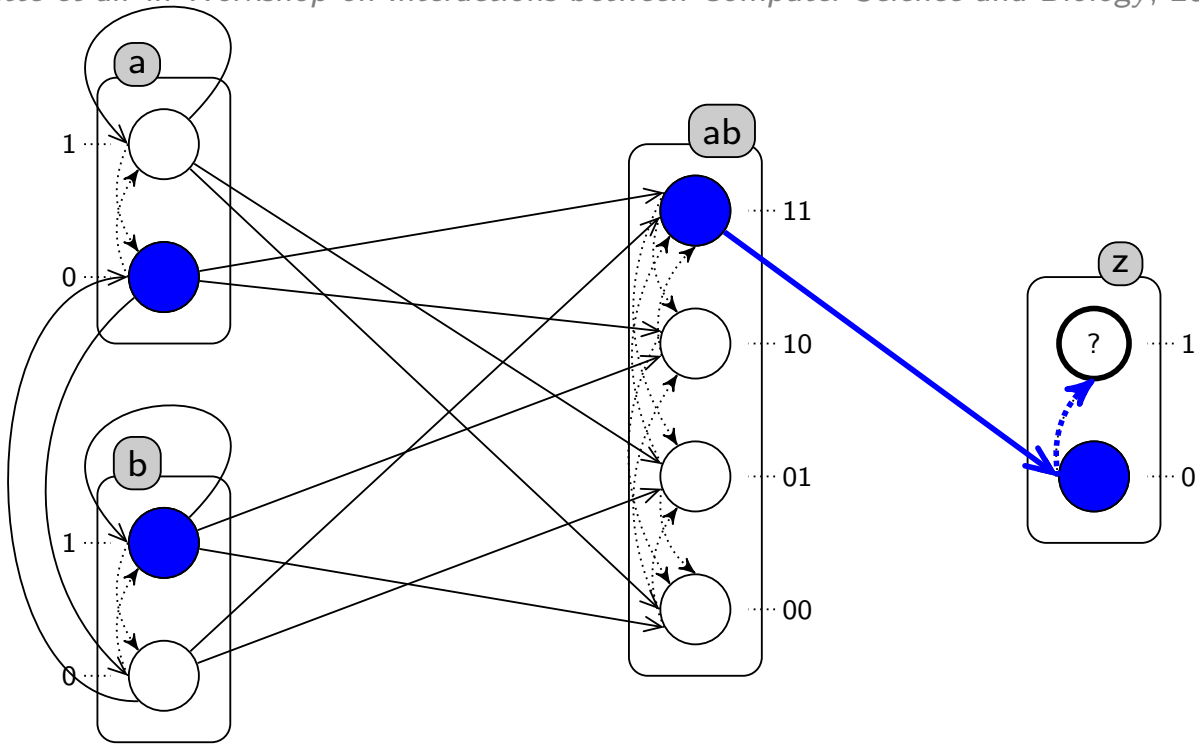


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{10}, z_0 \rangle \to \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\to \langle a_0, b_1, ab_{10}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

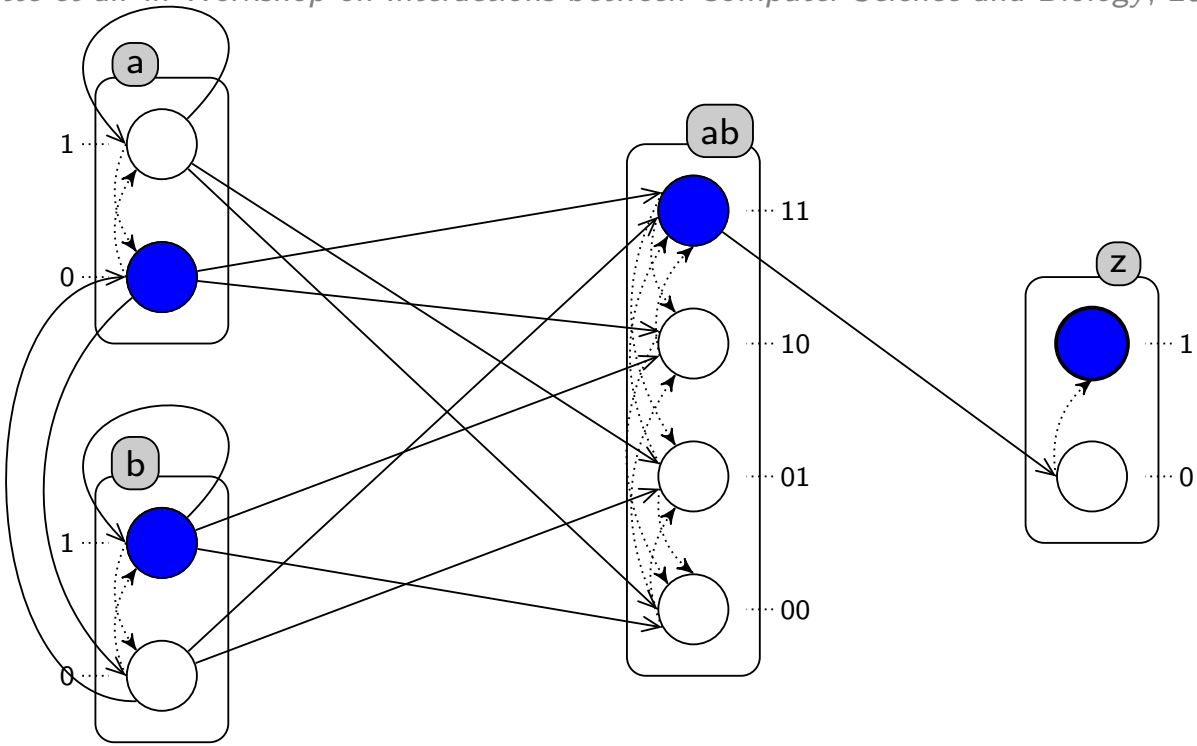


Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\rightarrow \langle a_0, b_1, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{11}, z_0 \rangle$

Temporal Shift in Cooperative Sorts

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



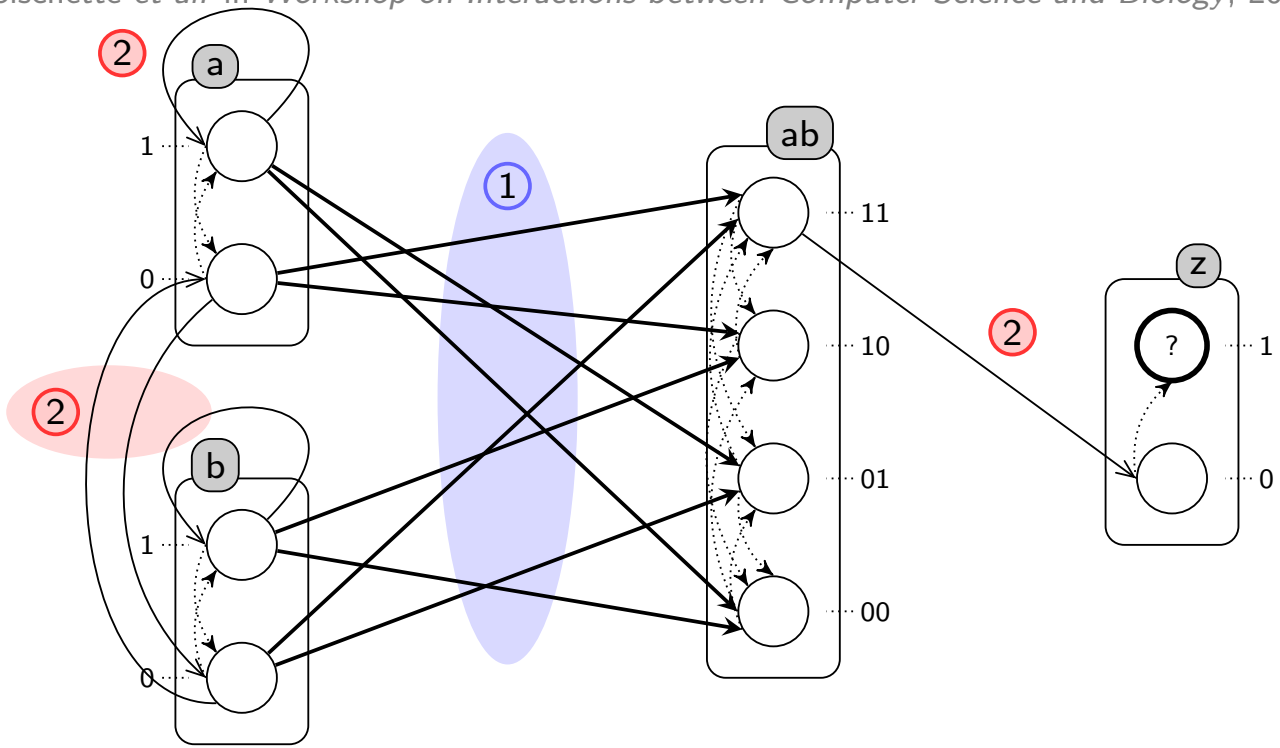
Drawback: the cooperative sorts are too "loose" (temporal shift)

$\langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{10}, z_0 \rangle \to \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\to \langle a_0, b_1, ab_{10}, z_0 \rangle \to \langle a_0, b_1, ab_{11}, z_0 \rangle \to \langle a_0, b_1, ab_{11}, z_1 \rangle$

Expected behavior: $a_1 \wedge b_1$ **simultaneously** i.e. "in the same state"
Obtained behavior: $\mathbf{P}(a_1) \wedge \mathbf{P}(b_1)$ with $\mathbf{P}$ = "previously"

Canonical Process Hitting

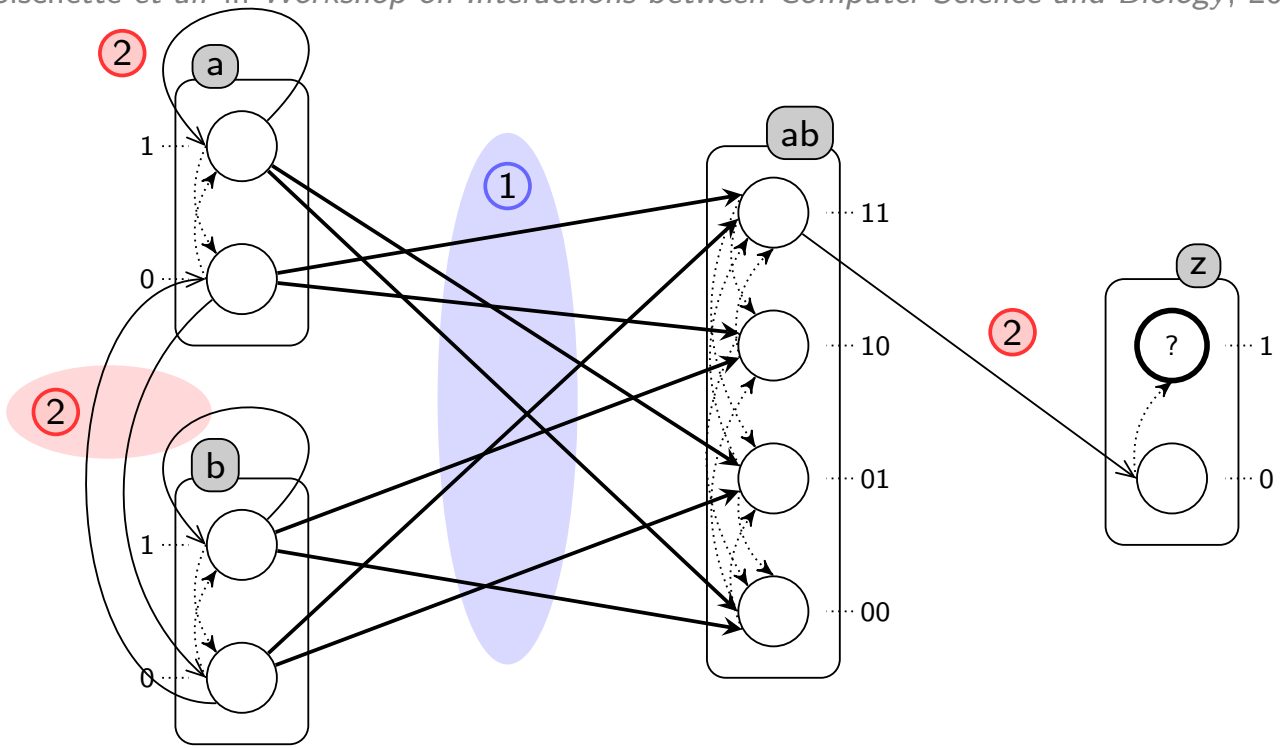
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

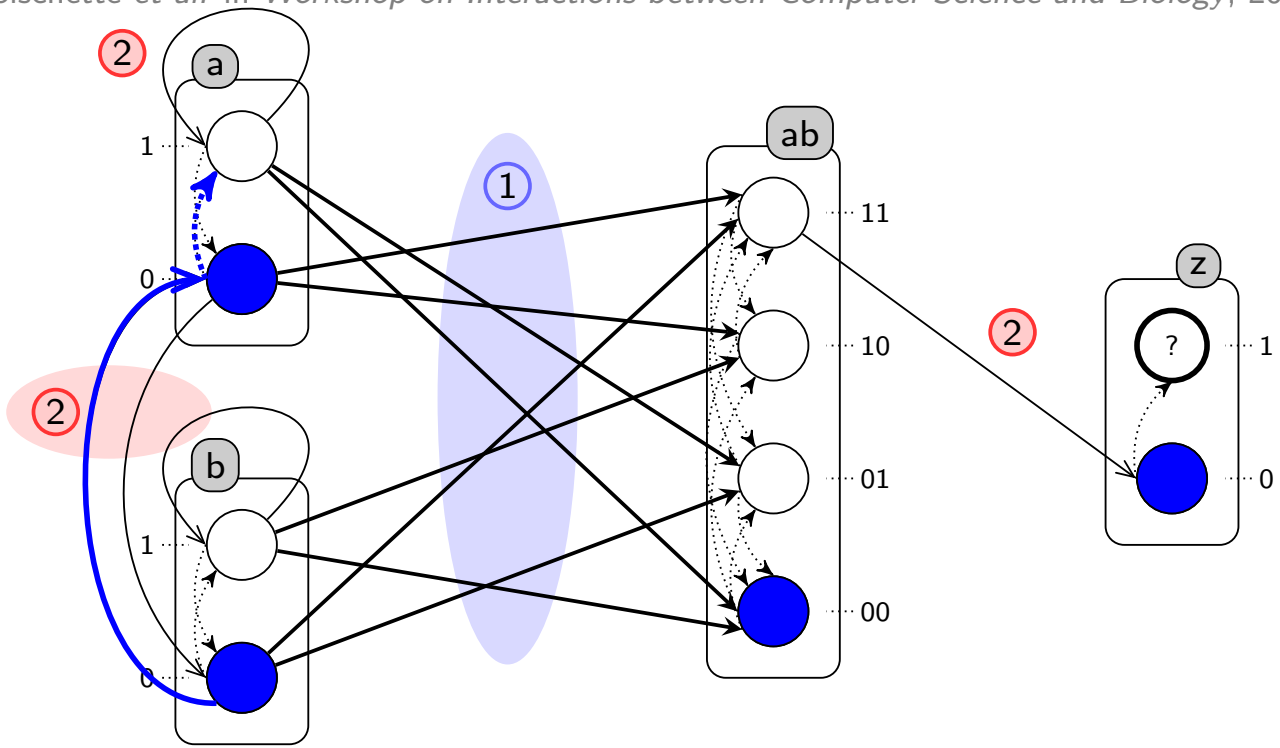


- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



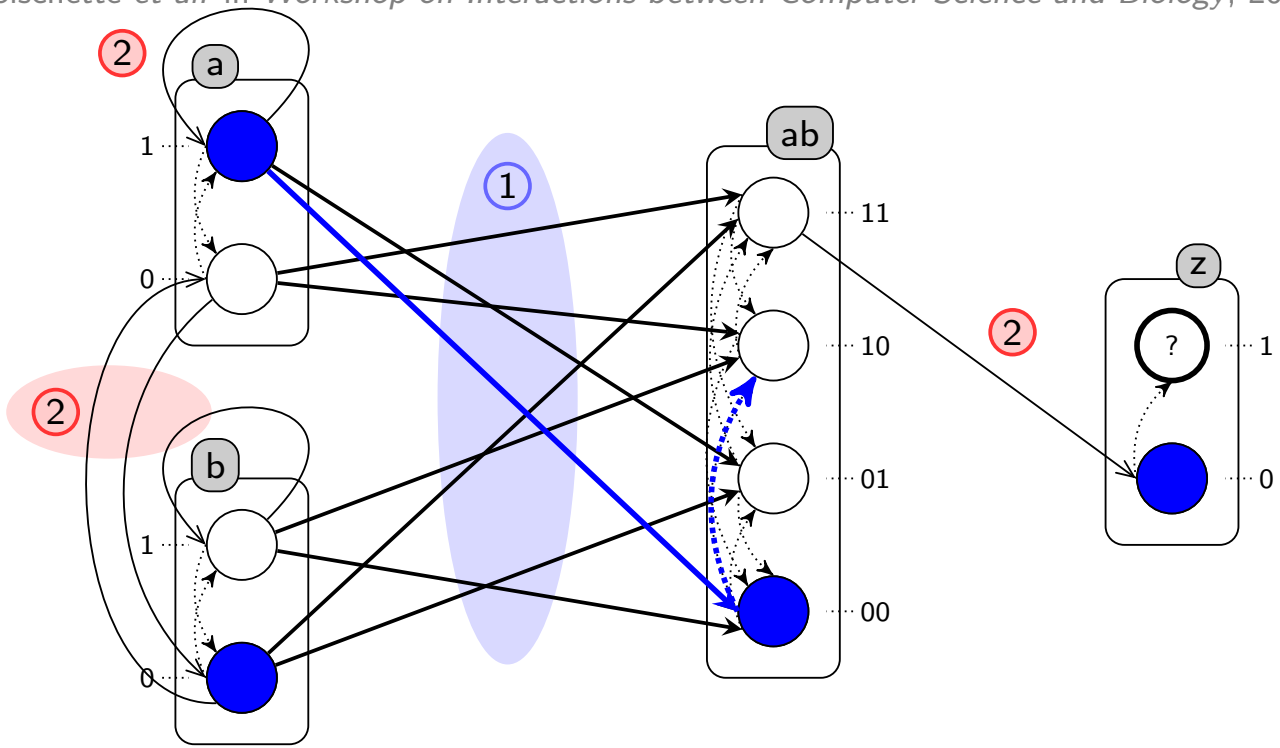
- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle$

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



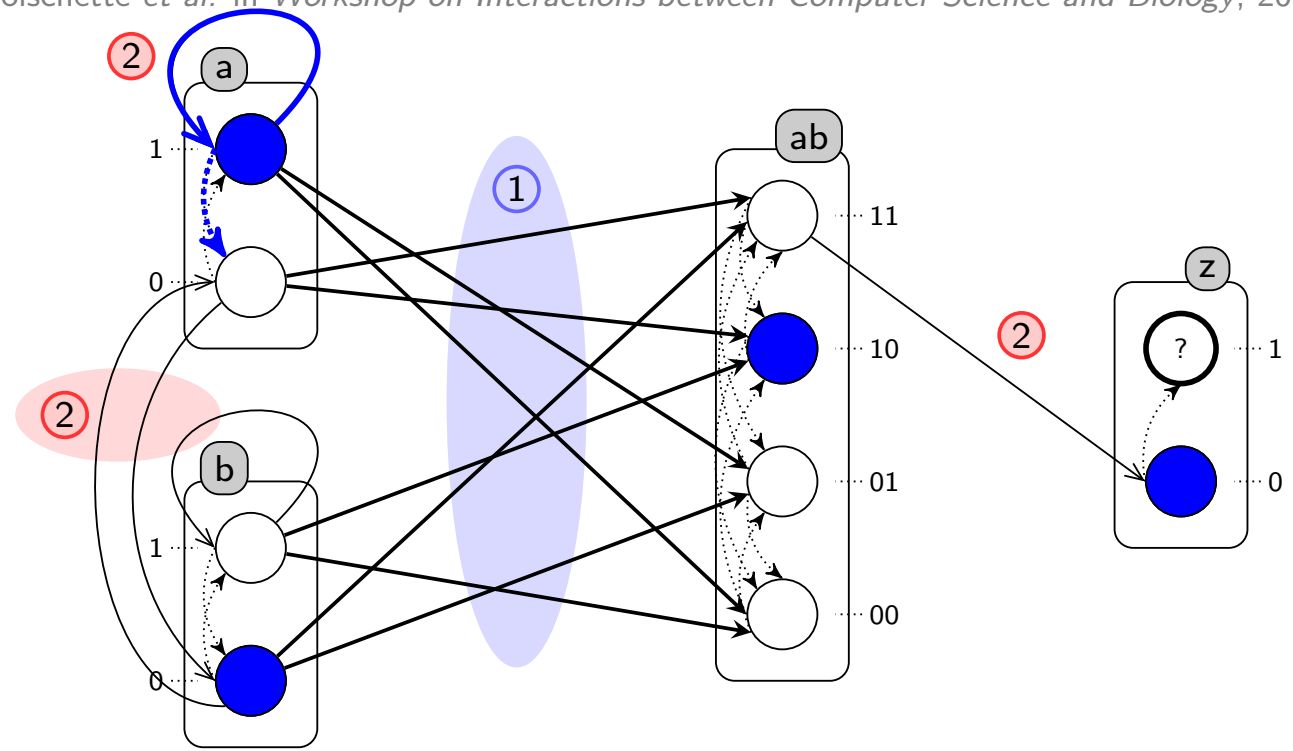
- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle$

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



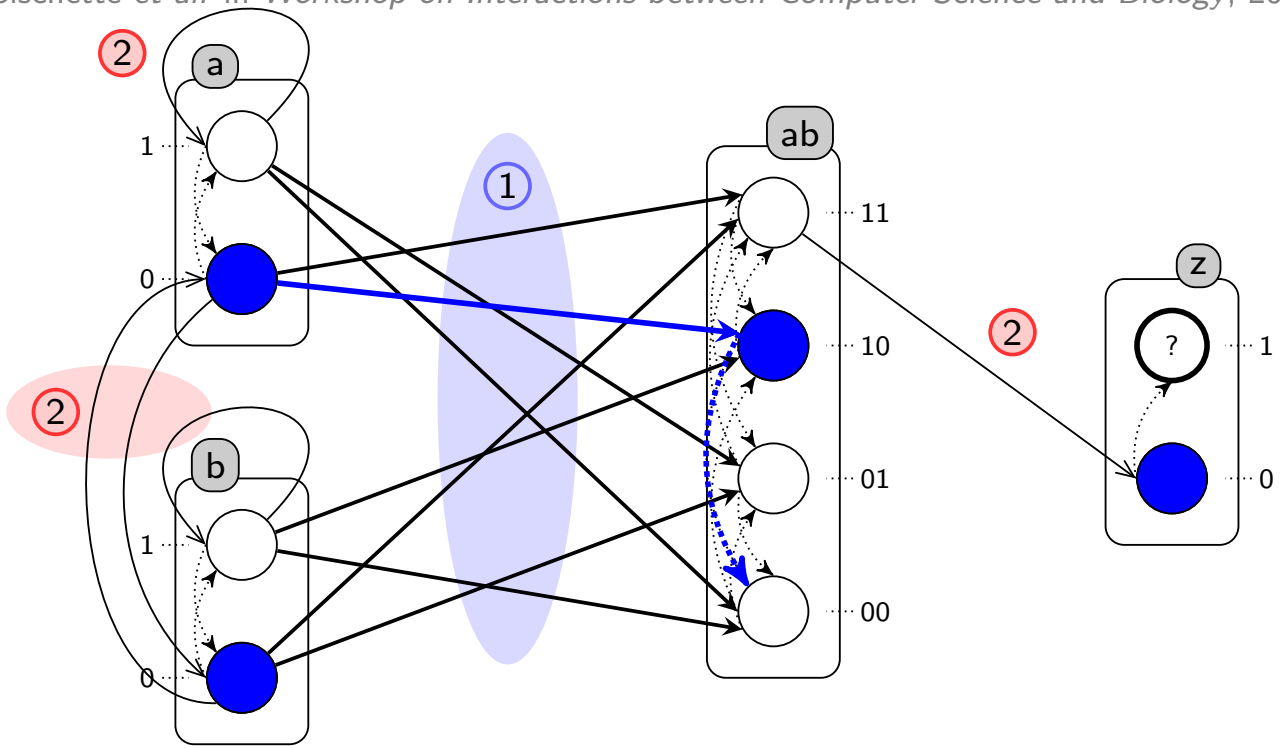
- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle$

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



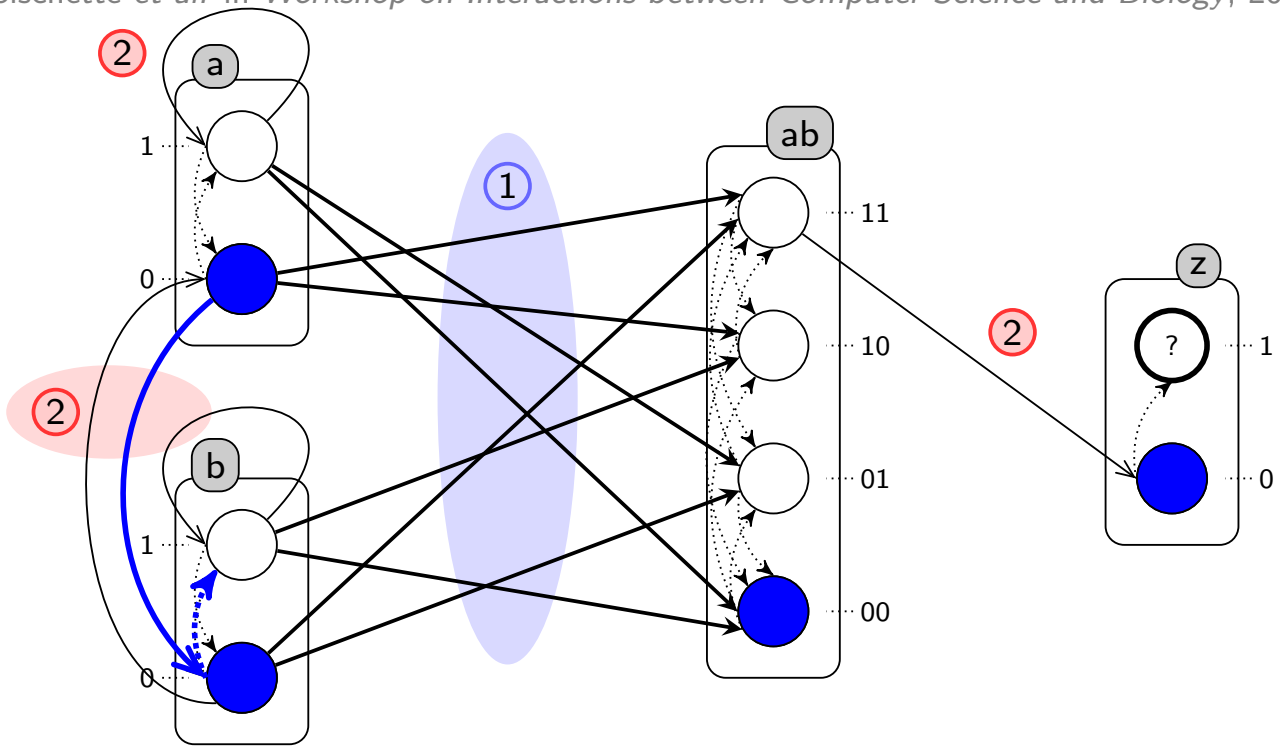
- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle$

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



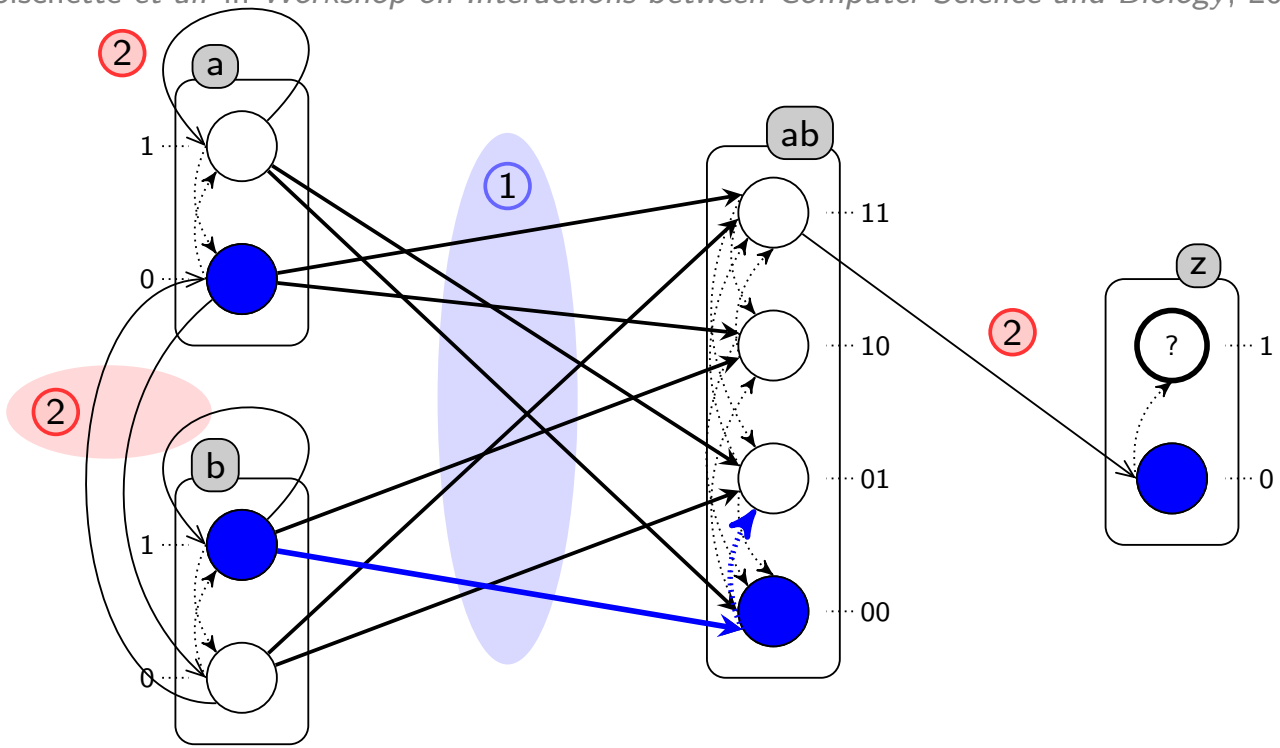
- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\rightarrow \langle a_0, b_0, ab_{00}, z_0 \rangle$

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



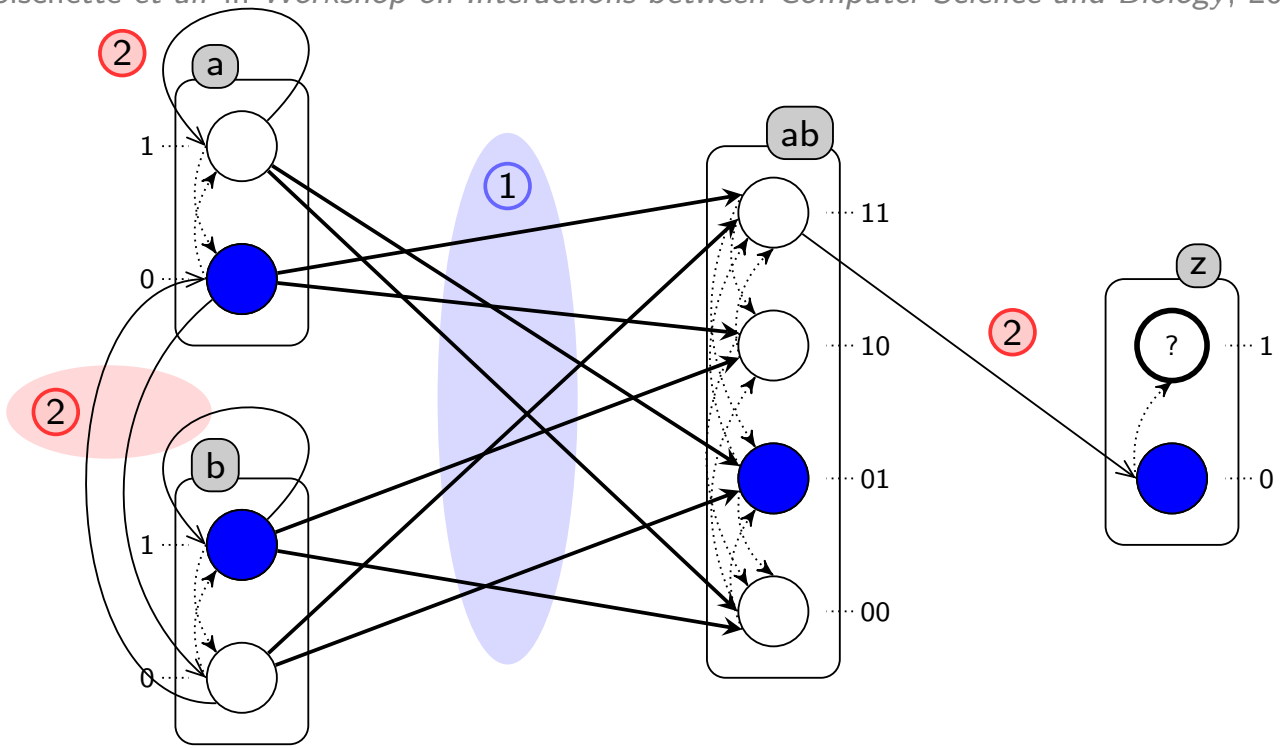
- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{10}, z_0 \rangle \to \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\to \langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_0, b_1, ab_{00}, z_0 \rangle$

Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

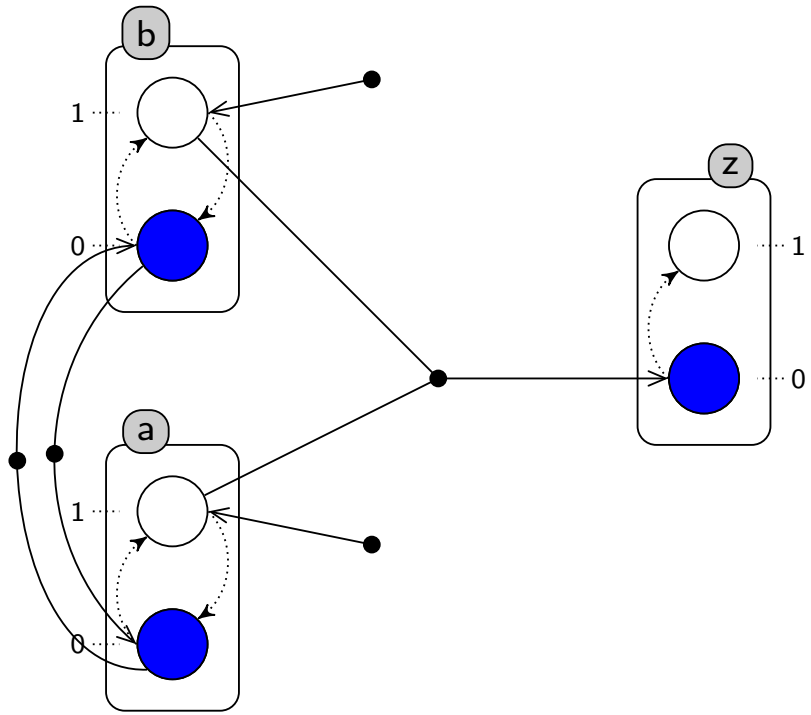


- Primary actions (updating cooperative sorts) → (1)
  non-biological / non-controllable actions
- Secondary actions (all the other ones) → (2)
  biological / controllable actions / with delays

⇒ Whenever a secondary action is played, all cooperative sorts are already updated

$\langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{10}, z_0 \rangle \to \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\to \langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_0, b_1, ab_{00}, z_0 \rangle \to \langle a_0, b_1, ab_{01}, z_0 \rangle$

Canonical Process Hitting with Synchronous Actions



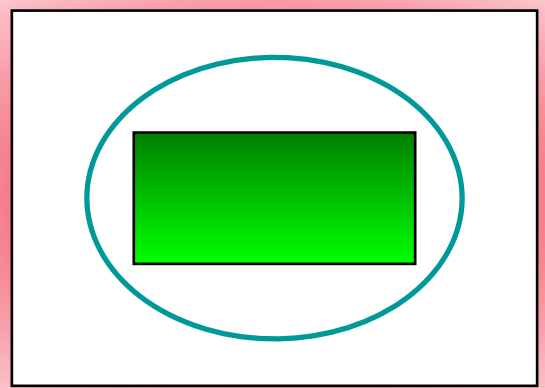
- Equivalent dynamics
- Sub-class of synchronous automata networks
- No priorities (no ill-formed model)
- No interfering updates and less intertwining

Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation

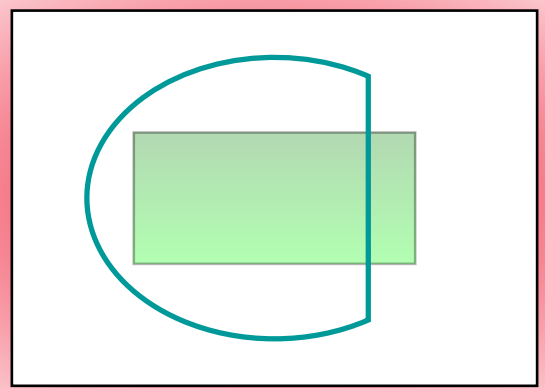


Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Adding priorities restricts the possible dynamics (preemptions)

$\rightarrow$ Invalidates the previous under-approximation

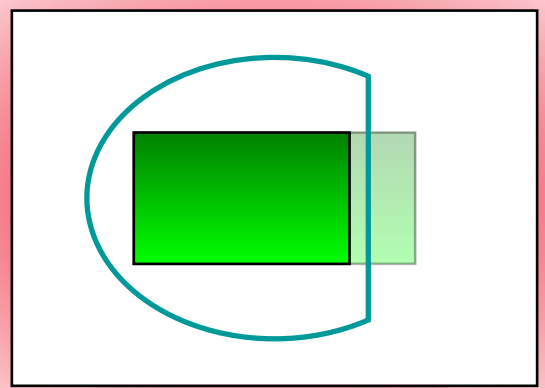


Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation

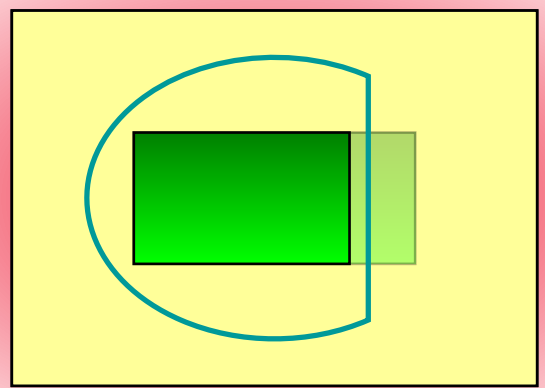


Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation

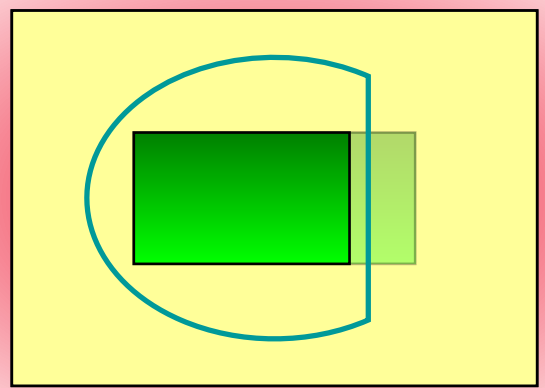


Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Adding priorities restricts the possible dynamics (preemptions)

→ Invalidates the previous under-approximation



Similar complexity for a more expressive formalism

→ Still efficient for big models

→ Finer under-approximation

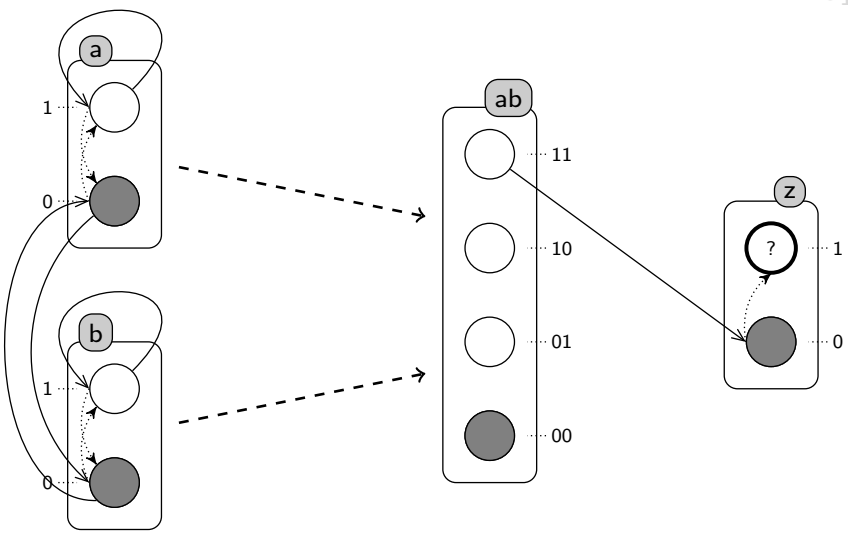
Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive



$\boxed{z_1}$	Required process
$z_0 \Uparrow^* z_1$	Objective
○	Solution to an objective
$\boxed{\{a_1, b_1\}}$	Synchronization of several processes

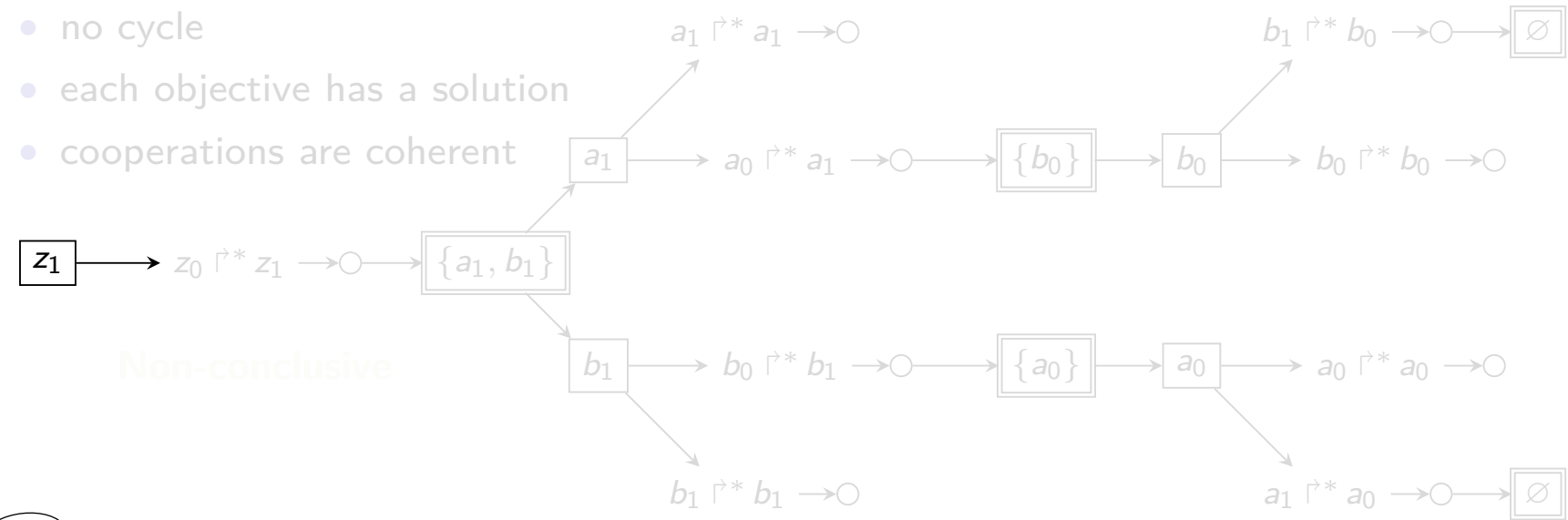
Static Analysis of Canonical Process Hitting

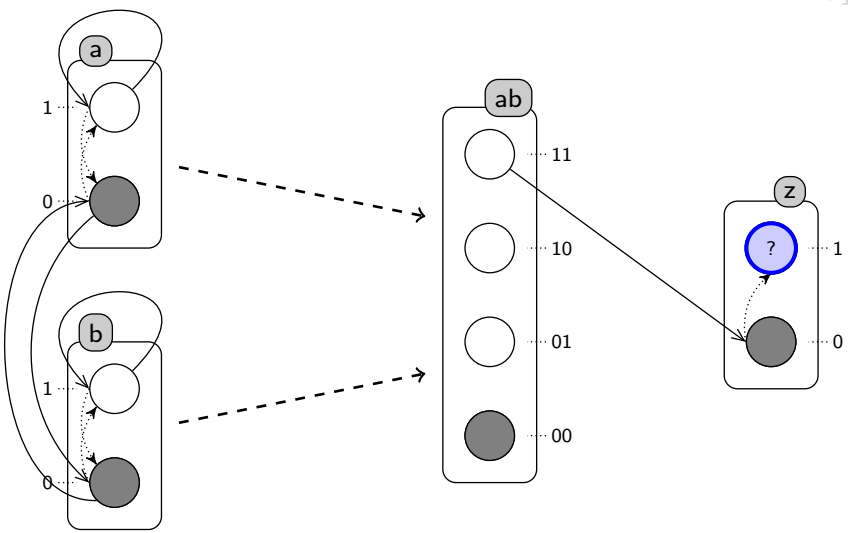
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive





$\boxed{z_1}$	Required process
$z_0 \Rsh^* z_1$	Objective
○	Solution to an objective
$\boxed{\{a_1, b_1\}}$	Synchronization of several processes

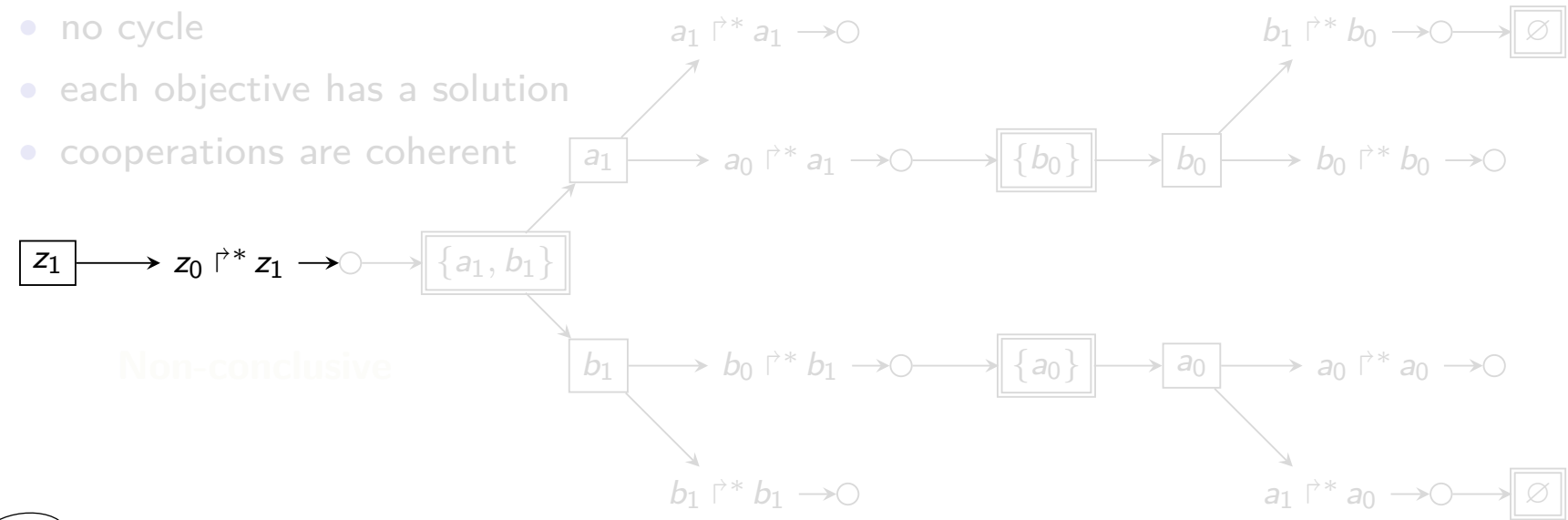
Static Analysis of Canonical Process Hitting

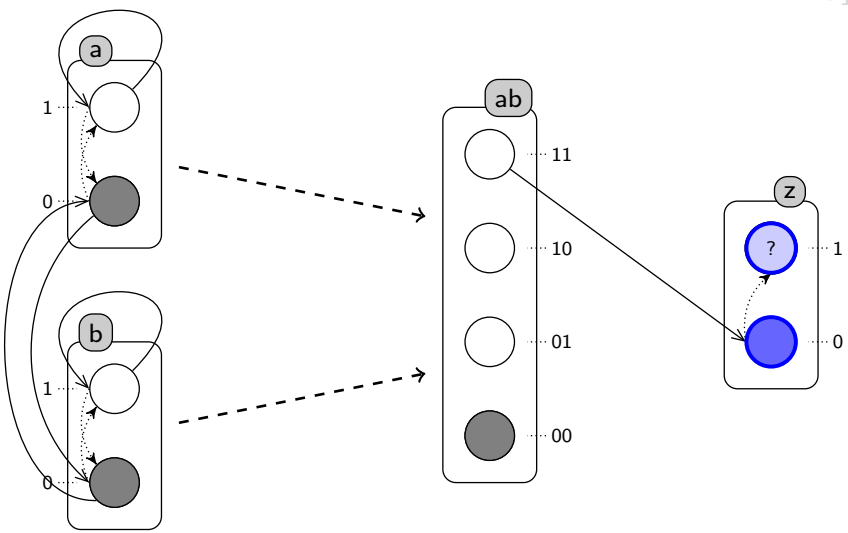
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive





z_1	Required process
$z_0 \Rightarrow^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

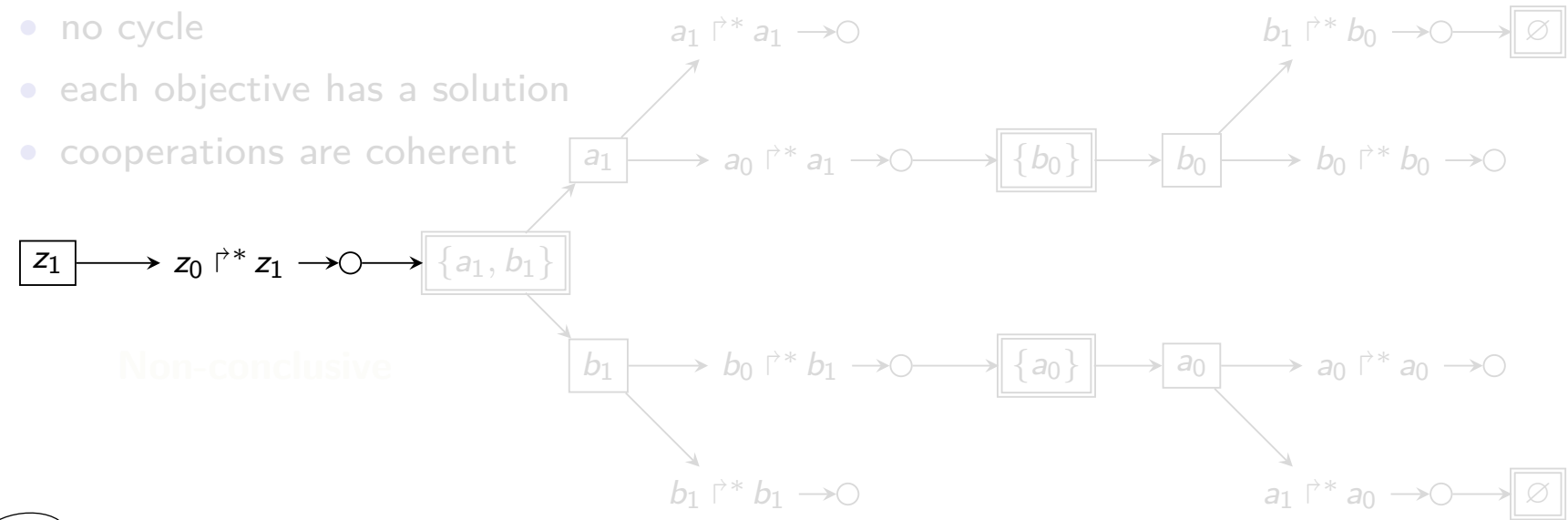
Static Analysis of Canonical Process Hitting

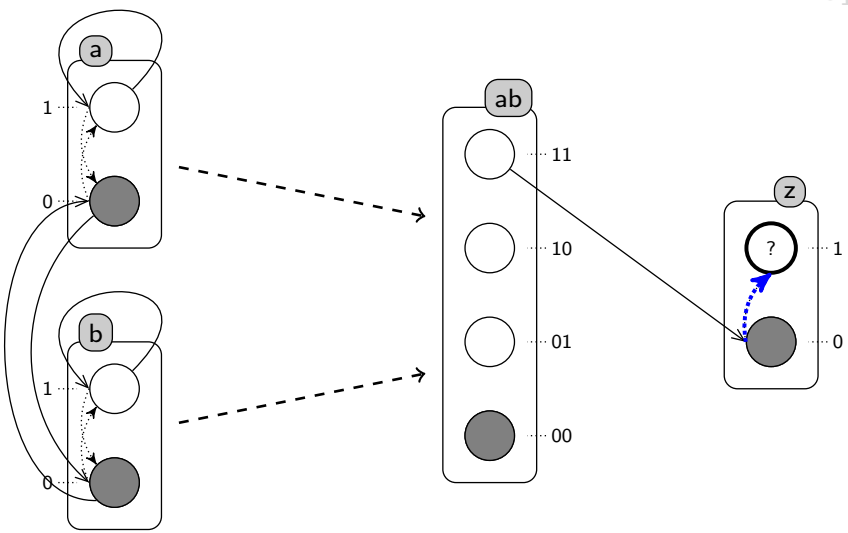
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive





z_1	Required process
$z_0 \Rsh^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

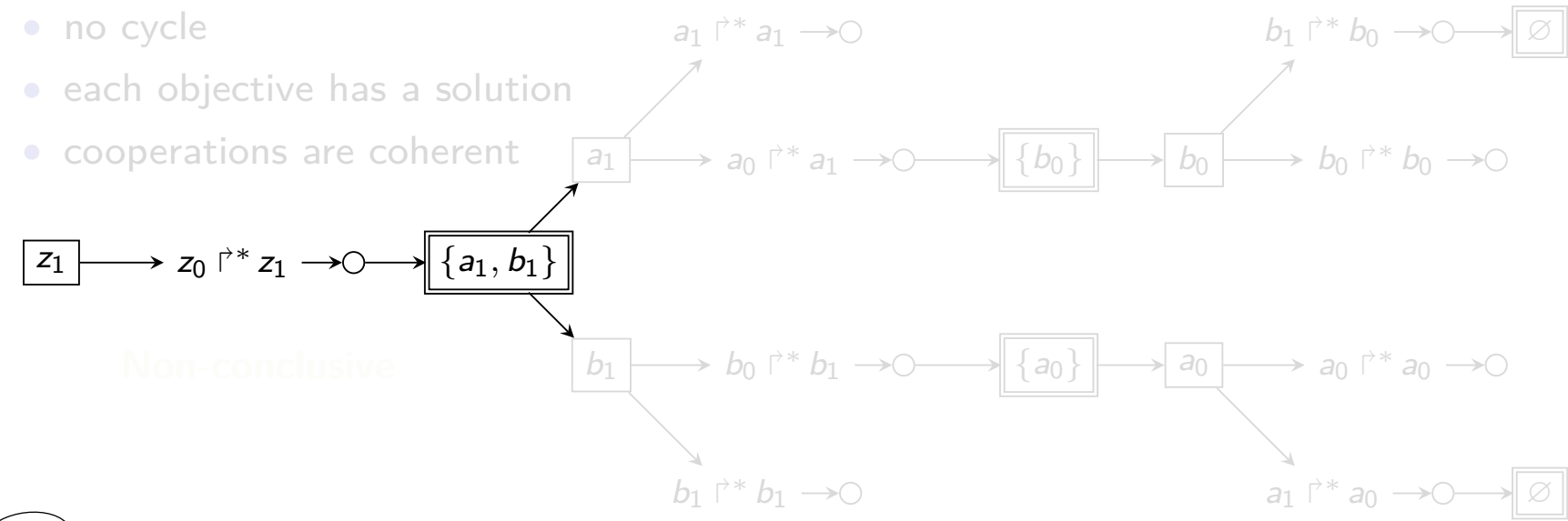
Static Analysis of Canonical Process Hitting

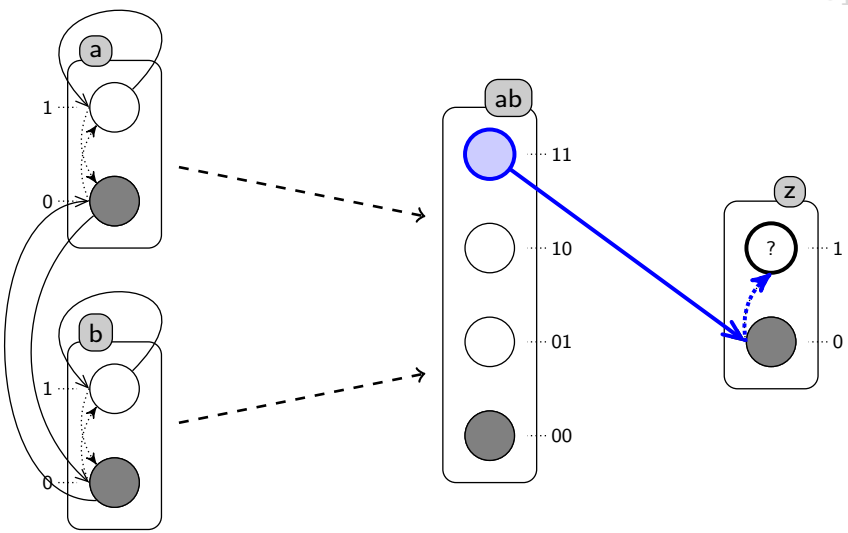
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive





z_1	Required process
$z_0 \Rsh^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

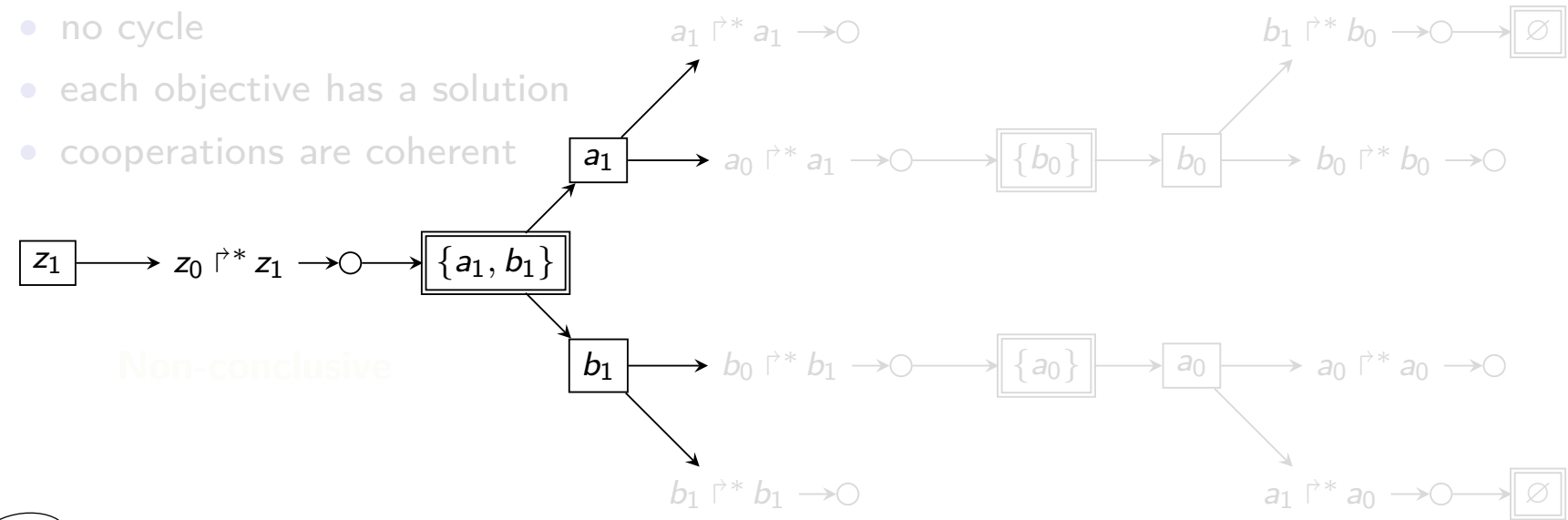
Static Analysis of Canonical Process Hitting

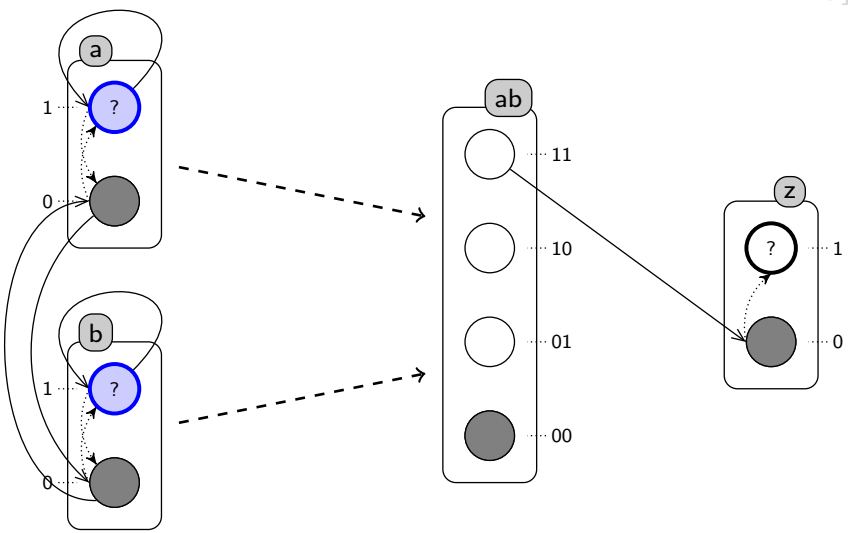
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive





z_1	Required process
$z_0 \upharpoonright^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

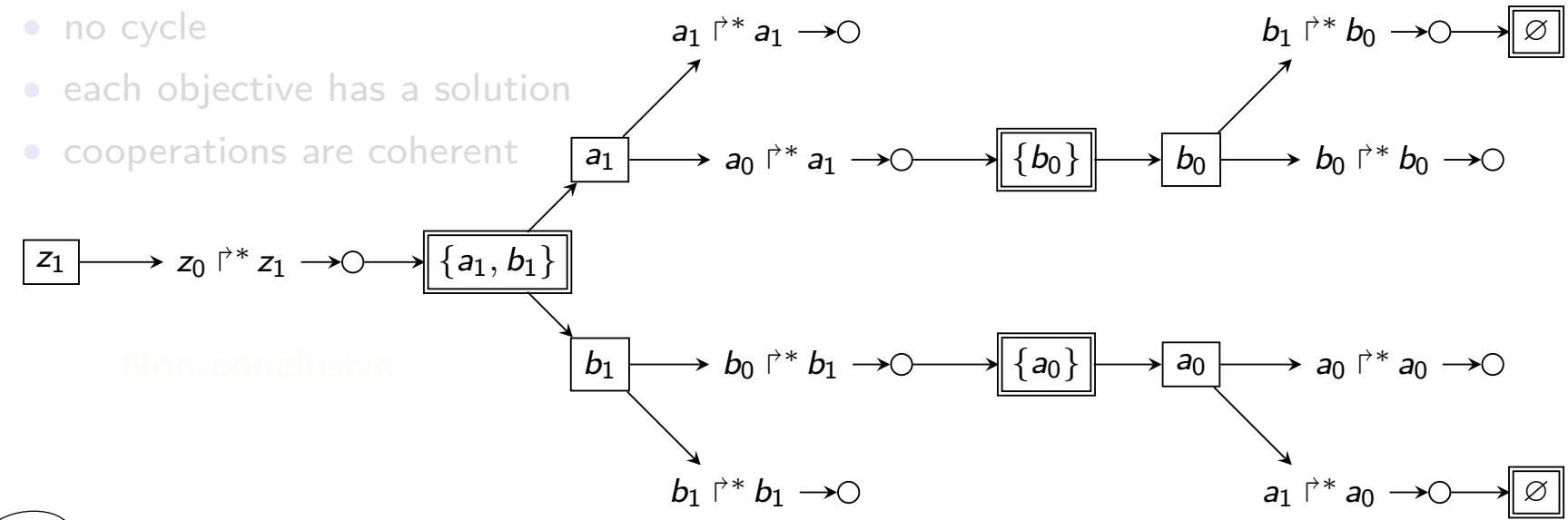
Static Analysis of Canonical Process Hitting

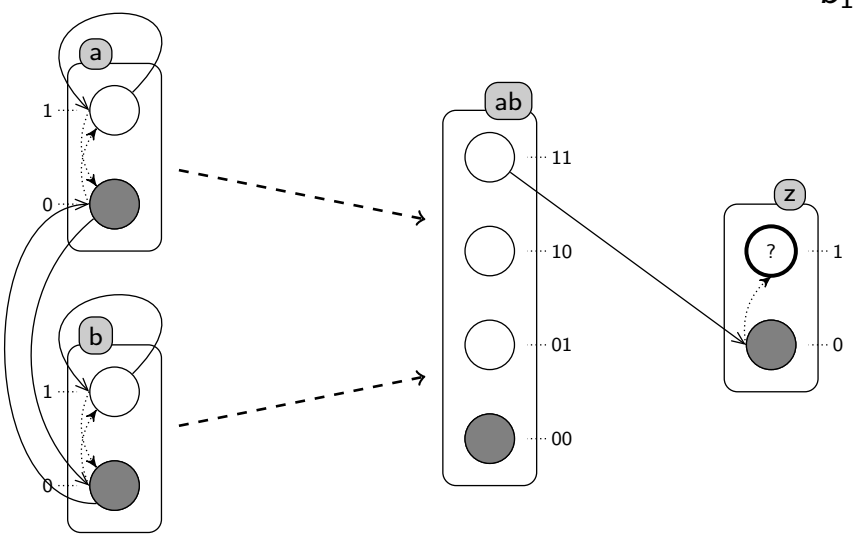
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent

Non-conclusive





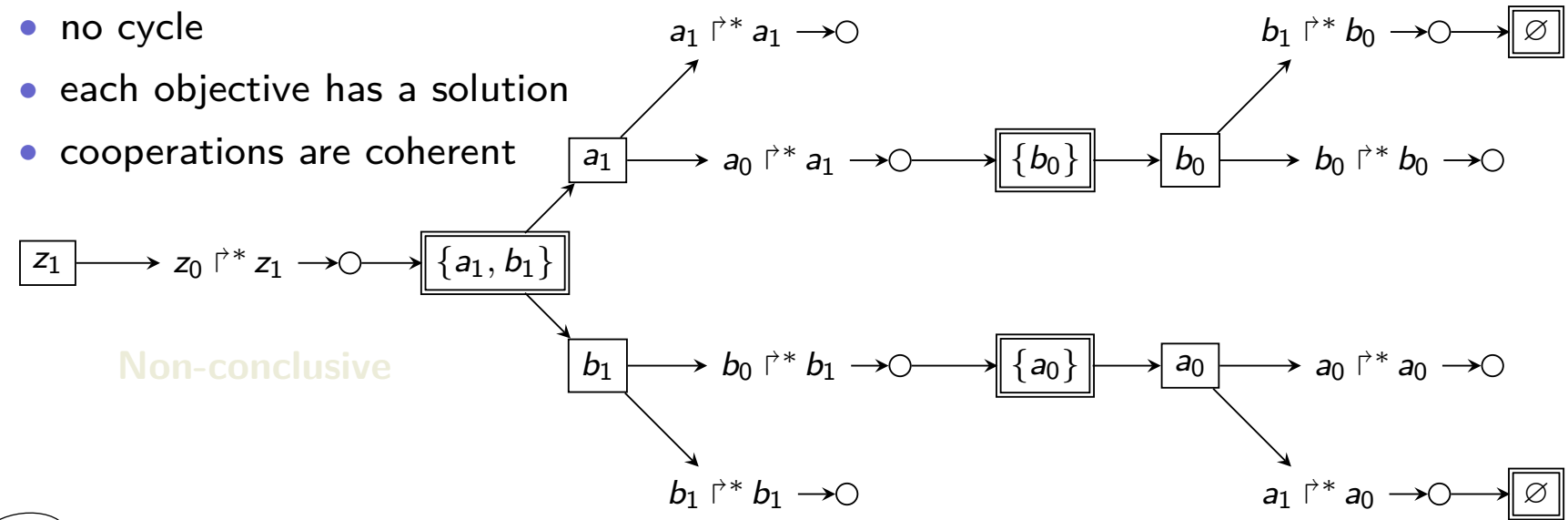
z_1	Required process
$z_0 \Rsh^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

Static Analysis of Canonical Process Hitting

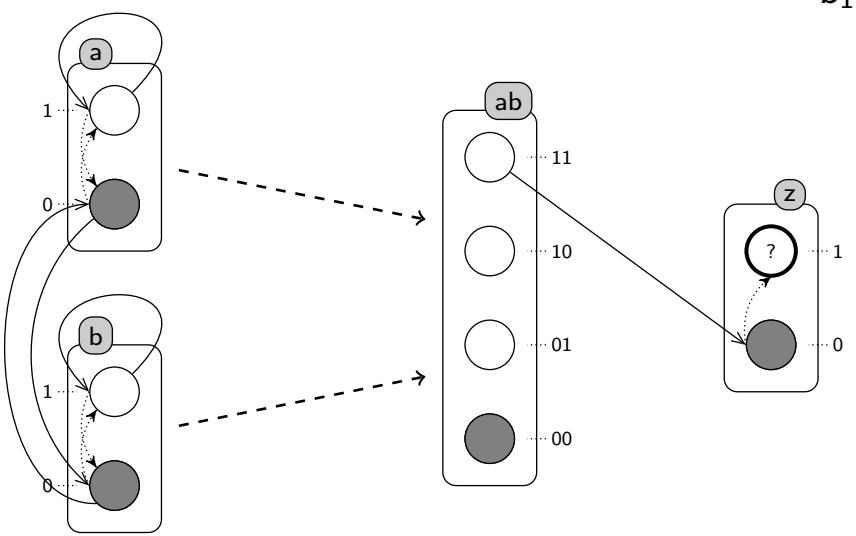
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent



Non-conclusive



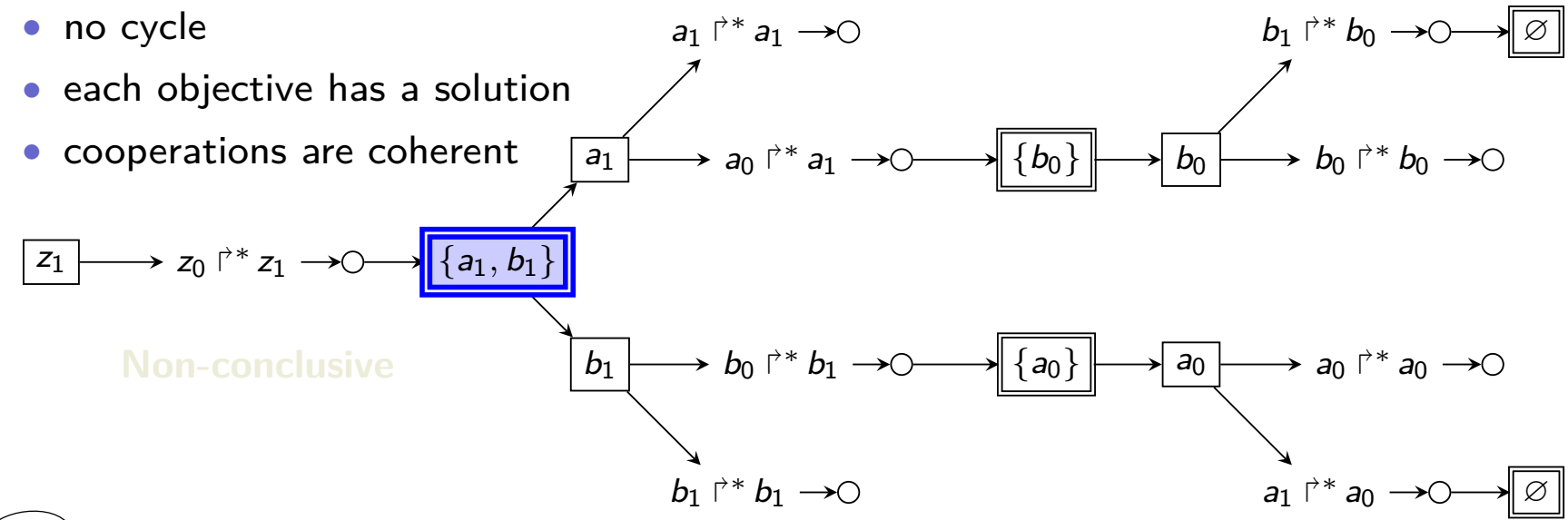
z_1	Required process
$z_0 \upharpoonright^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

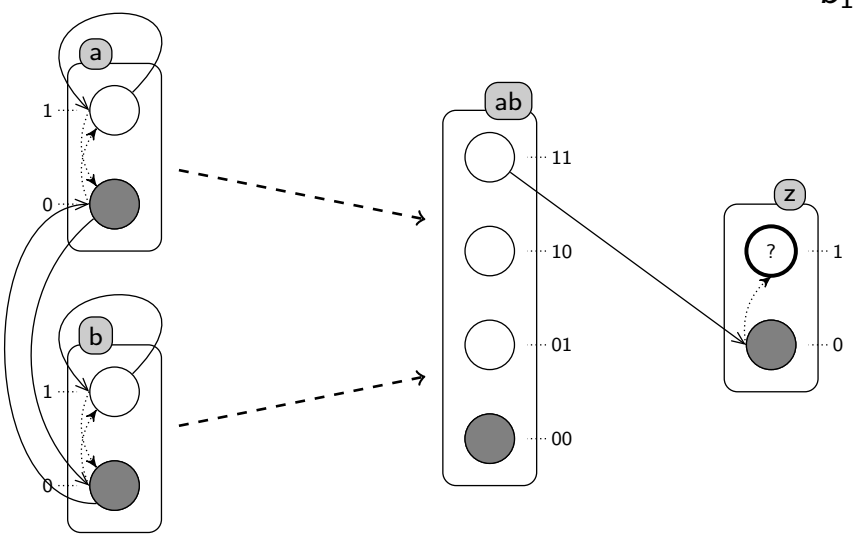
Static Analysis of Canonical Process Hitting

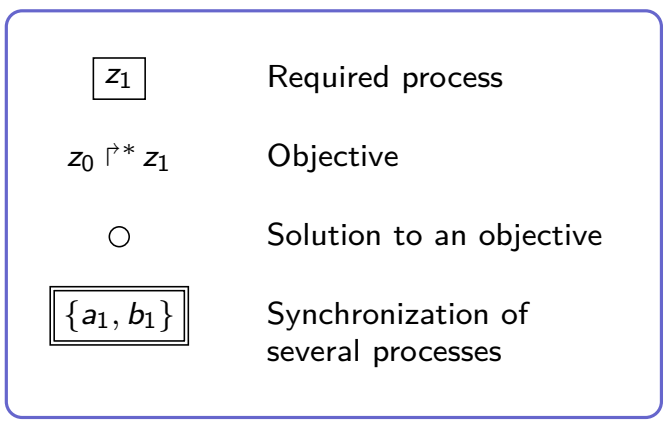
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent





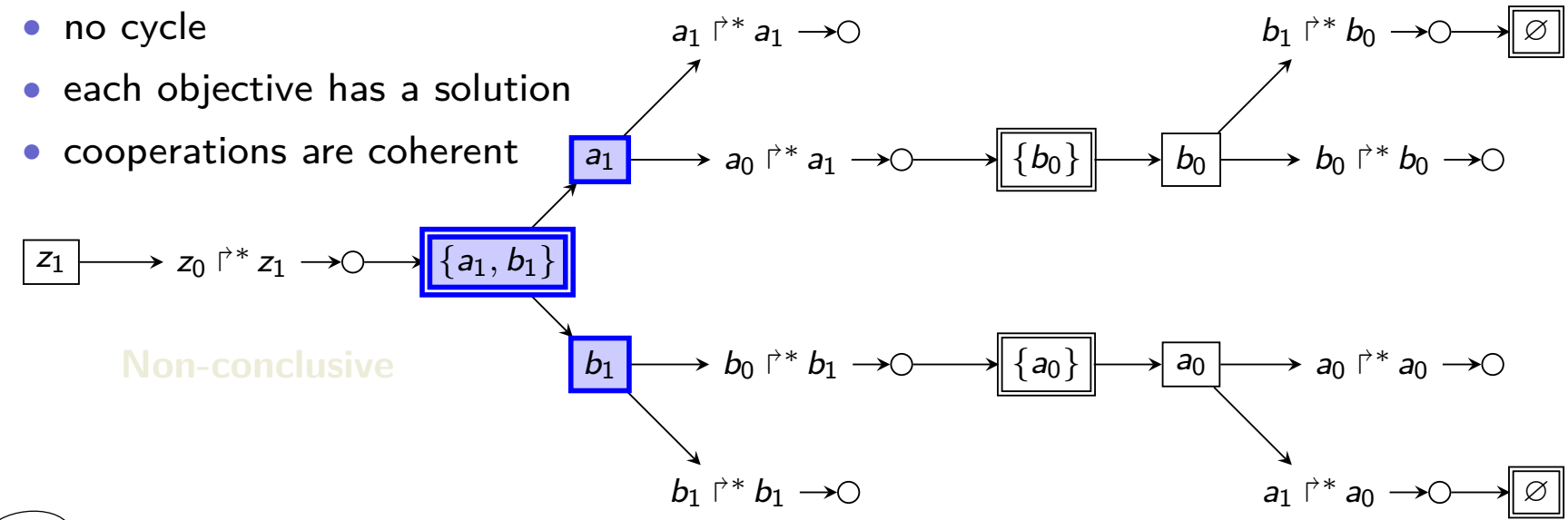


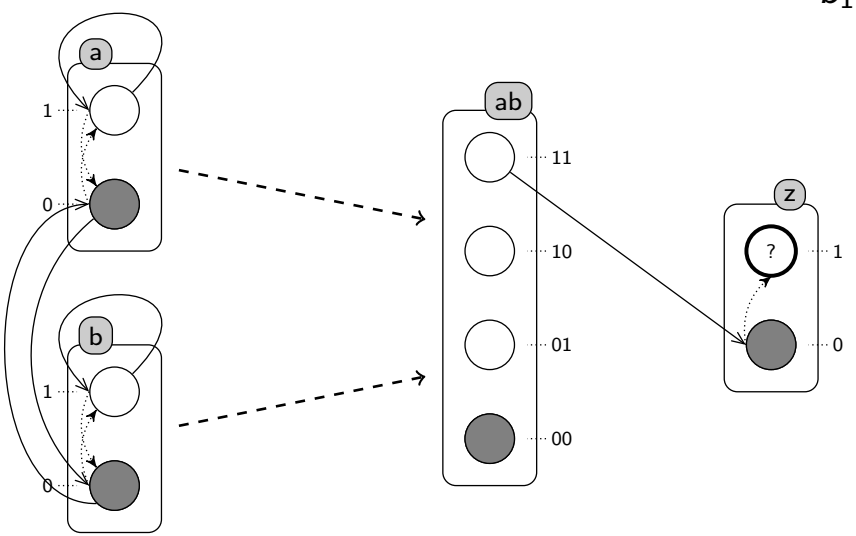
Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent





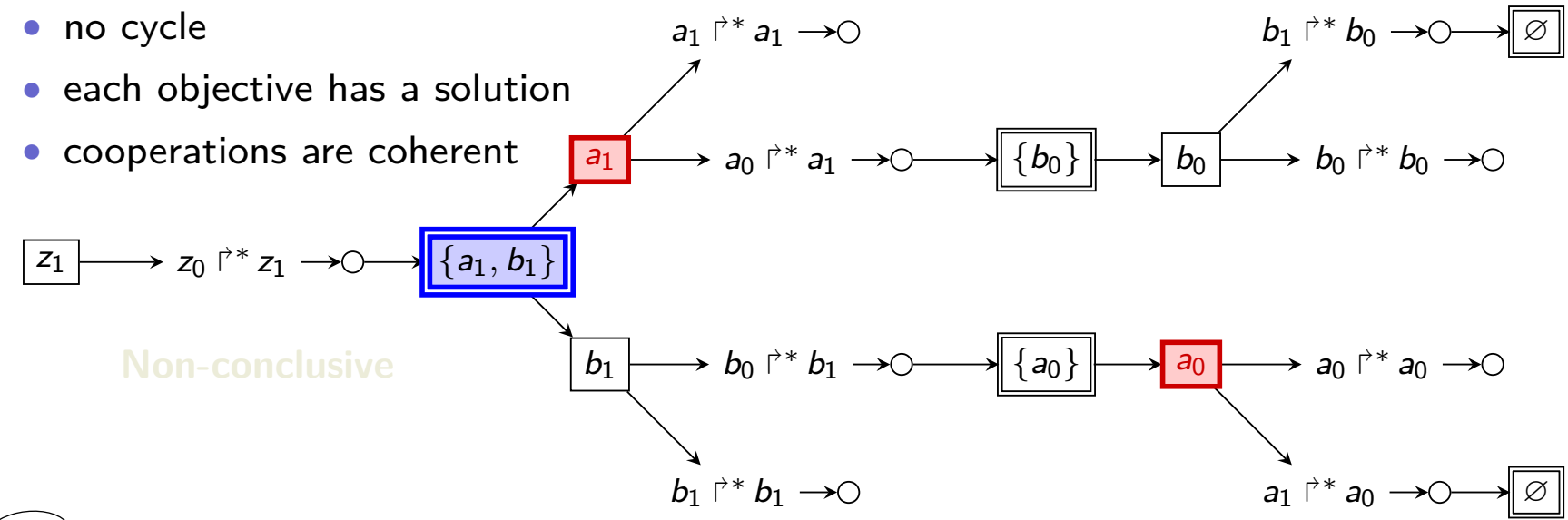
z_1	Required process
$z_0 \Rightarrow^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

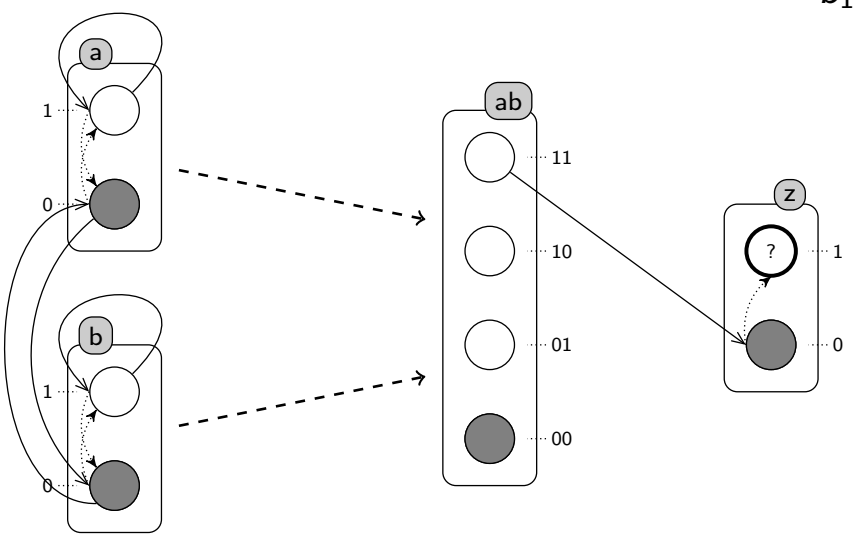
Static Analysis of Canonical Process Hitting

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- cooperations are coherent





z_1	Required process
$z_0 \Rightarrow^* z_1$	Objective
○	Solution to an objective
$\{a_1, b_1\}$	Synchronization of several processes

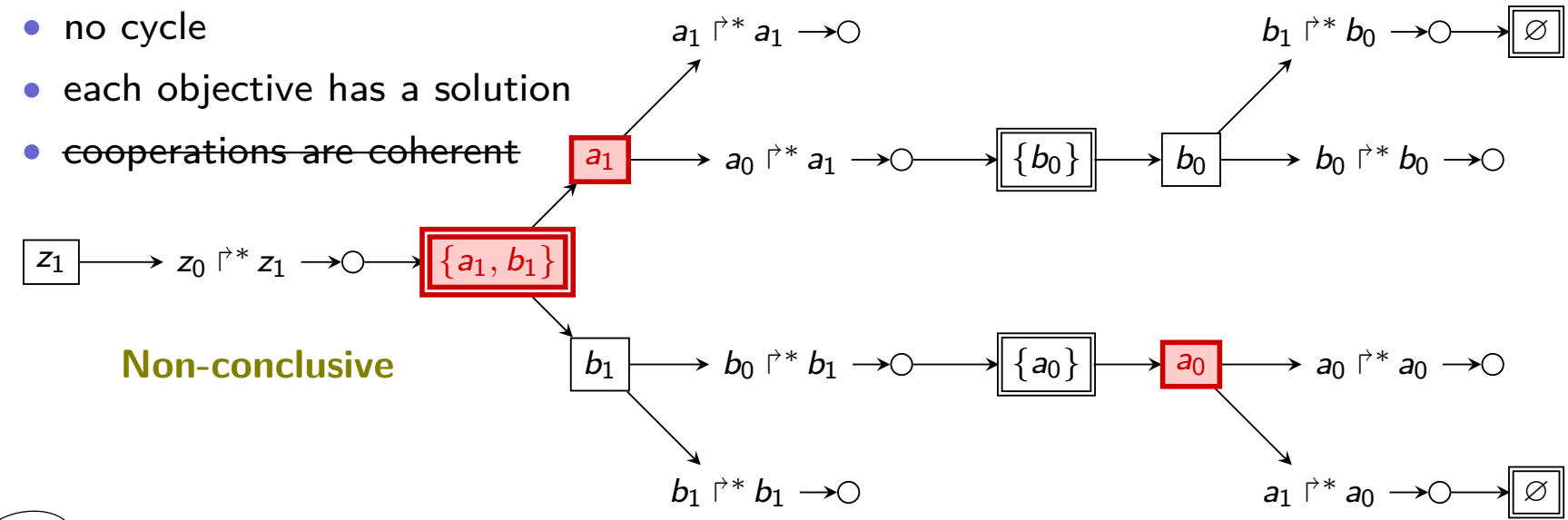
Static Analysis of Canonical Process Hitting

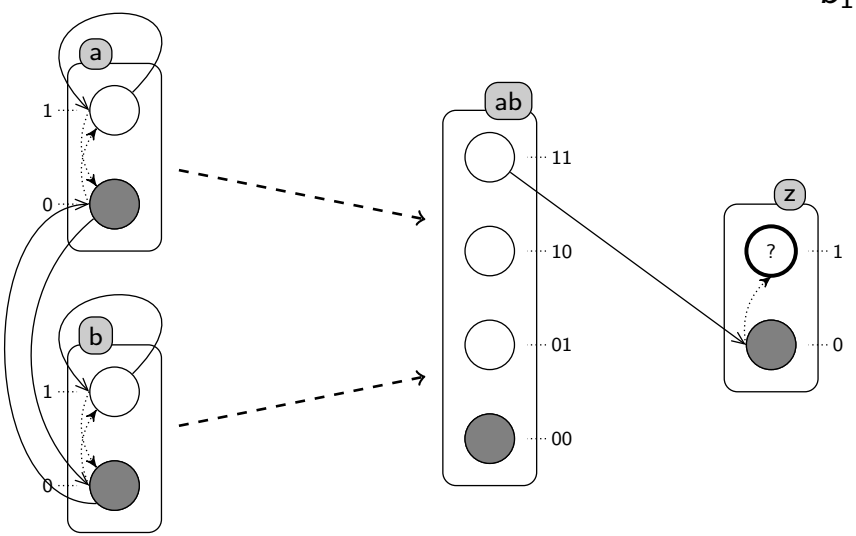
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

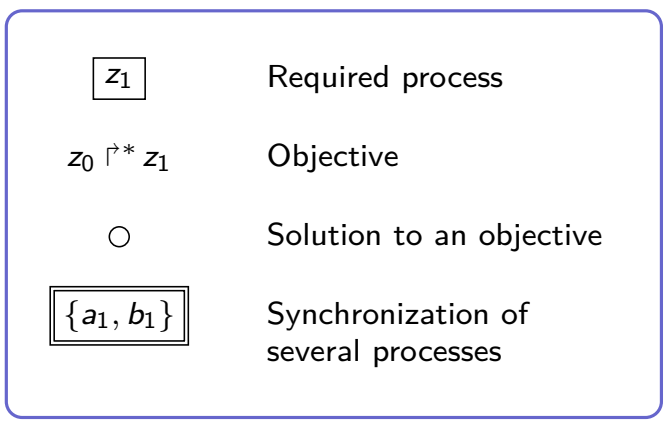
Sufficient condition:

- no cycle
- each objective has a solution
- ~~cooperations are coherent~~

Non-conclusive







Implementation of the Static Analysis Into PINT

Complexity:

- Computation of the local causality graph:
  - Polynomial in the number of sorts
  - Exponential in the number of processes of each sort
- Analysis of the graph (sufficient condition):
  - Polynomial in the size of the graph

Implementation of the Static Analysis Into PINT

Complexity:

- Computation of the local causality graph:
  - Polynomial in the number of sorts
  - Exponential in the number of processes of each sort
- Analysis of the graph (sufficient condition):
  - Polynomial in the size of the graph

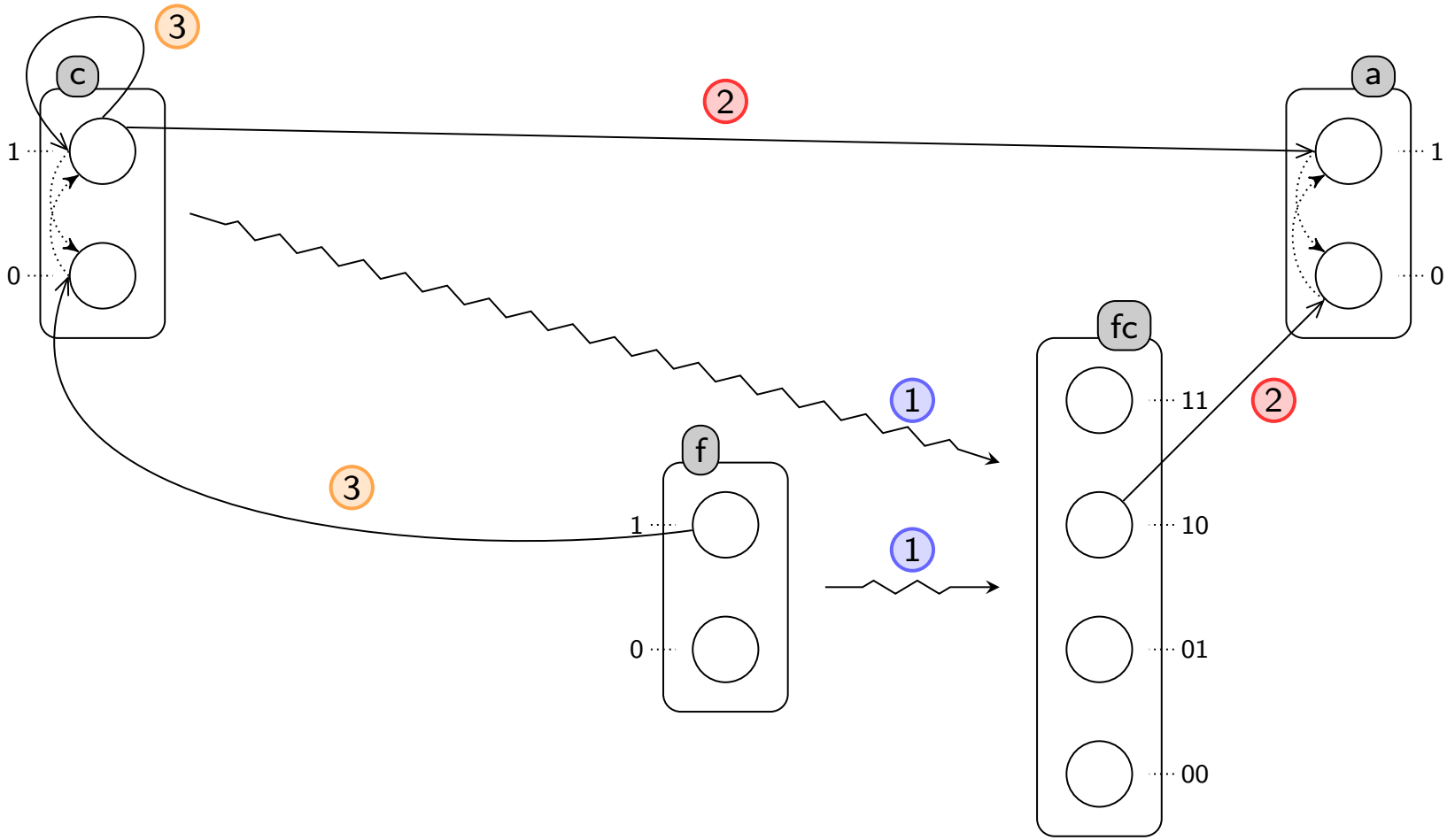
Makes the study of large networks tractable:

Modèle	Sortes	Processus	Actions	États	libddd[1]	GINsim[2]	PINT
egfr20	35	196	670	2^{64}		$<$1s	**0.02s**
tcrsig40	54	156	301	2^{73}		∞	**0.02s**
tcrsig94	133	448	1124	2^{194}	[13min – ∞]		**0.03s**
egfr104	193	748	2356	2^{320}			**0.16s**

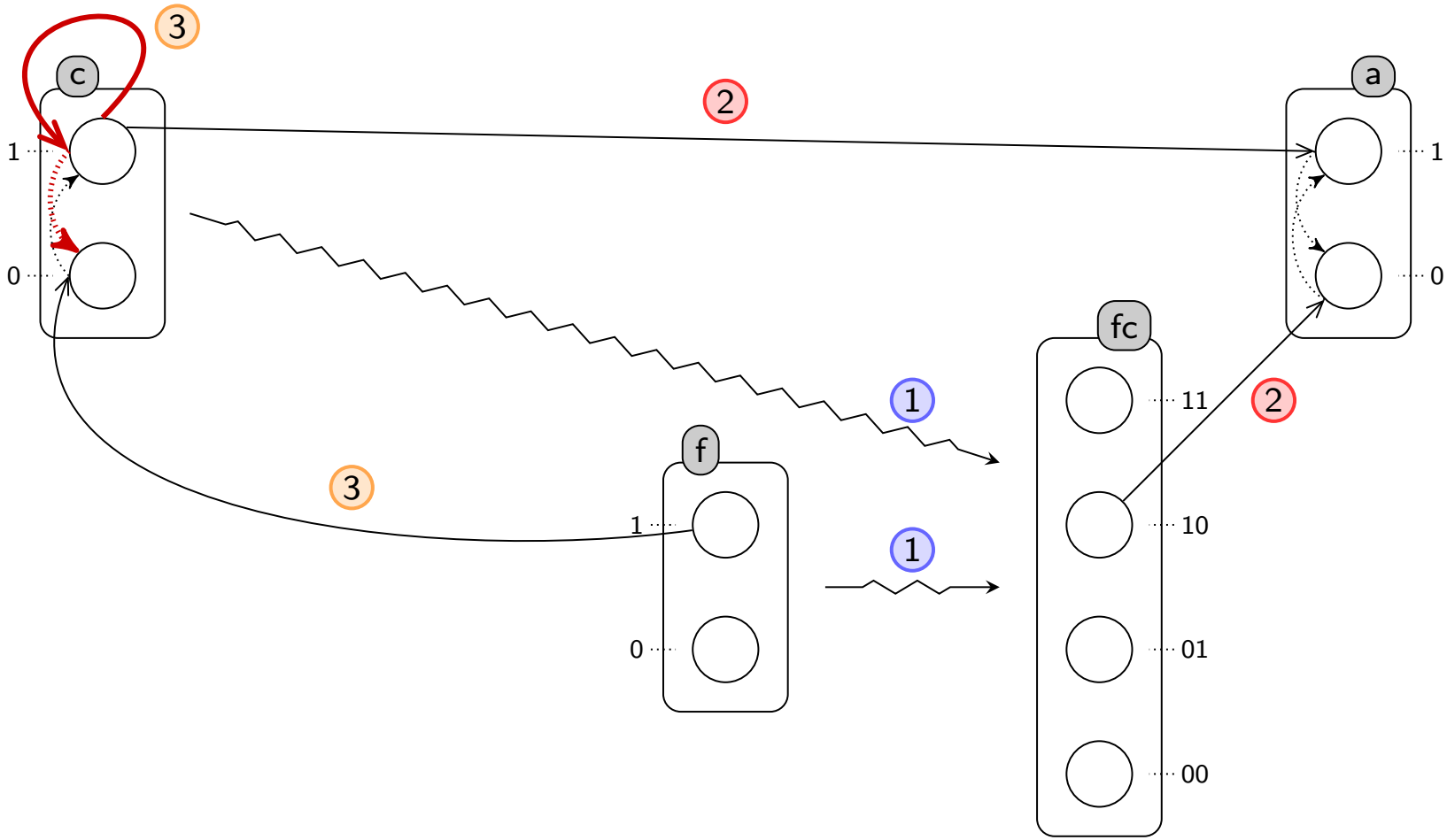
[1] LIP6/Move [Couvreur *et al.*, *Lecture Notes in Computer Science*, 2002]
[2] TAGC/IGC [Chaouiya, Naldi, Thieffry, *Methods in Molecular Biology*, 2012]

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]

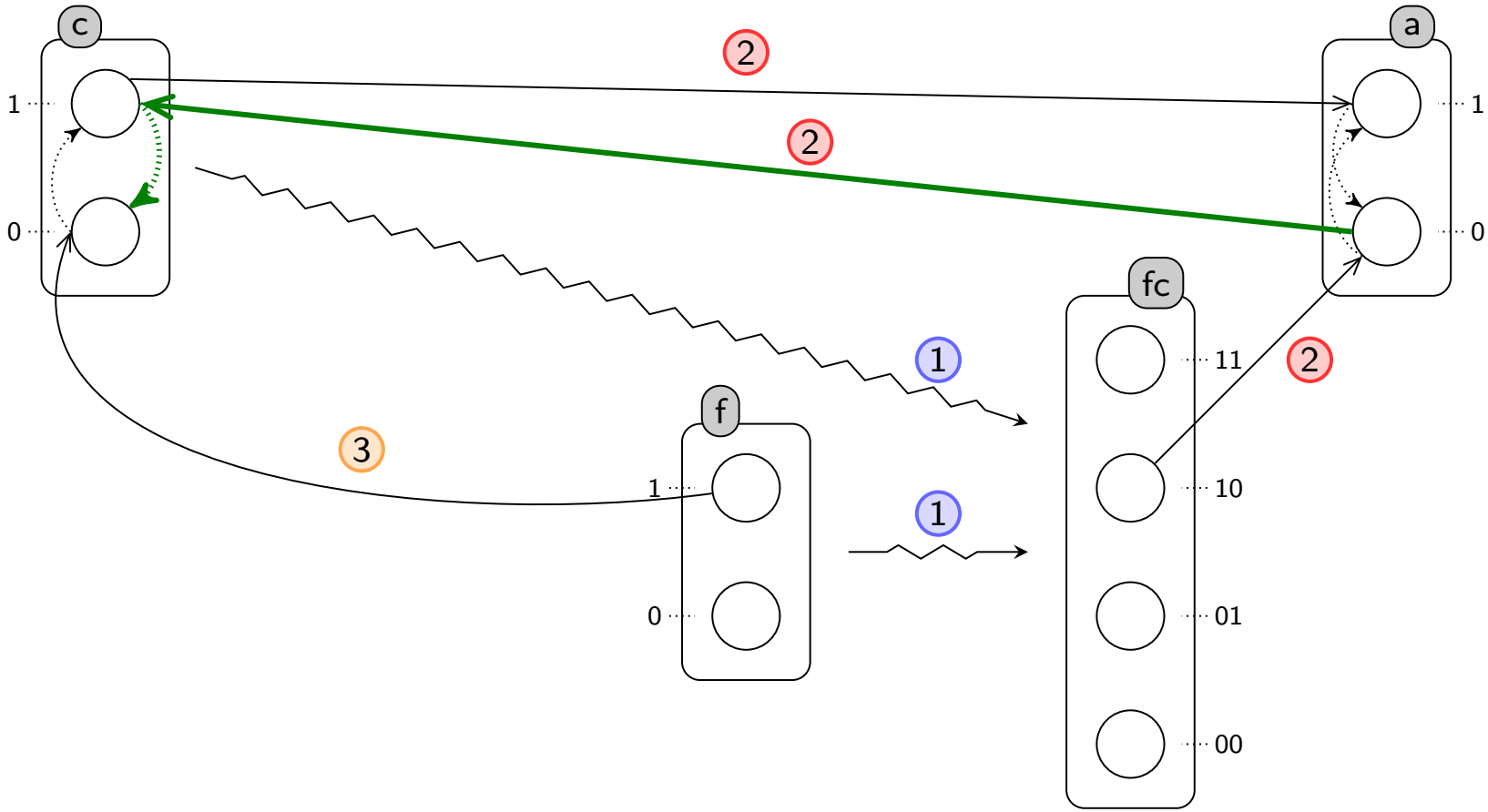
Formal Translation Into Canonical Form



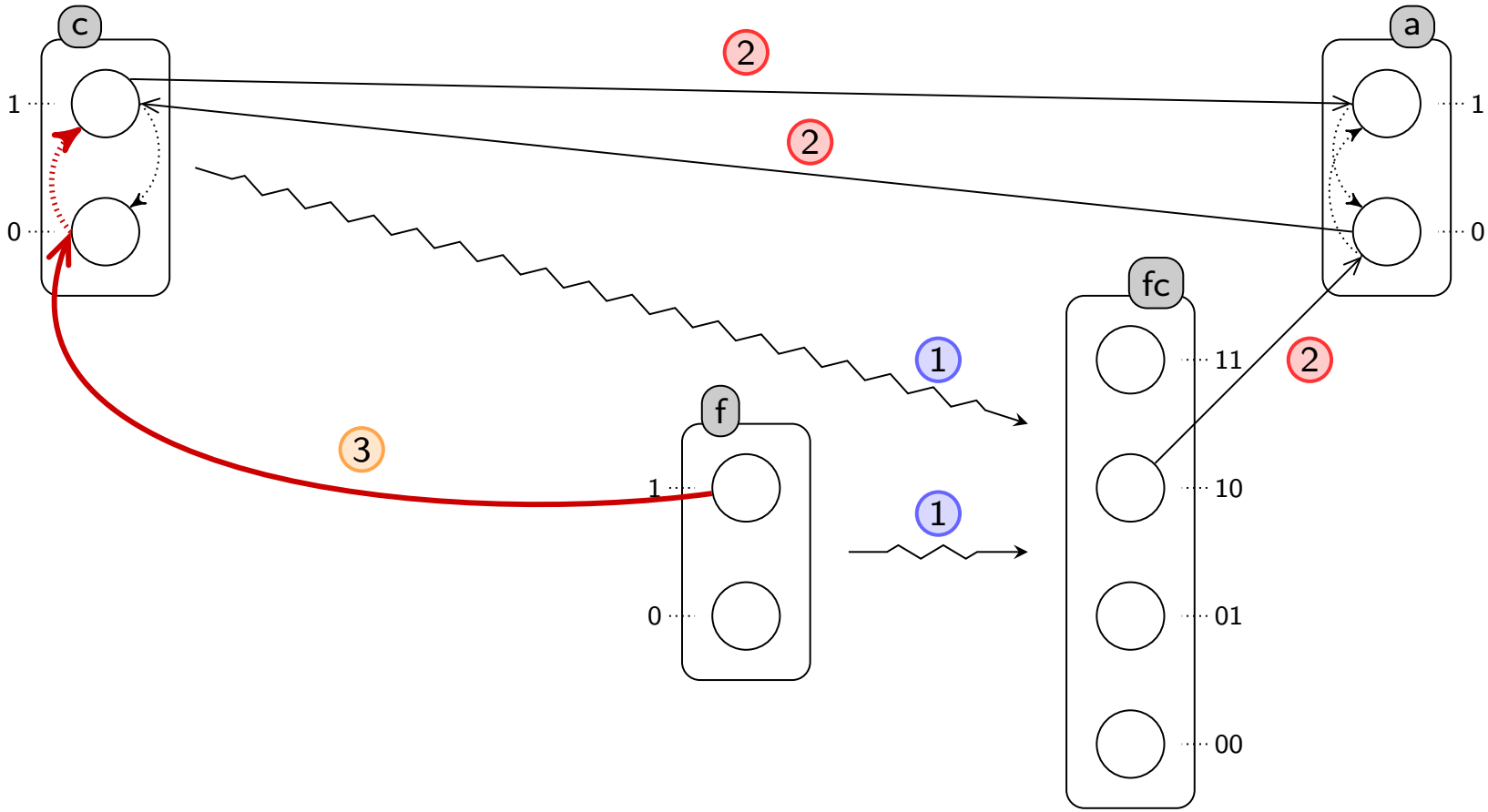
Formal Translation Into Canonical Form



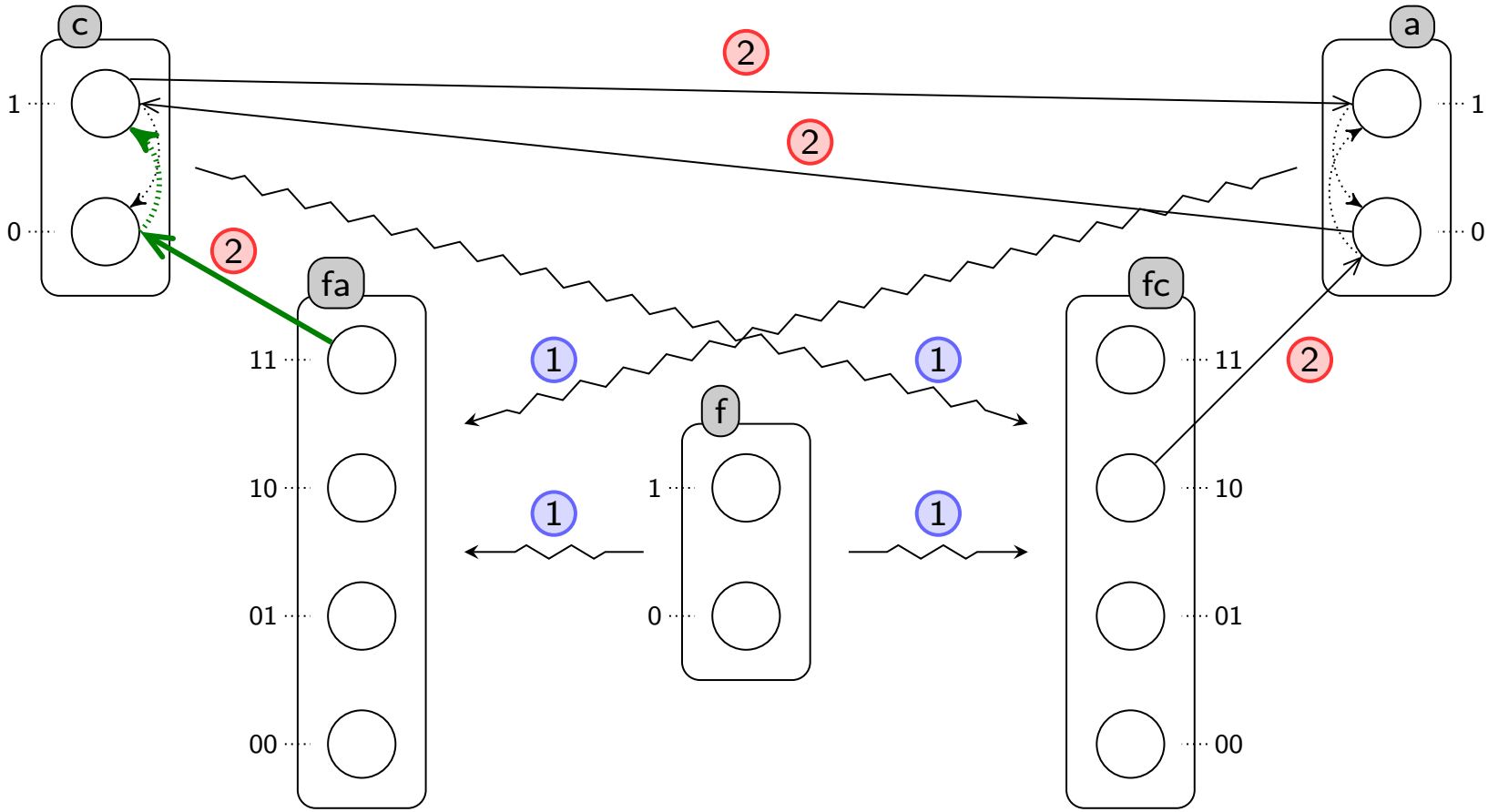
Formal Translation Into Canonical Form



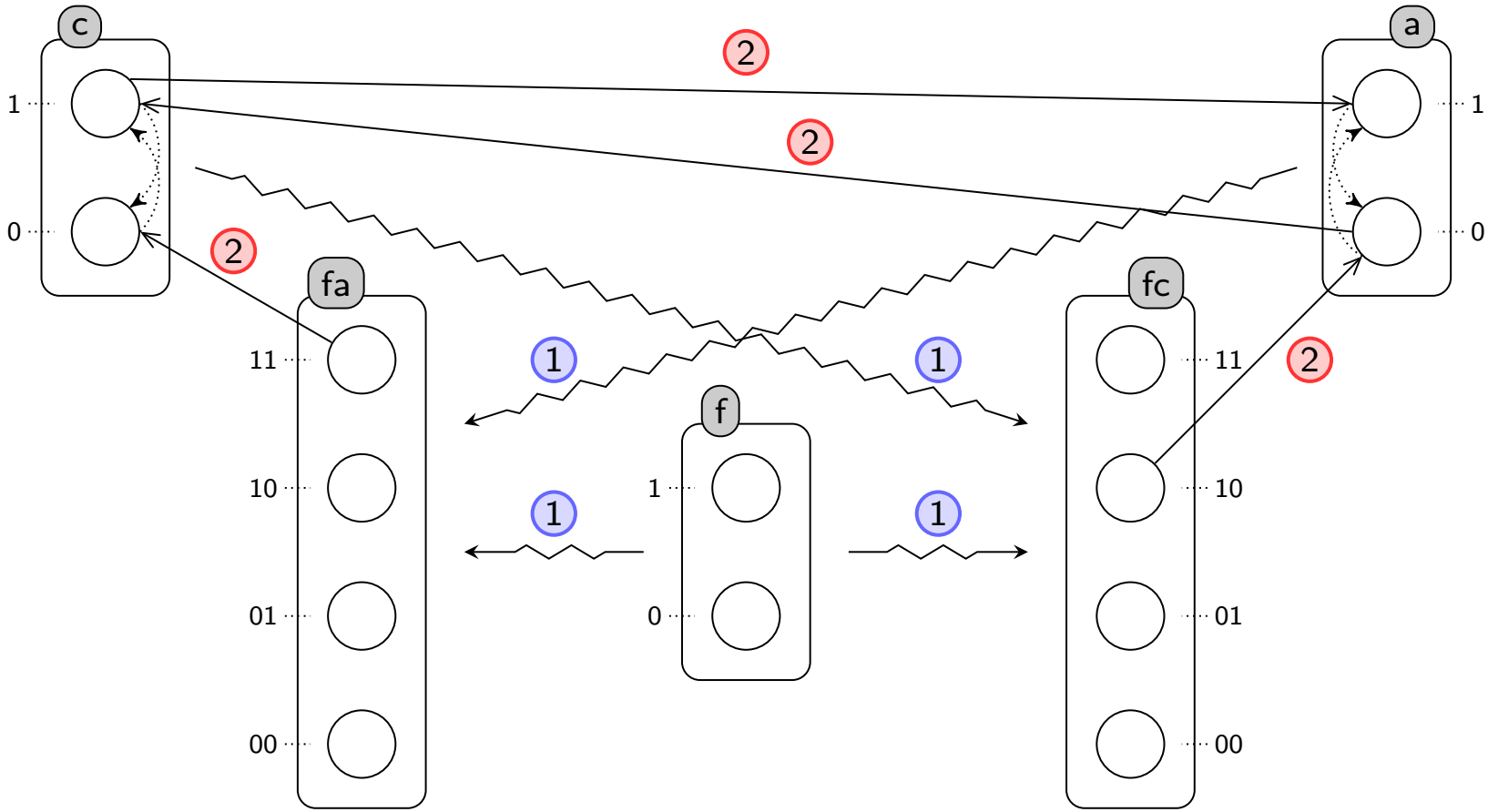
Formal Translation Into Canonical Form



Formal Translation Into Canonical Form



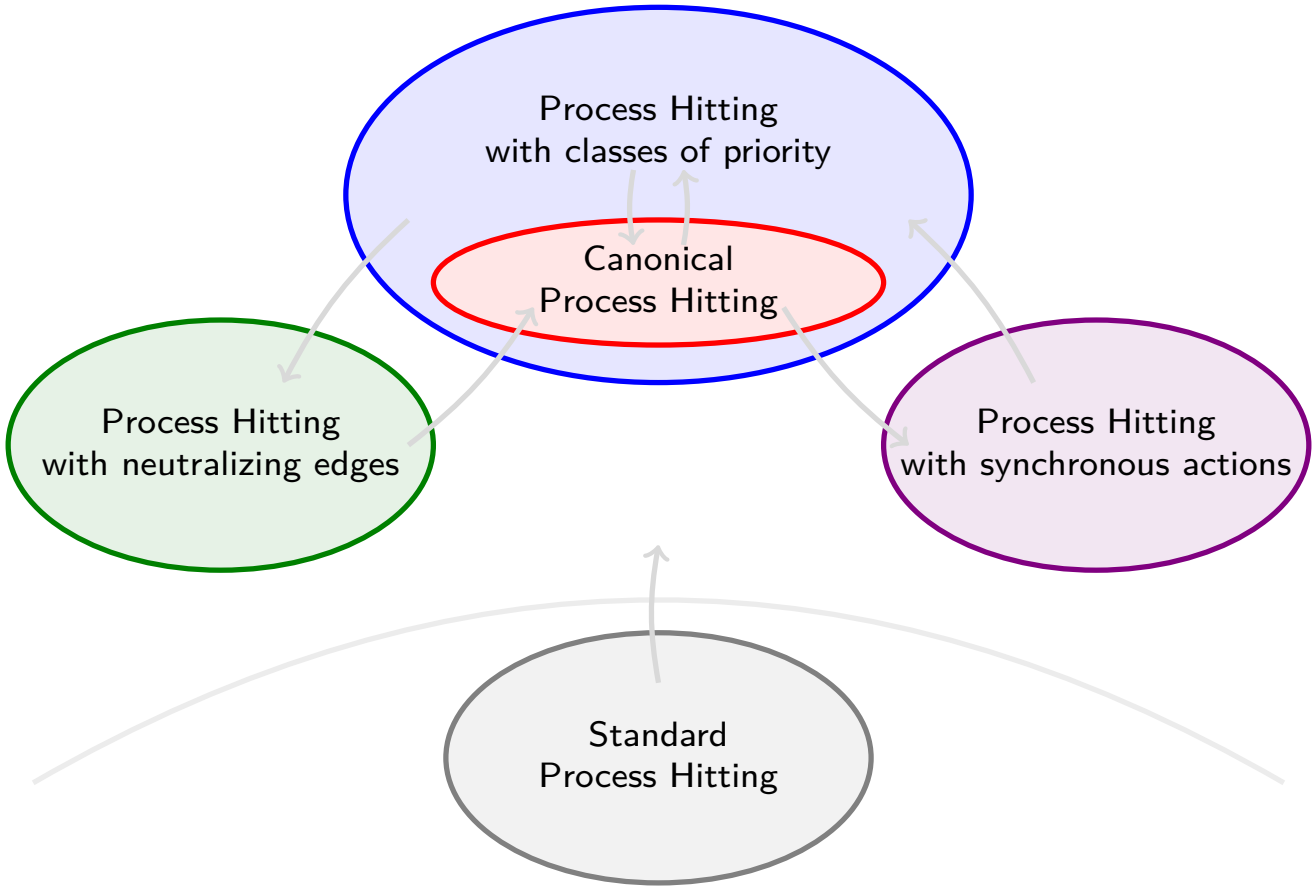
Formal Translation Into Canonical Form



→ Same dynamics (with supplemental cooperative sorts)

→ The canonical form can be computed for all Process Hitting extensions, with classes of priorities, neutralizing edges or synchronous actions

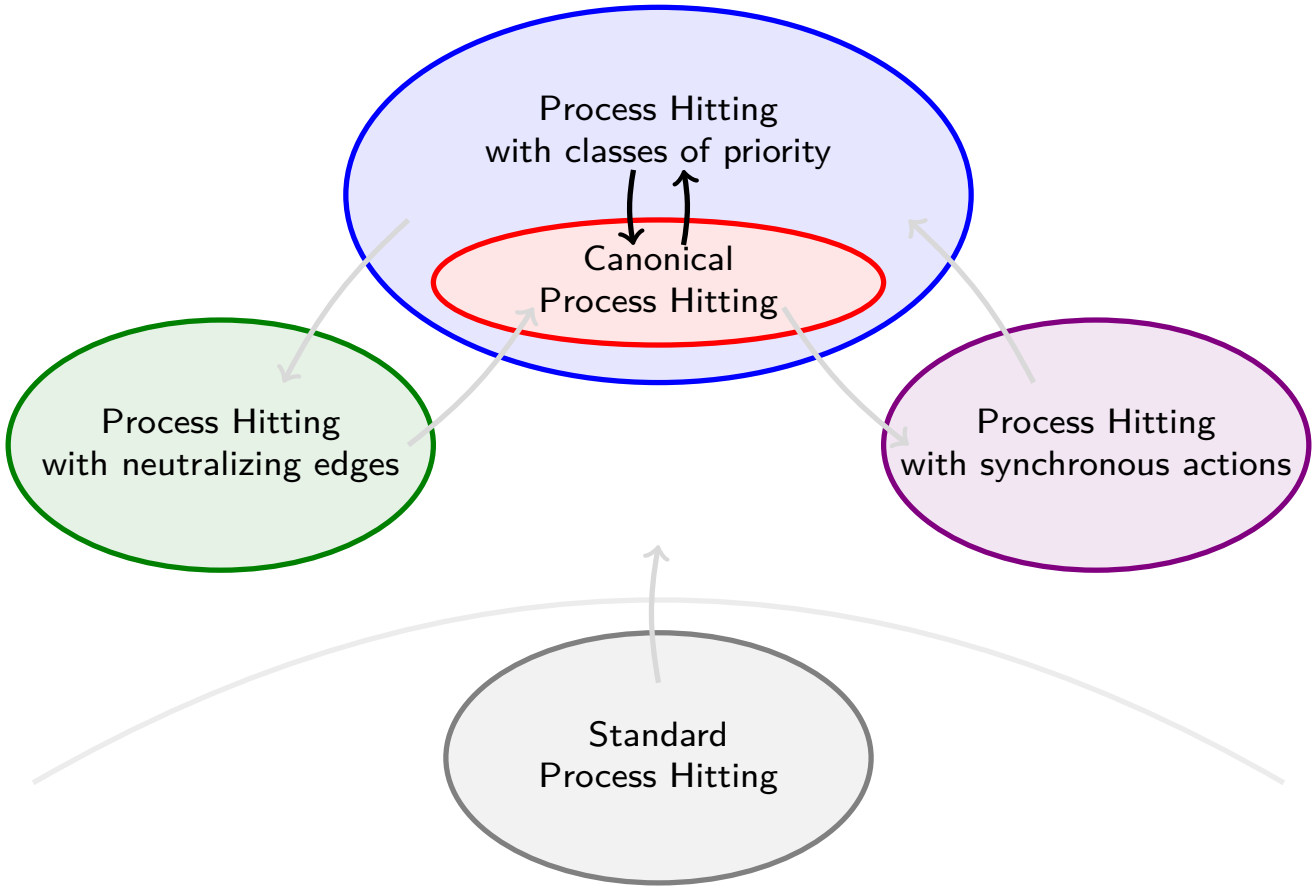
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation

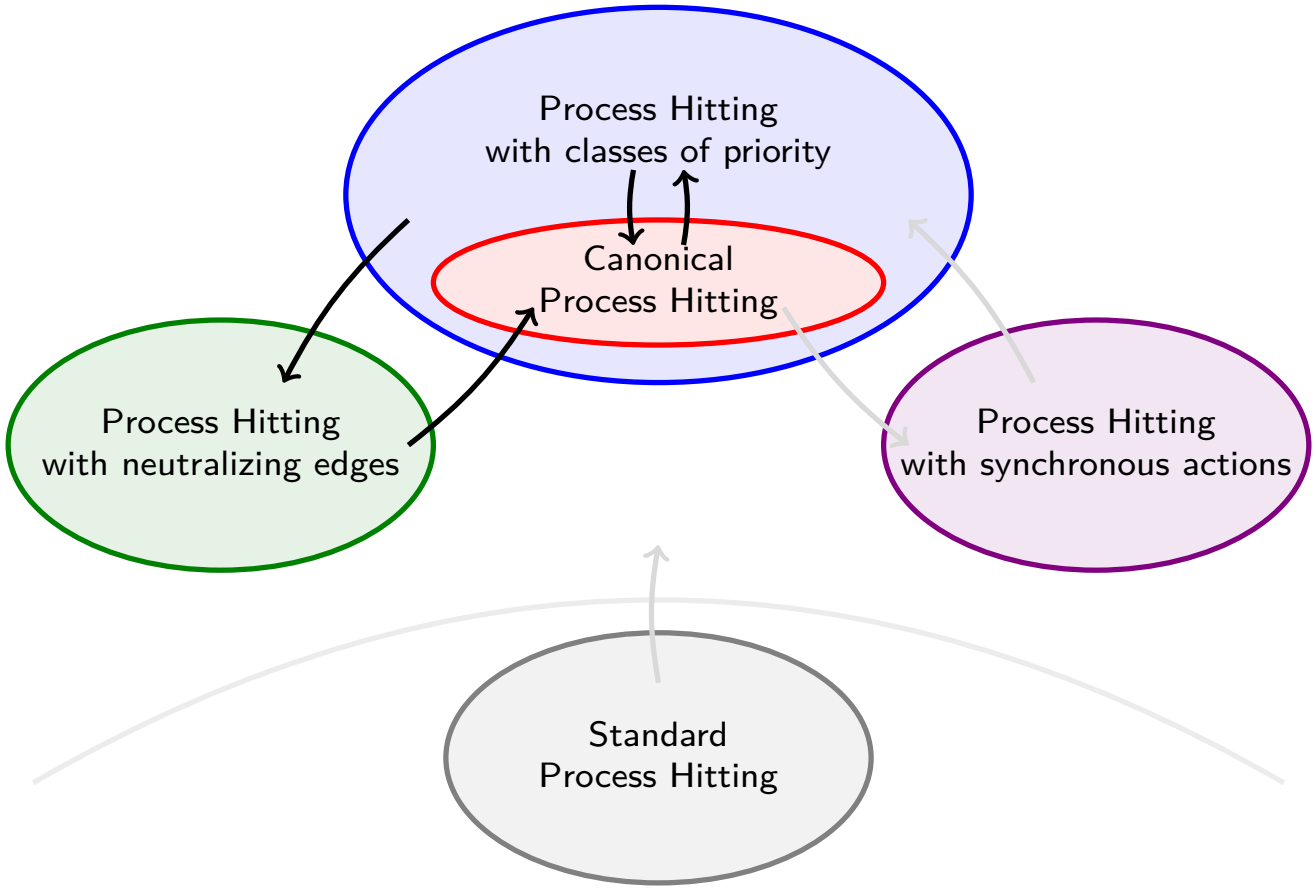
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation

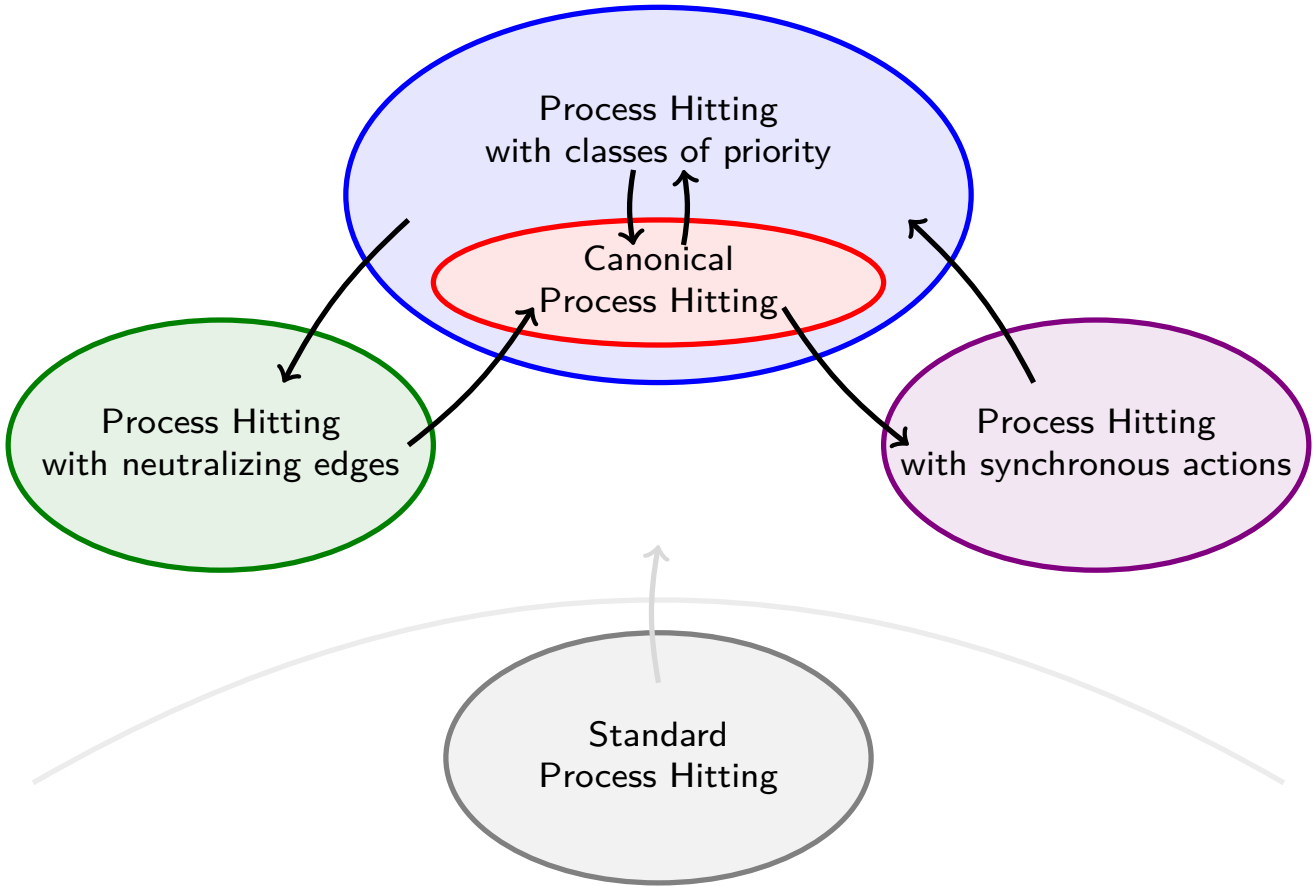
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation

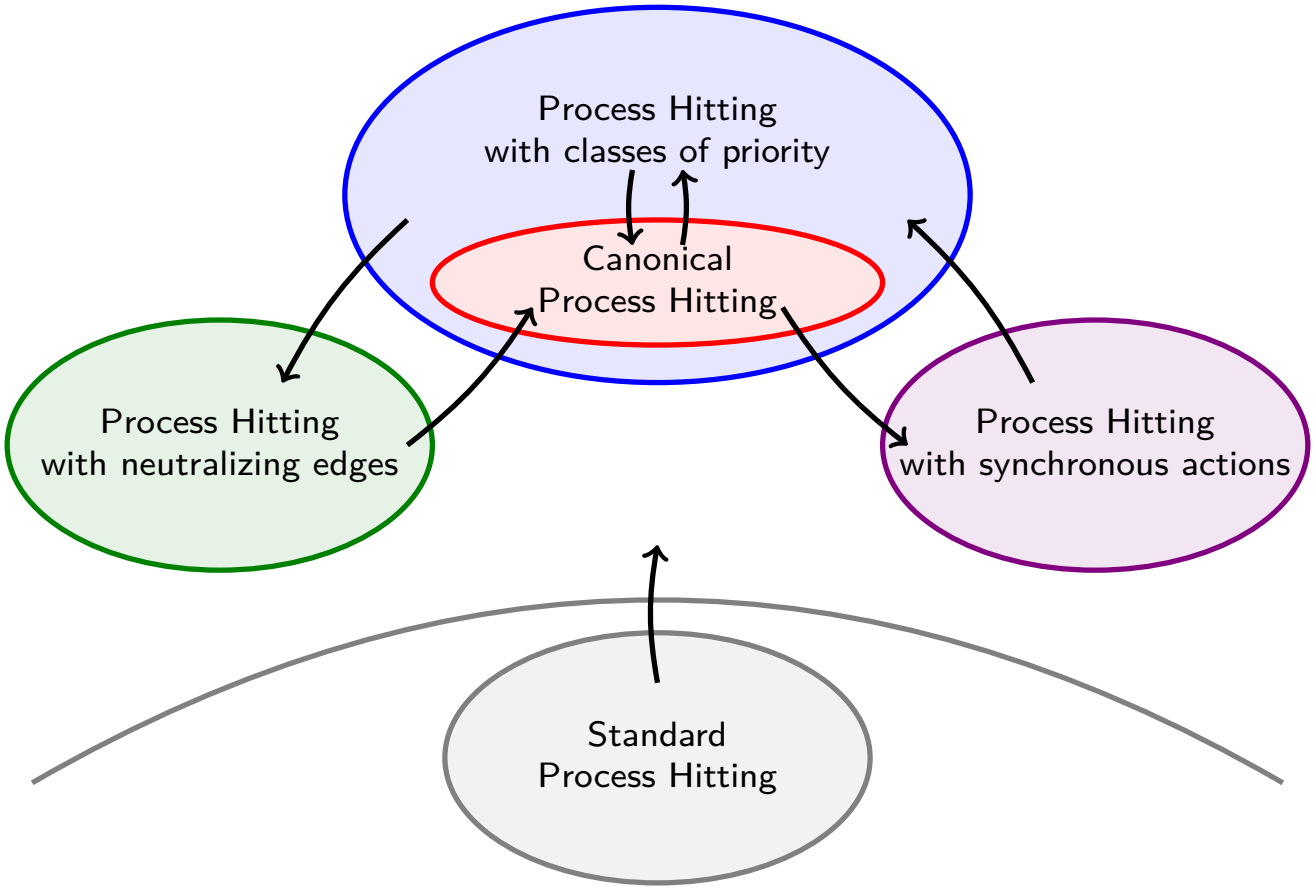
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation

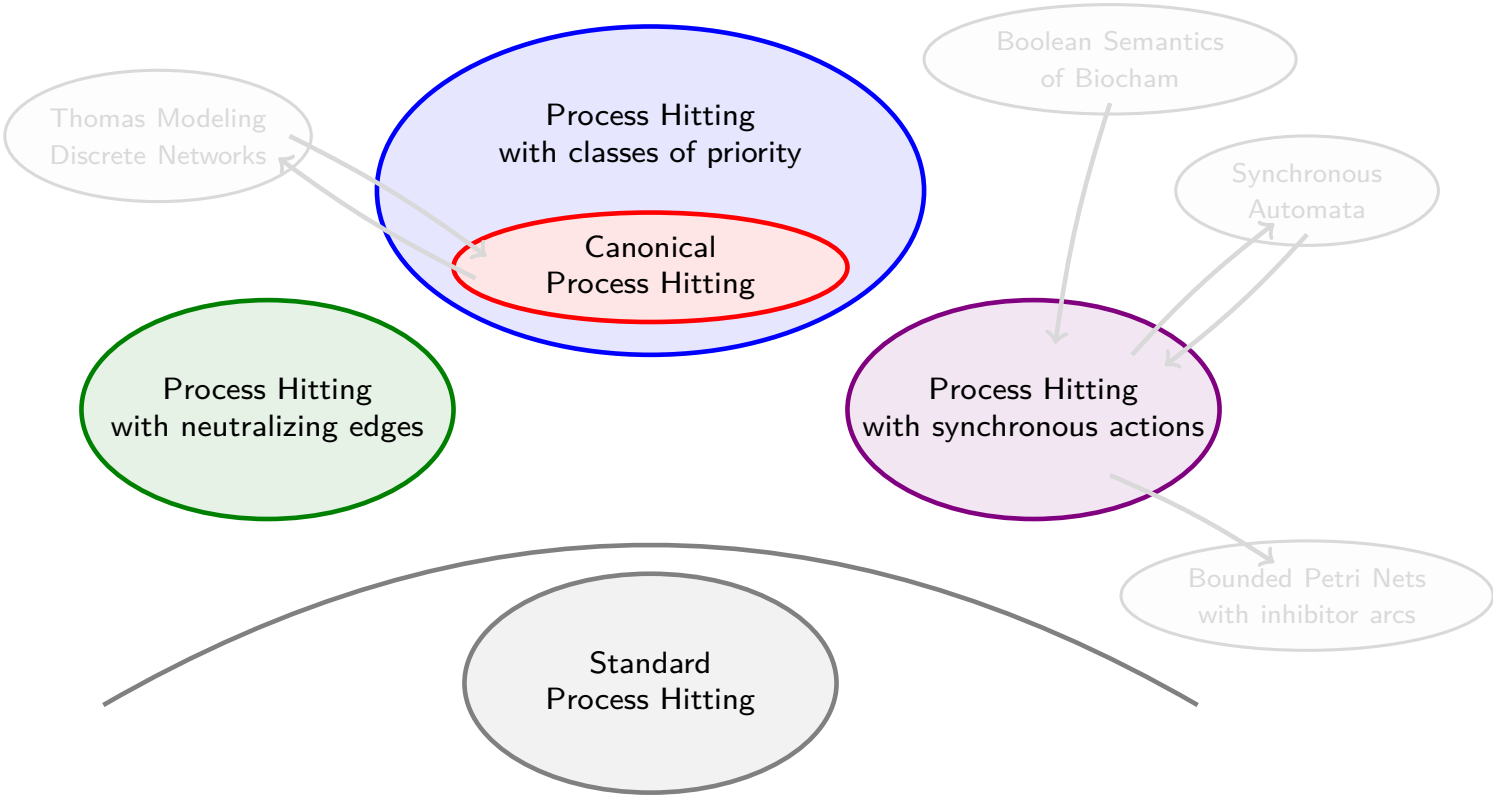
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

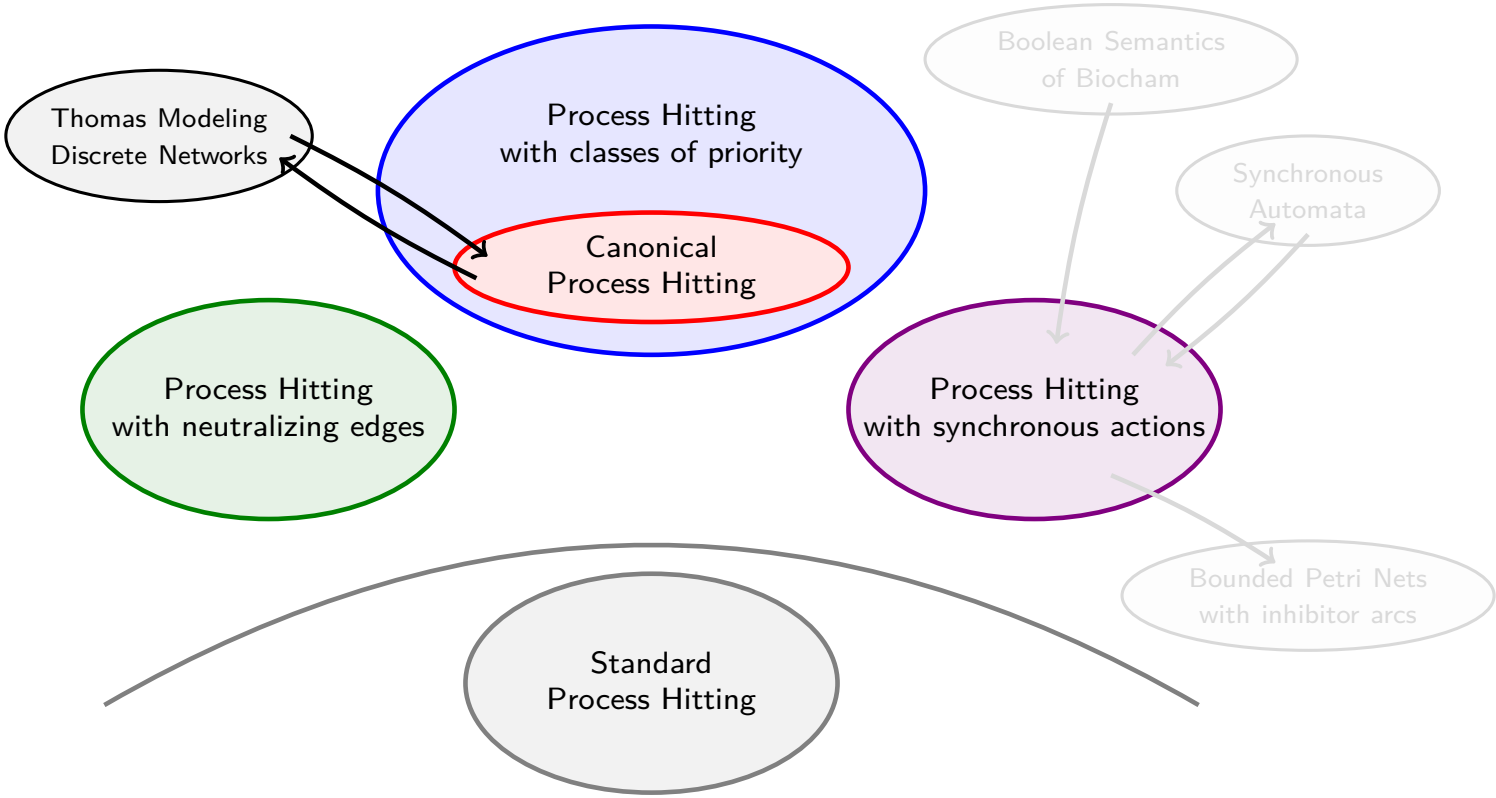
- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation

Translation From and To Other Discrete Models



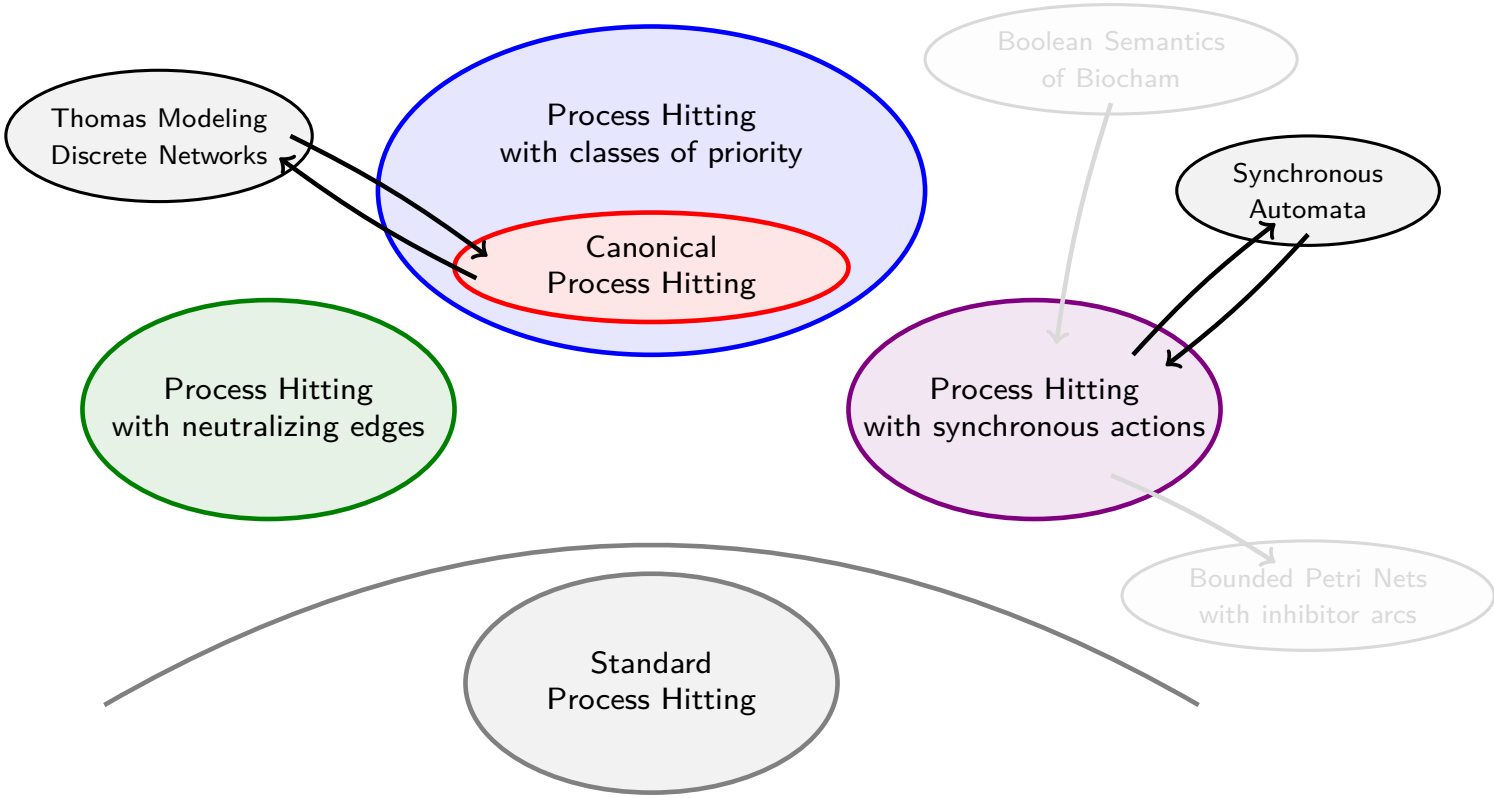
- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



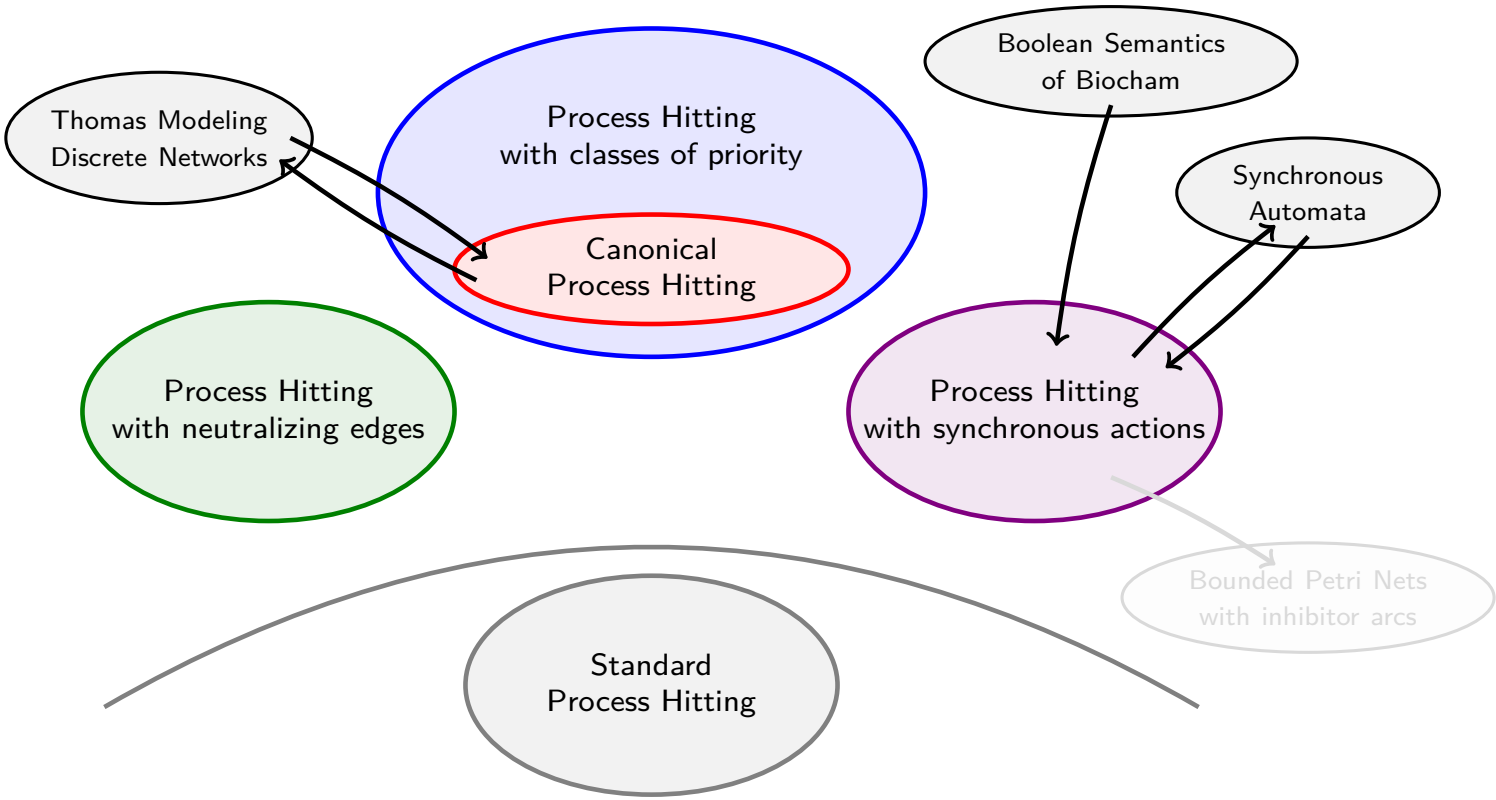
- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



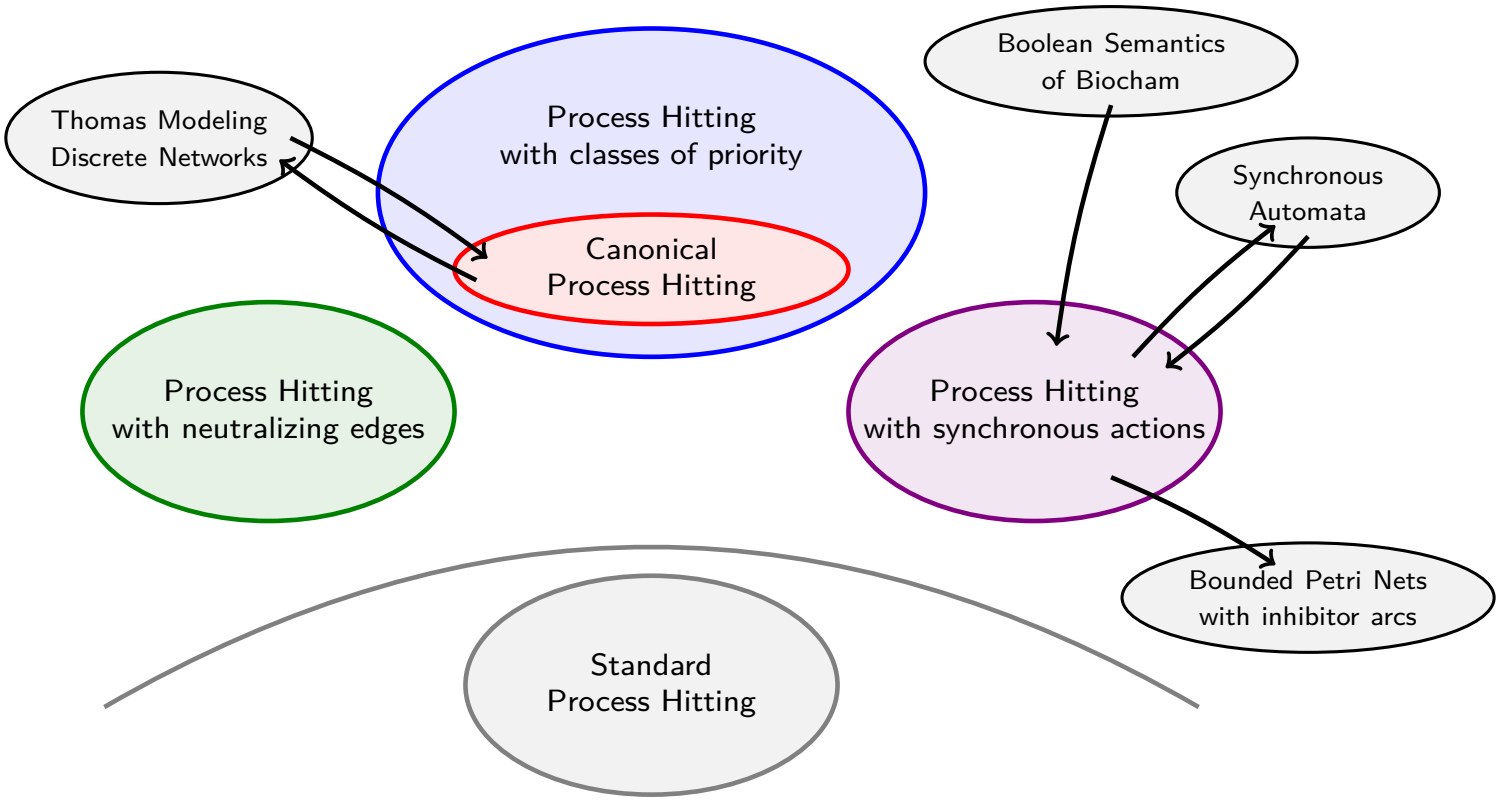
- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



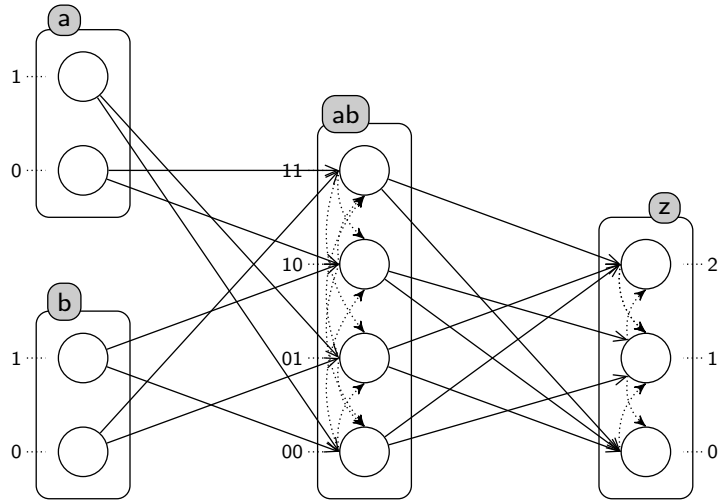
- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



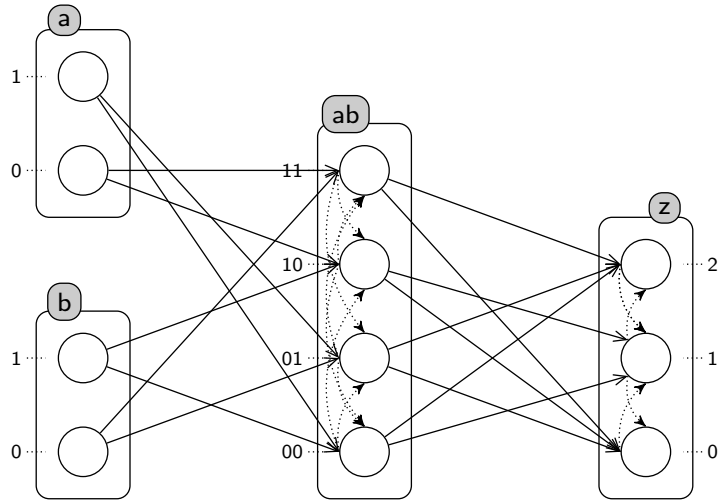
- Equivalence with discrete networks / Thomas modeling
- Equivalence with synchronous automata networks
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

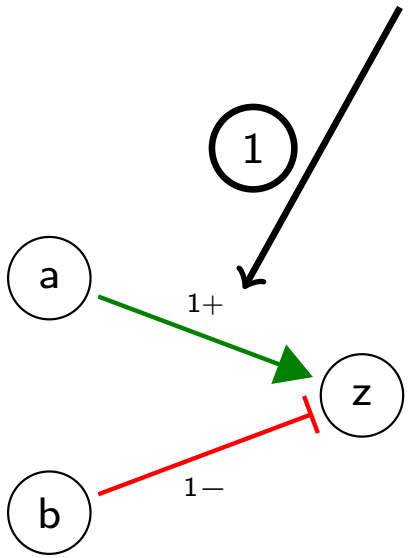
Inferring a BRN with Thomas' parameters



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

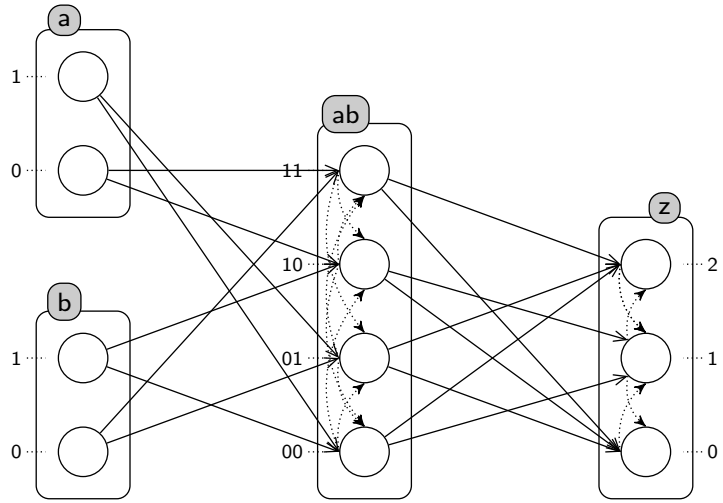
Inferring a BRN with Thomas' parameters

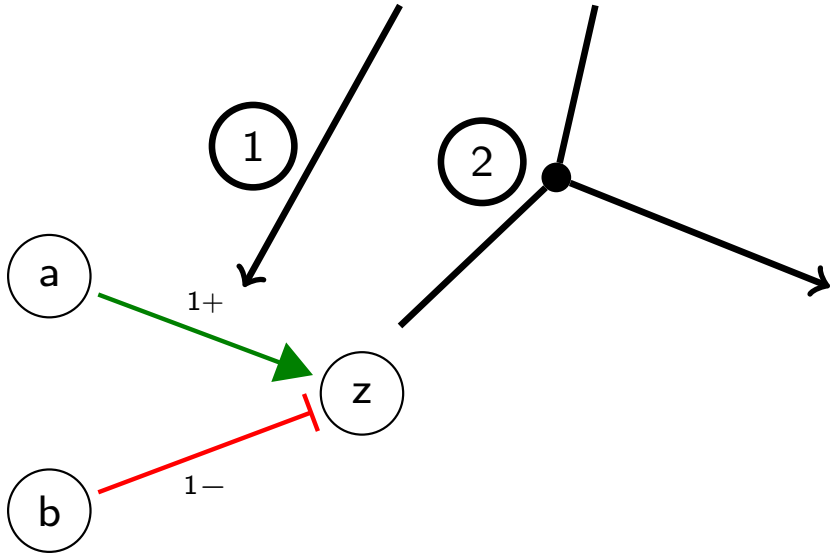




ω		$k_{z,\omega}$
a	b	
−	+	1
−	−	0
+	+	2
+	−	1

Inferring a BRN with Thomas' parameters

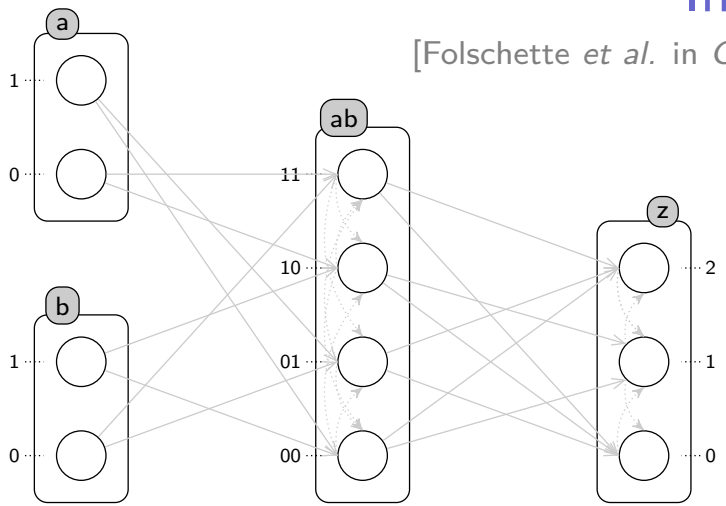


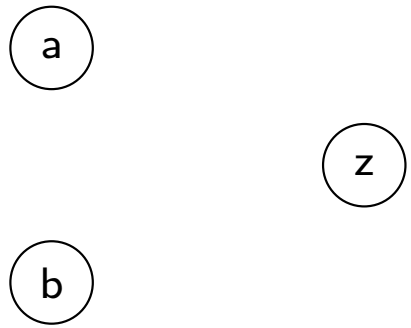


ω		$k_{z,\omega}$
a	b	
−	+	1
−	−	0
+	+	2
+	−	1

Inferring the Interaction Graph

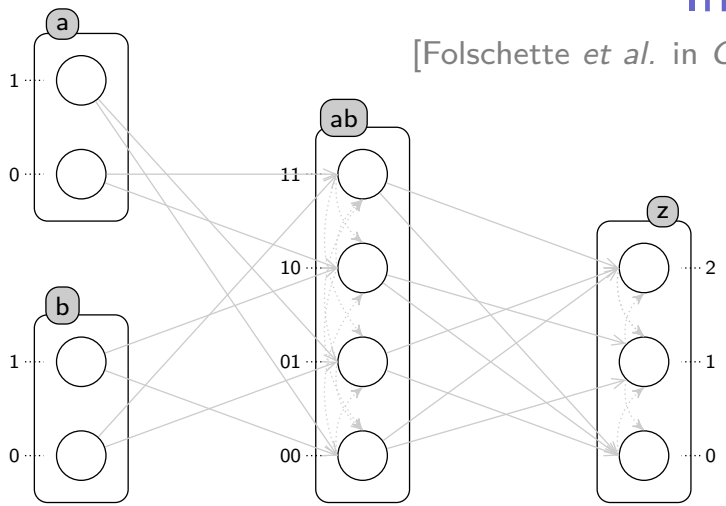
[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]

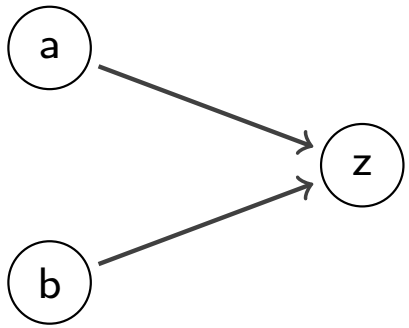




Inferring the Interaction Graph

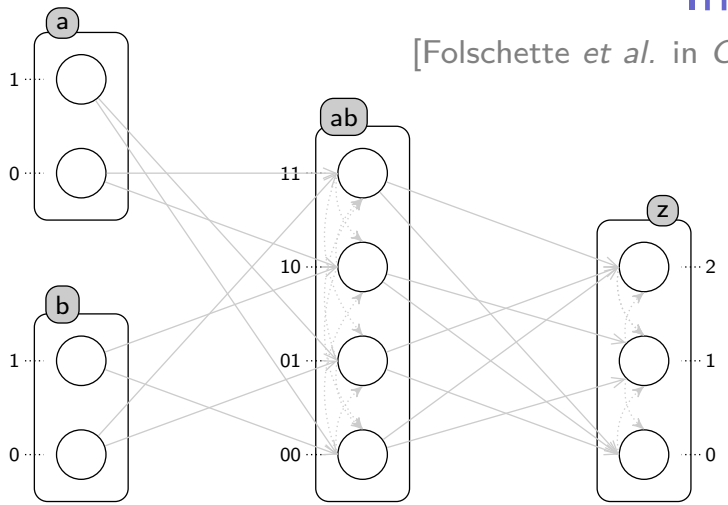
[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]

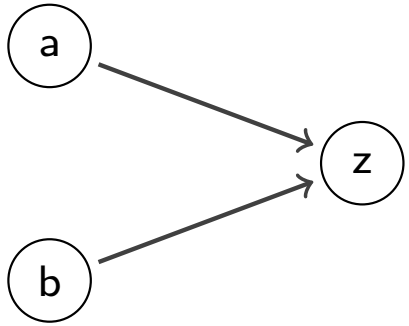




Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]

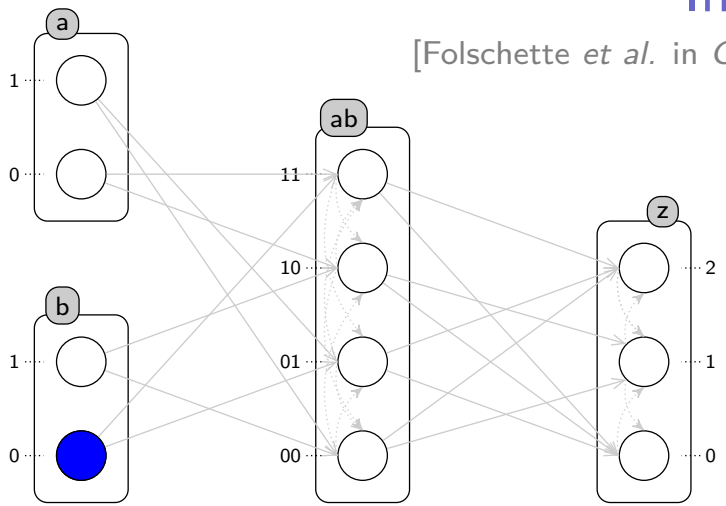


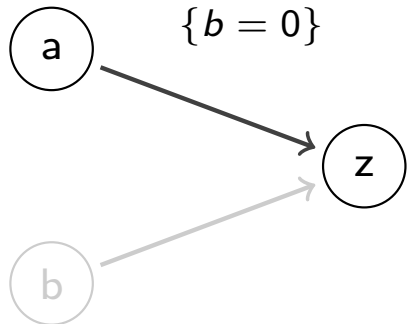


→ **Exhaustive search in all possible configurations**

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



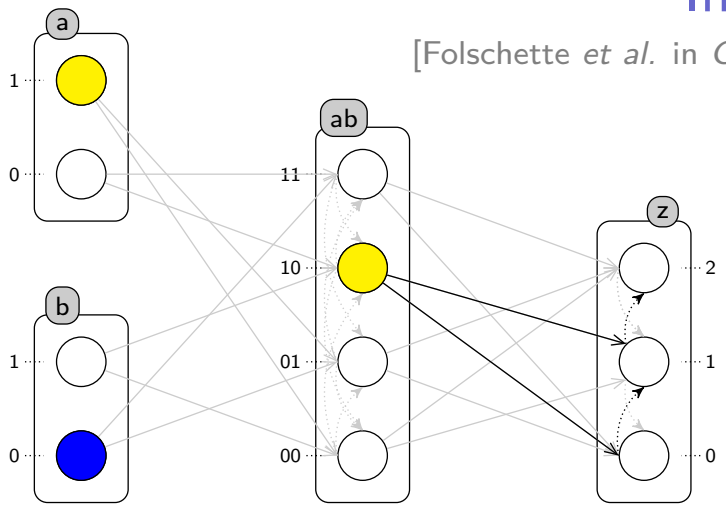


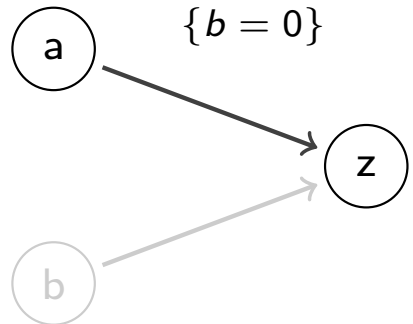
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



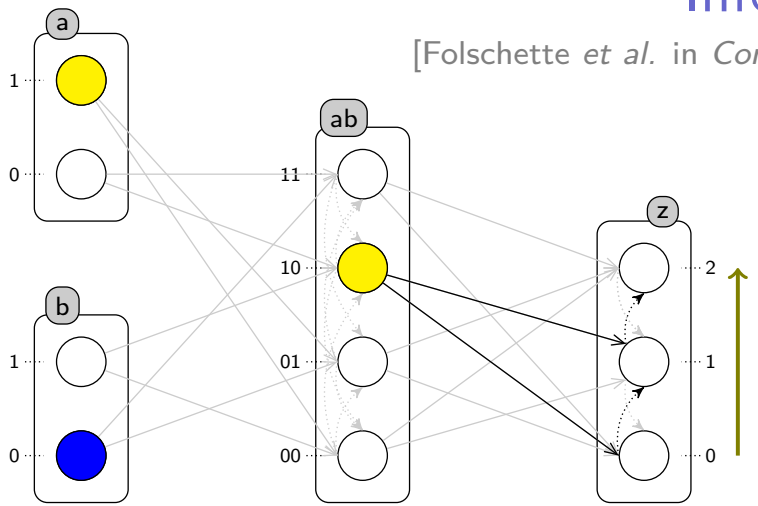


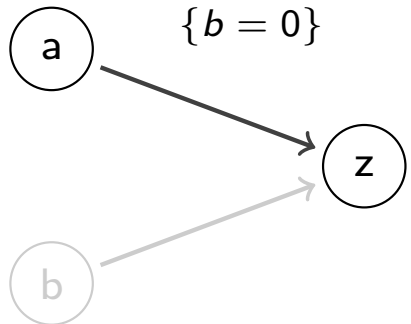
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of the regulator [a_0, a_1] and watch the **evolution**.

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



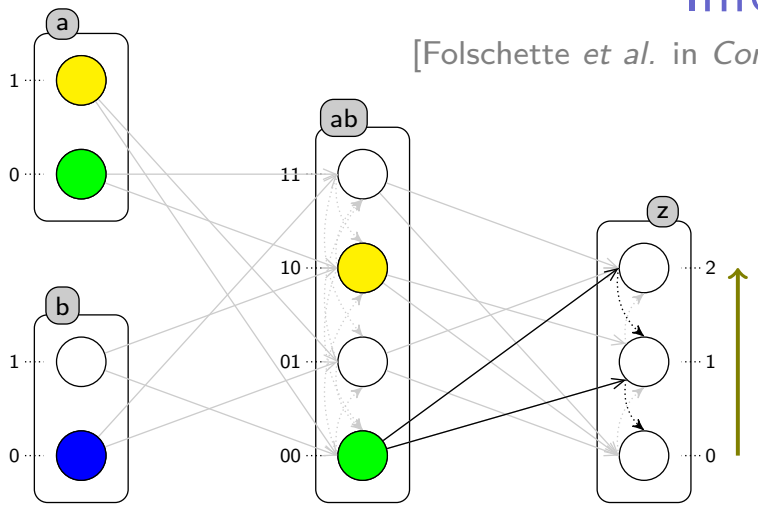


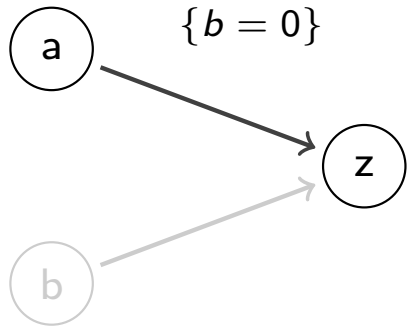
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of the regulator [a_0, a_1] and watch the **evolution**.

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



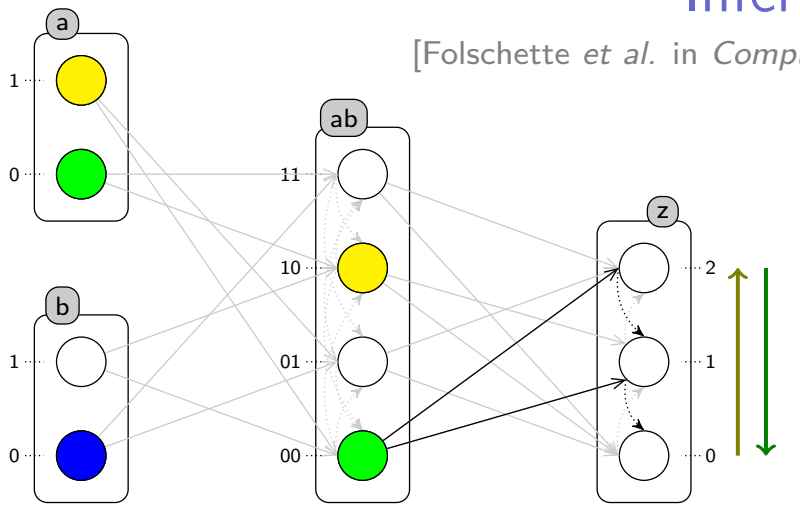


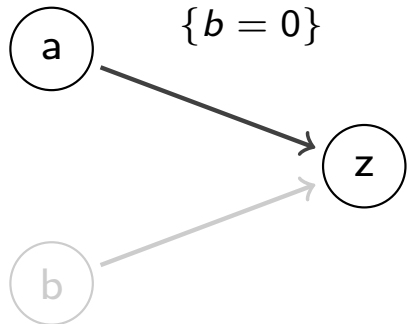
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of the regulator [a_0, a_1] and watch the **evolution**.

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



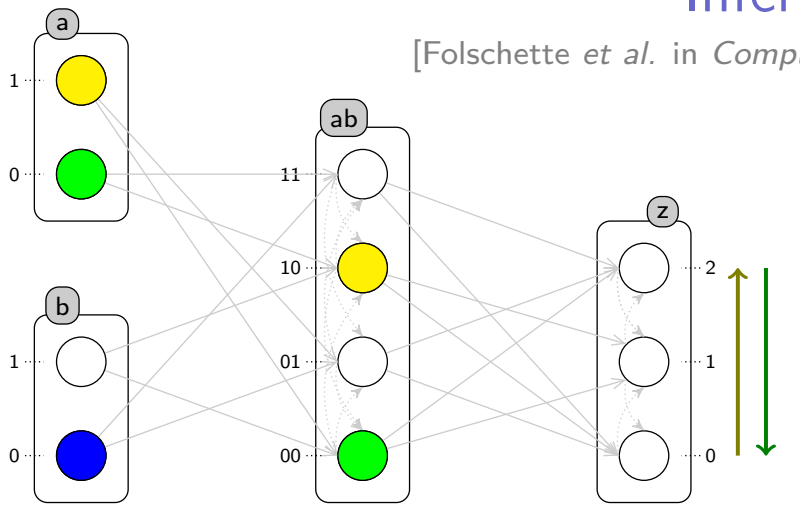


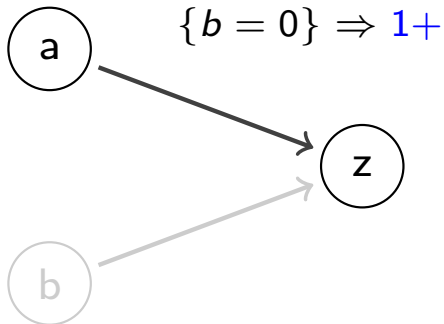
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of the regulator [a_0, a_1] and watch the **evolution**.

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



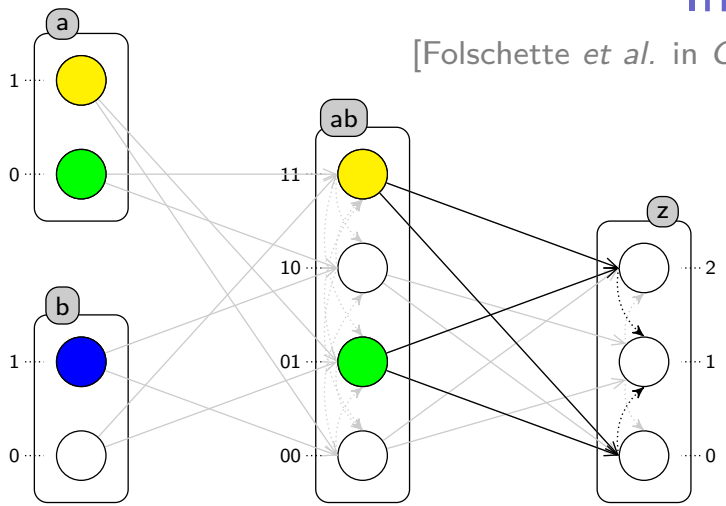


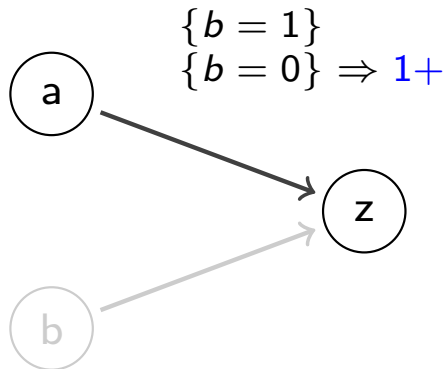
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



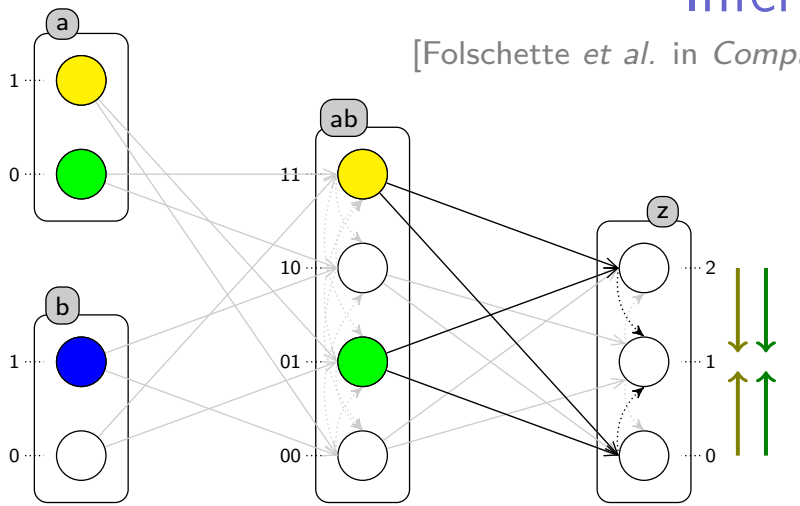


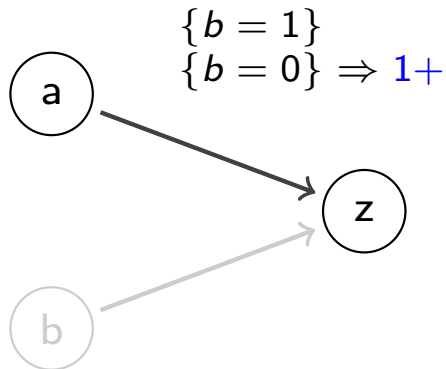
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



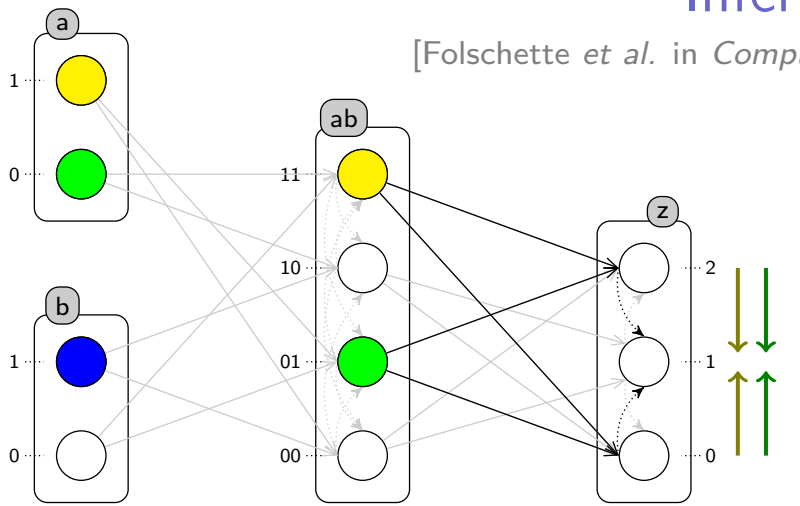


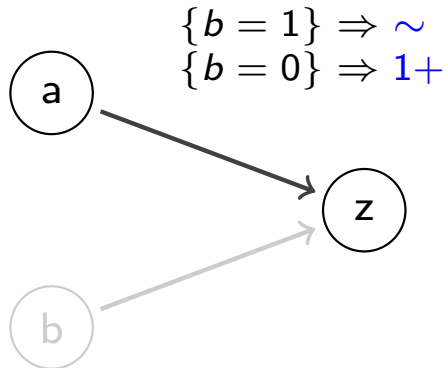
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



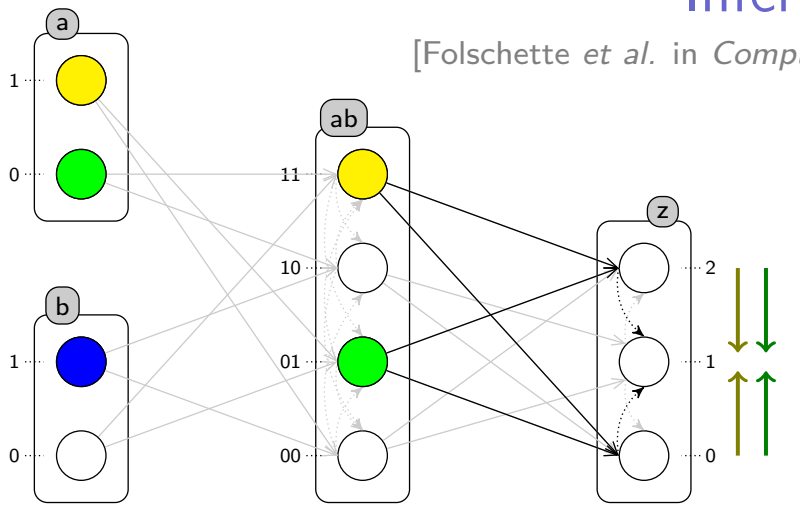


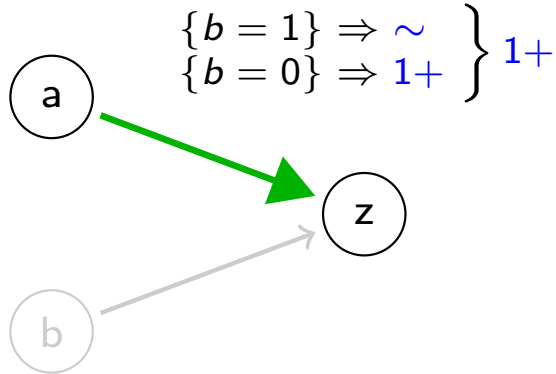
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]



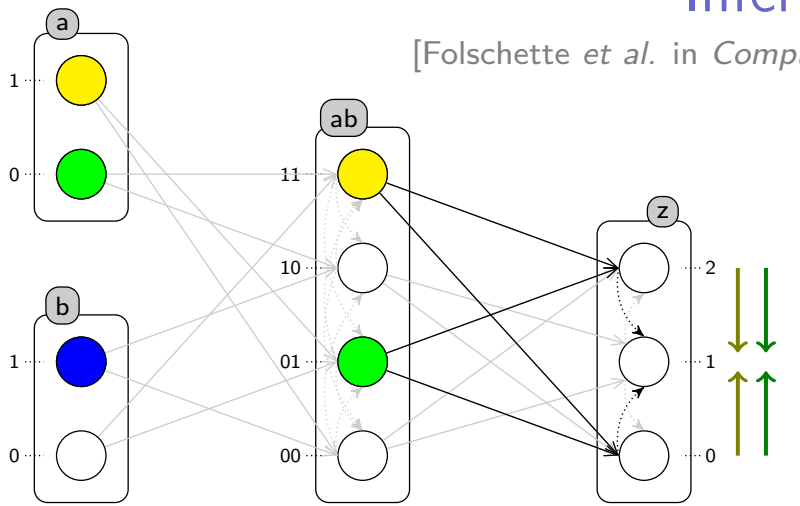


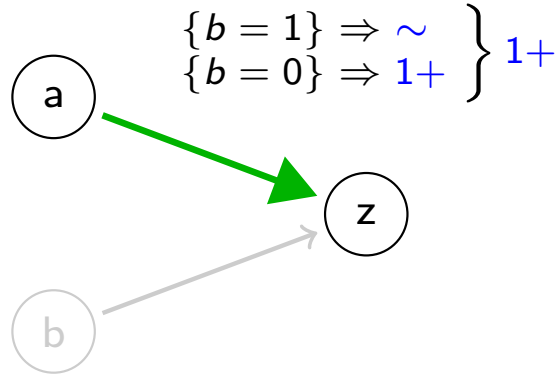
→ **Exhaustive search in all possible configurations**

1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of the regulator [a_0, a_1] and watch the **evolution**.
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

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]





→ **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 the regulator [a_0, a_1] and watch the **evolution**.
3. Conclude locally: ($a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2$) $\Rightarrow$ activation ($+$) & threshold $= 1$.
4. Iterate and conclude globally.

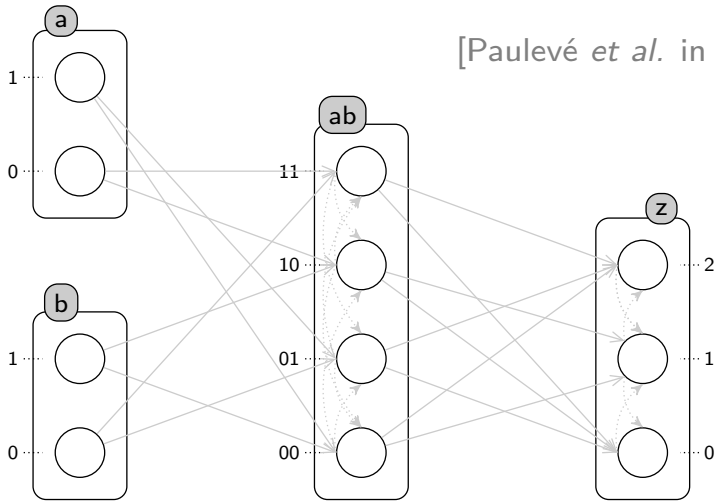
Problematic cases:

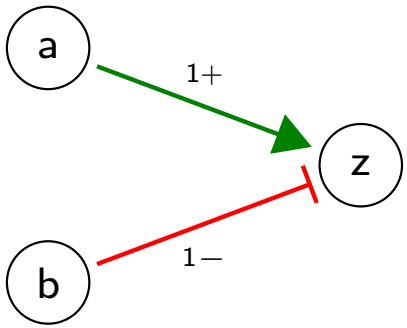
→ No focal processes (cycle)
→ Opposite influences ($+$ & $-$)

$\Rightarrow$ Unsigned edge

Inferring Parameters

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

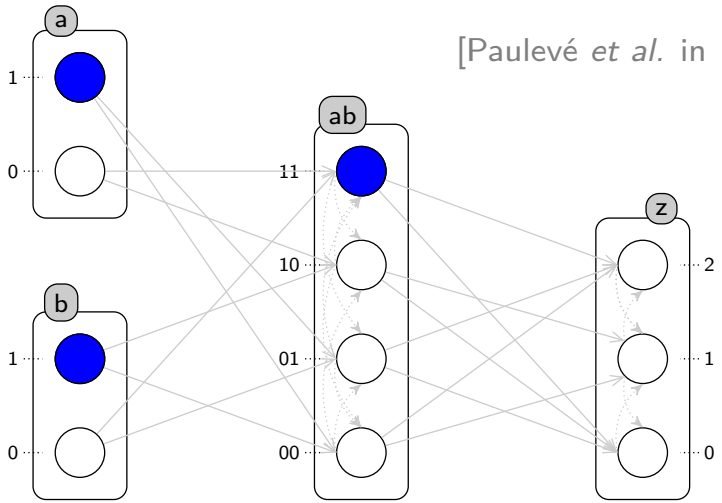


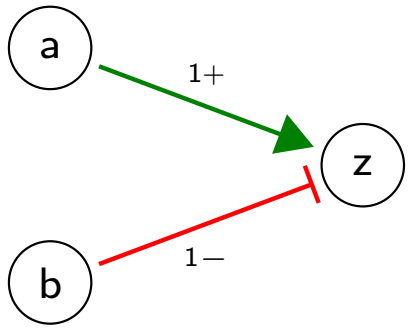


ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	

Inferring Parameters

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



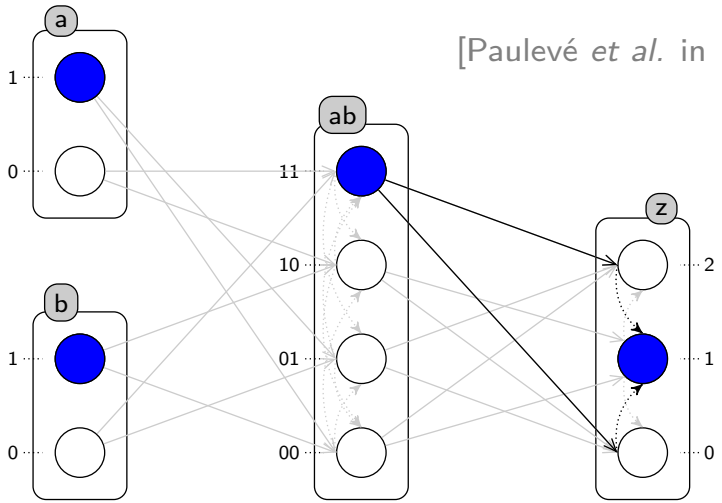


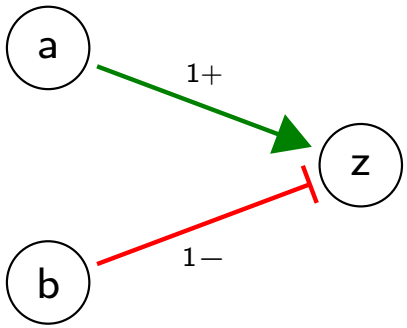
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	

1. For each configuration of resources $[\omega = \{a^+, b^-\}]$

Inferring Parameters

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



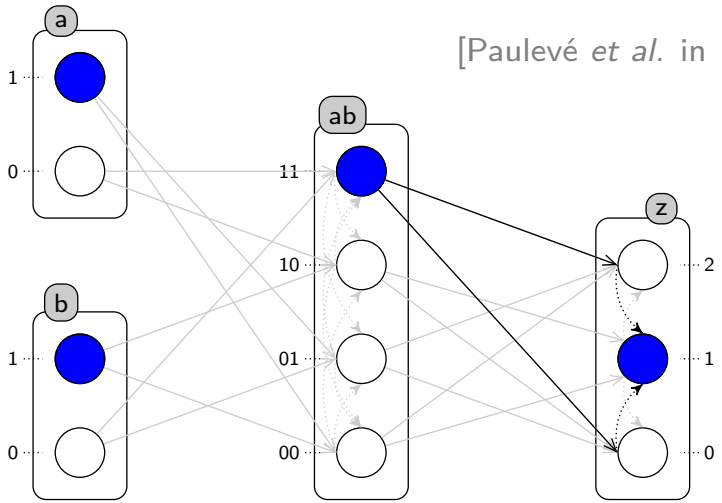


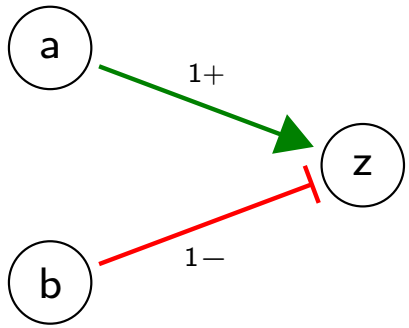
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	

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

Inferring Parameters

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



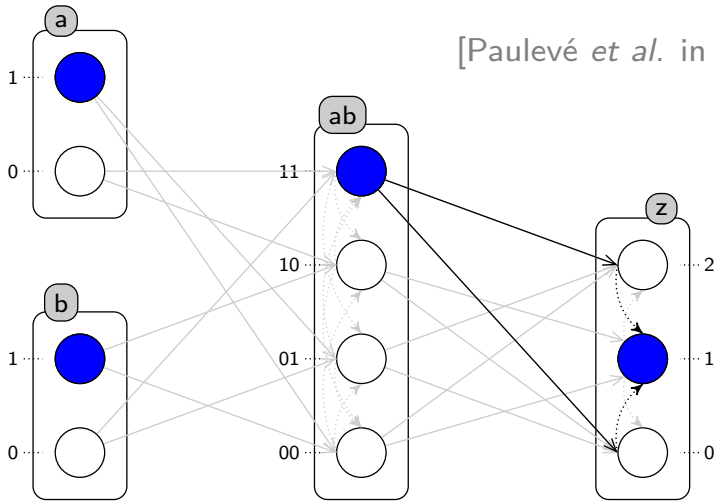


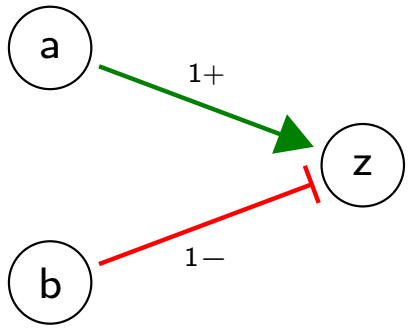
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

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

Inferring Parameters

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





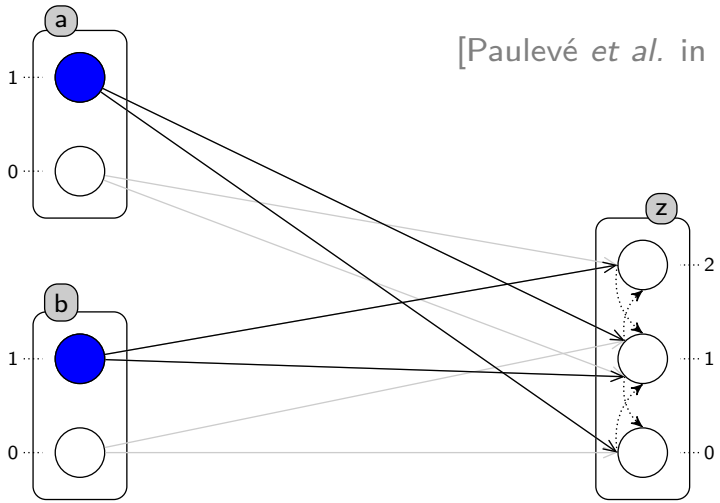
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

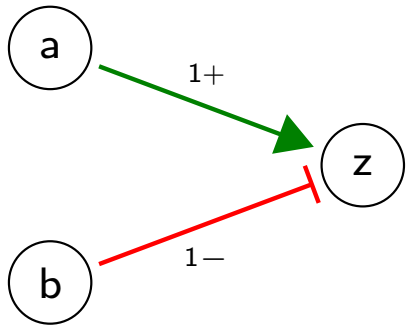
1. For each configuration of resources $[\omega = \{a^+, b^-\}]$
   find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

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

Inferring Parameters

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





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 [Bernot *et al.* in *Concurrent Models in Molecular Biology*, 2007]
   - The dynamics of the Process Hitting

$$[k_{z,\{a^+,b^-\}} \in \{0;1;2\};\ k_{z,\{a^-,b^+\}} \in \{0;1;2\}]$$

Translation to Thomas Modeling

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]

- Two successive inferences: 1) interaction graph; 2) parameters
- Exhaustive analysis of the local dynamics for each regulator
- enumeration of all parametrizations compatible with the dynamics

Complexity:
Linear in the number of genes,
Exponential in the number of regulators of one component

Translation to Thomas Modeling

[Folschette *et al.* in *Computational Methods in Systems Biology*, 2012]

- Two successive inferences: 1) interaction graph; 2) parameters
- Exhaustive analysis of the local dynamics for each regulator
- enumeration of all parametrizations compatible with the dynamics

Complexity:

Linear in the number of genes,
Exponential in the number of regulators of one component

	Models			Inference the IG		Inference of parameters	
Name	Sorts	Processes	Actions	Duration	Edges	Durations	Parameters
egfr20	42	152	399	**1s**	51	**1s**	192
tcrsig40	54	156	305	**1s**	55	**1s**	143
tcrsig94	133	448	1082	**100s**	197	**1s**	578
egfr104	193	744	2304	**200s**	280	**3s**	27'496

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]

General Conclusion

Standard Process Hitting allows to represent biological regulatory networks in an **atomistic** fashion:

- Existing efficient static analysis
- But temporal shift issues
- Limited modeling power

Extensions of the Process Hitting to improve the expressivity:

- Rectification of the temporal shift $\longrightarrow$ Strictly higher expressivity
- Allows to abstract temporal parameters
- New links to other formalisms (Thomas, PN, etc.)

Static analysis of the Canonical Process Hitting:

- Efficient analysis of reachability properties
- Applicable to the extensions at the cost of a translation
- New kind of property: simultaneous activation

Outlooks

New **exploitation** possibilities:

- Modeling and analysis of full databases
- Study of uncontrollable behaviors or punctual perturbations
- Research of interesting properties (attractors, oscillations, ...)

Improvement of the **static analysis**:

- Refining in order to reduce the non-conclusiveness
- New methods using by-products such as the local causality graph
- New properties to check (temporal logic, counters, ...)

Enrichment of the **modeling power**:

- Abstraction of temporal parameters: find properties to avoid Zeno behavior
- Dynamical classes of priorities
- Guarded actions or complex logic gates
- New model checking tools (Hoare logic, ...)

Thank you for your attention

Personal Contributions

Book chapter:

- Loïc Paulevé, Courtney Chancellor, Maxime Folschette, Morgan Magnin, Olivier Roux. Analyzing Large Network Dynamics with Process Hitting, In Luis Farinas del Cerro and Katsumi Inoue, editors: *Logical Modeling of Biological Systems*, 2014.

Journal article:

- Maxime Folschette, Loïc Paulevé, Katsumi Inoue, Morgan Magnin, Olivier Roux. Constructing Biological Regulatory Networks from Process Hitting models, *Theoretical Computer Science*, Vol. 586, 2015.

Conference:

- Maxime Folschette, Loïc Paulevé, Morgan Magnin, Olivier Roux. Under-approximation of reachability in multivalued asynchronous networks, CS2Bio'13, *Electronic Notes in Theoretical Computer Science*, Vol. 299, 2013.

Workshops:

- Maxime Folschette, Loïc Paulevé, Katsumi Inoue, Morgan Magnin, Olivier Roux. Concretizing the process hitting into biological regulatory networks, CMSB'12, *Lecture Notes in Computer Science*, 2012.
- Maxime Folschette, Loïc Paulevé, Katsumi Inoue, Morgan Magnin, Olivier Roux. Abducing Biological Regulatory Networks from Process Hitting models, *ECML-PKDD'12 / LDSSB'12*, 2012.

Bibliography

- Adrien Richard, Jean-Paul Comet, Gilles Bernot. R. Thomas' logical method, 2008. Invité à *Tutorials on modelling methods and tools: Modelling a genetic switch and Metabolic Networks*, Spring School on Modelling Complex Biological Systems in the Context of Genomics.
- Stuart A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3), pages 437–467, 1969.
- René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3), pages 563–585, 1973.
- Élisabeth Remy, Paul Ruet and Denis Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Advances in Applied Mathematics*, 41(3), pages 335–350, Elsevier, 2008.
- Adrien Richard, Jean-Paul Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics*, 155(18), pages 2403–2413, 2007.
- Gilles Bernot, Jean-Paul Comet, Adrien Richard and Janine Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3), pages 339–347, Elsevier, 2004.
- Sohei Ito, Naoko Izumi, Shigeki Hagihara and Naoki Yonezaki. Qualitative analysis of gene regulatory networks by satisfiability checking of Linear Temporal Logic. In 2010 IEEE International Conference on *BioInformatics and BioEngineering*, pages 232–237, IEEE, 2010.

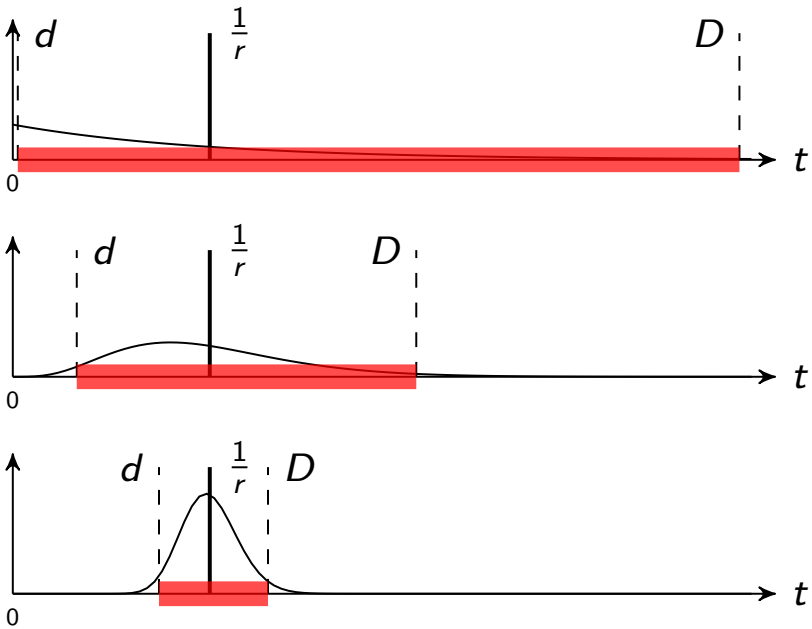
Bibliography

- Loïc Paulevé, Morgan Magnin, Olivier Roux. Refining dynamics of gene regulatory networks in a stochastic π-calculus framework. In Corrado Priami, Ralph-Johan Back, Ion Petre, and Erik de Vink, editors: Transactions on Computational Systems Biology XIII, *Lecture Notes in Computer Science* 6575, pages 171–191, 2011.
- Loïc Paulevé, Morgan Magnin, Olivier Roux. Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*, 2012.
- Gilles Bernot, Franck Cassez, Jean-Paul Comet, Franck Delaplace, Céline Müller, Olivier Roux. Semantics of Biological Regulatory Networks. Proceedings of the First Workshop on Concurrent Models in Molecular Biology, *Electronic Notes in Theoretical Computer Science* 180(3), pages 3–14, 2007.
- Paul François, Vincent Hakim, Eric D Siggia. Deriving structure from evolution : metazoan segmentation. *Molecular Systems Biology*, 3(1), 2007.
- Özgür Sahin *et al.* Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Systems Biology*, 3(1), 2009.
- Regina Samaga *et al.* The Logic of EGFR/ErbB Signaling: Theoretical Properties and Analysis of High-Throughput Data. *PLoS Computational Biology*, 5(8), 2009.
- Steffen Klamt *et al.* A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, 7(1), 2006.
- Julio Saez-Rodriguez *et al.* A Logical Model Provides Insights into T Cell Receptor Signaling. *PLoS Computational Biology*, 3(8), 2007.

Stochastic Parameters

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

- Introduction of temporal properties
- Stochastic parameters (r, sa) equivalent to a **firing interval** $[d; D]$

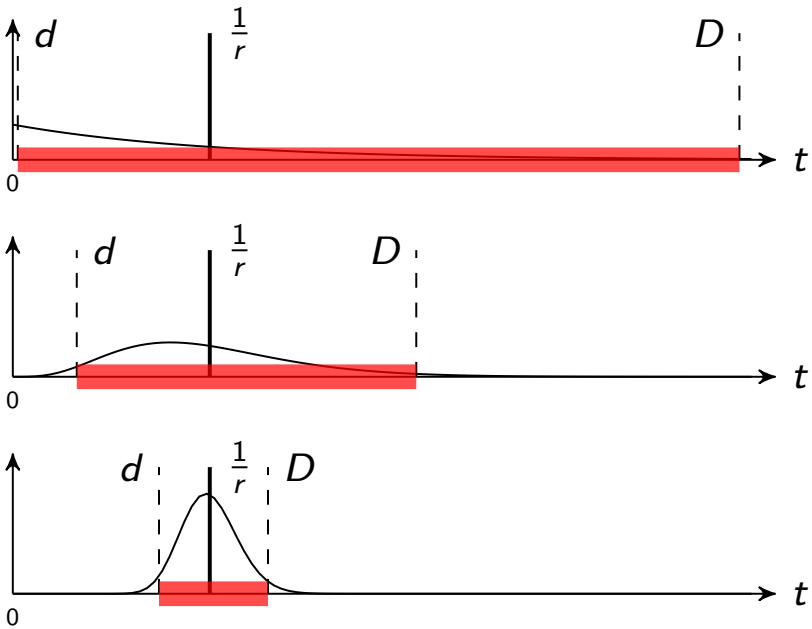


Intervalle de tir

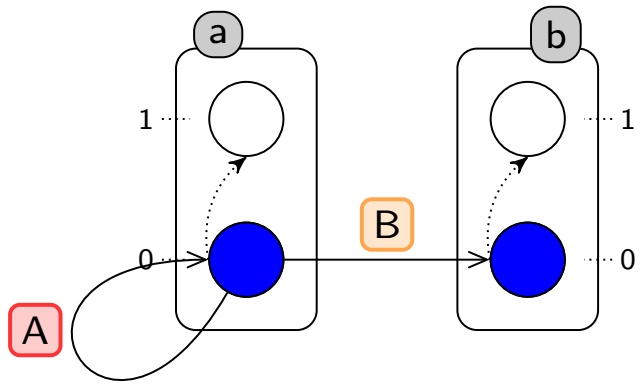
Stochastic Parameters

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

- Introduction of temporal properties
- Stochastic parameters (r, sa) equivalent to a **firing interval** $[d; D]$

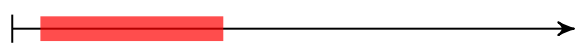


Intervalle de tir



$a_0 \rightarrow b_0 \Rsh b_1$ (B)

$a_0 \rightarrow a_0 \Rsh a_1$ (A)

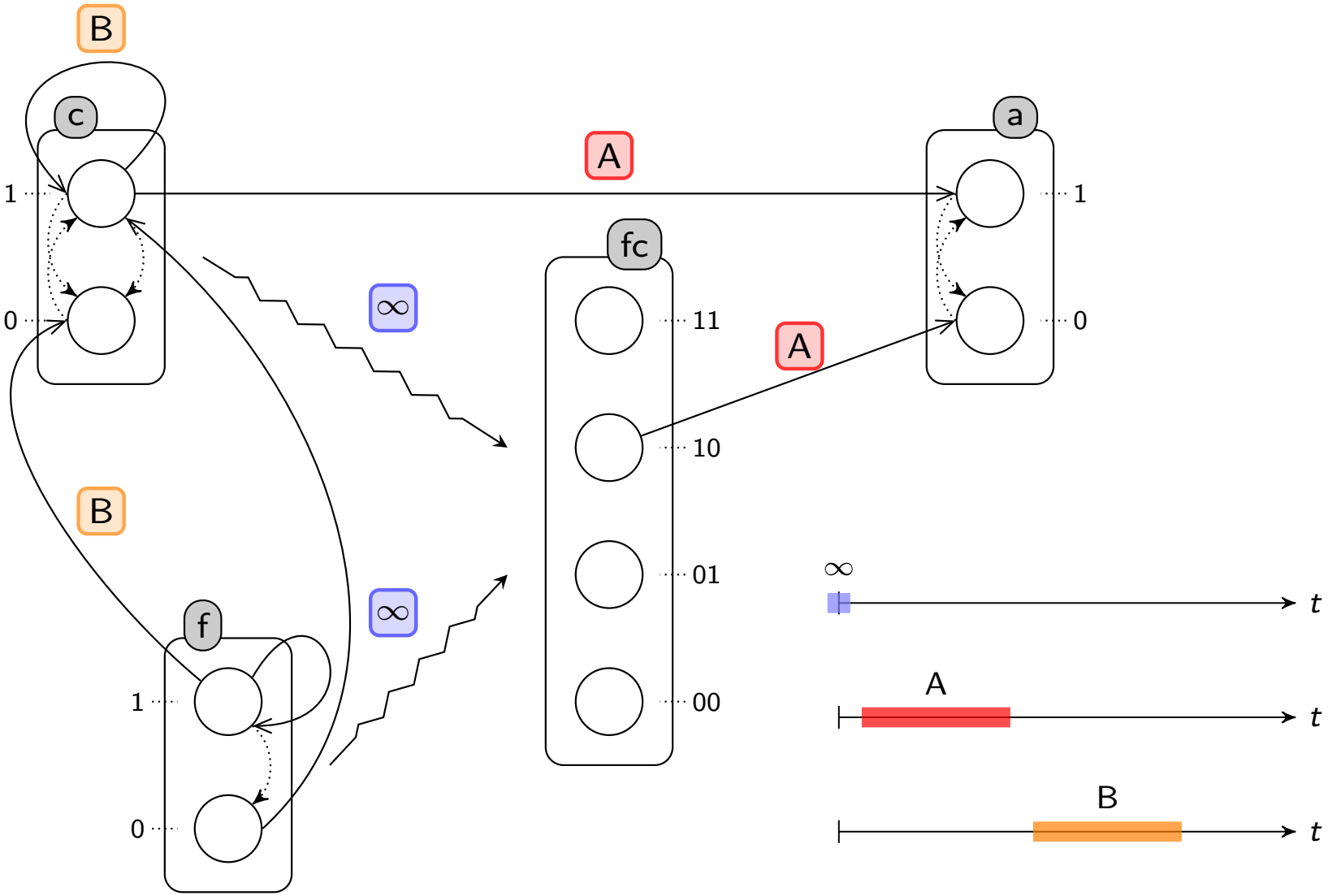


$\rightarrow$ **Very low probability** to reach b_1

- Simulation $\rightarrow$ not formal
- *Model-checking* $\rightarrow$ High complexity for an acceptable precision

Use of Stochastic Parameters

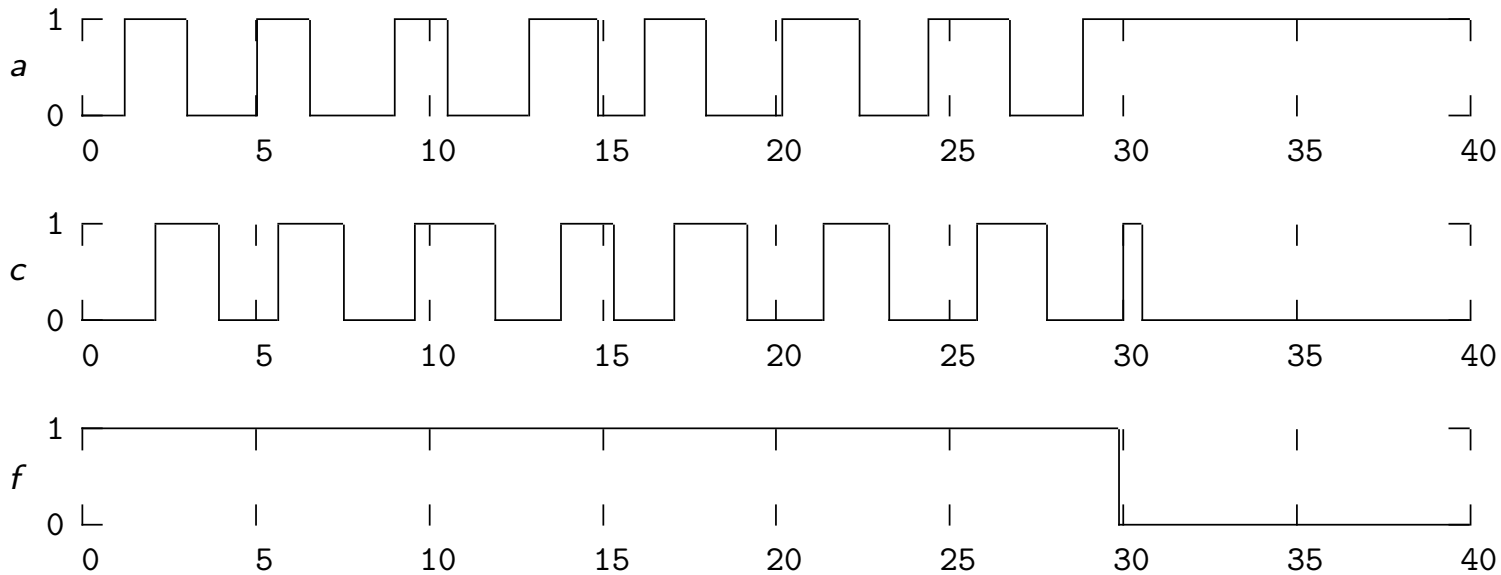
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]



Temporal Simulation

[Paulevé (PhD thesis), 2011]

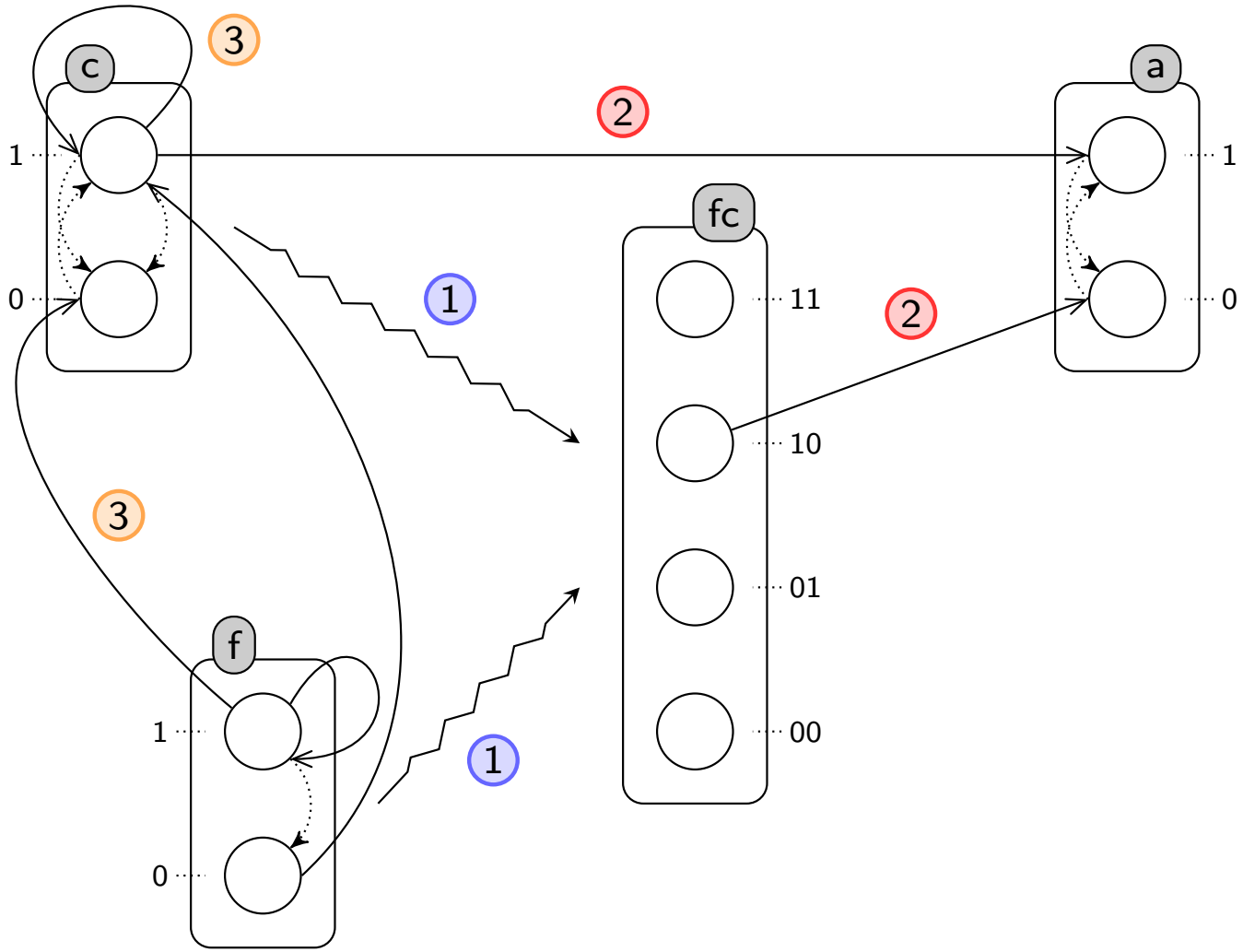
- Simulation with stochastic parameters:



- Other possible analysis: stochastic model checkers (PRISM)
  → But combinatoric explosion: PRISM fails for more than 5 components

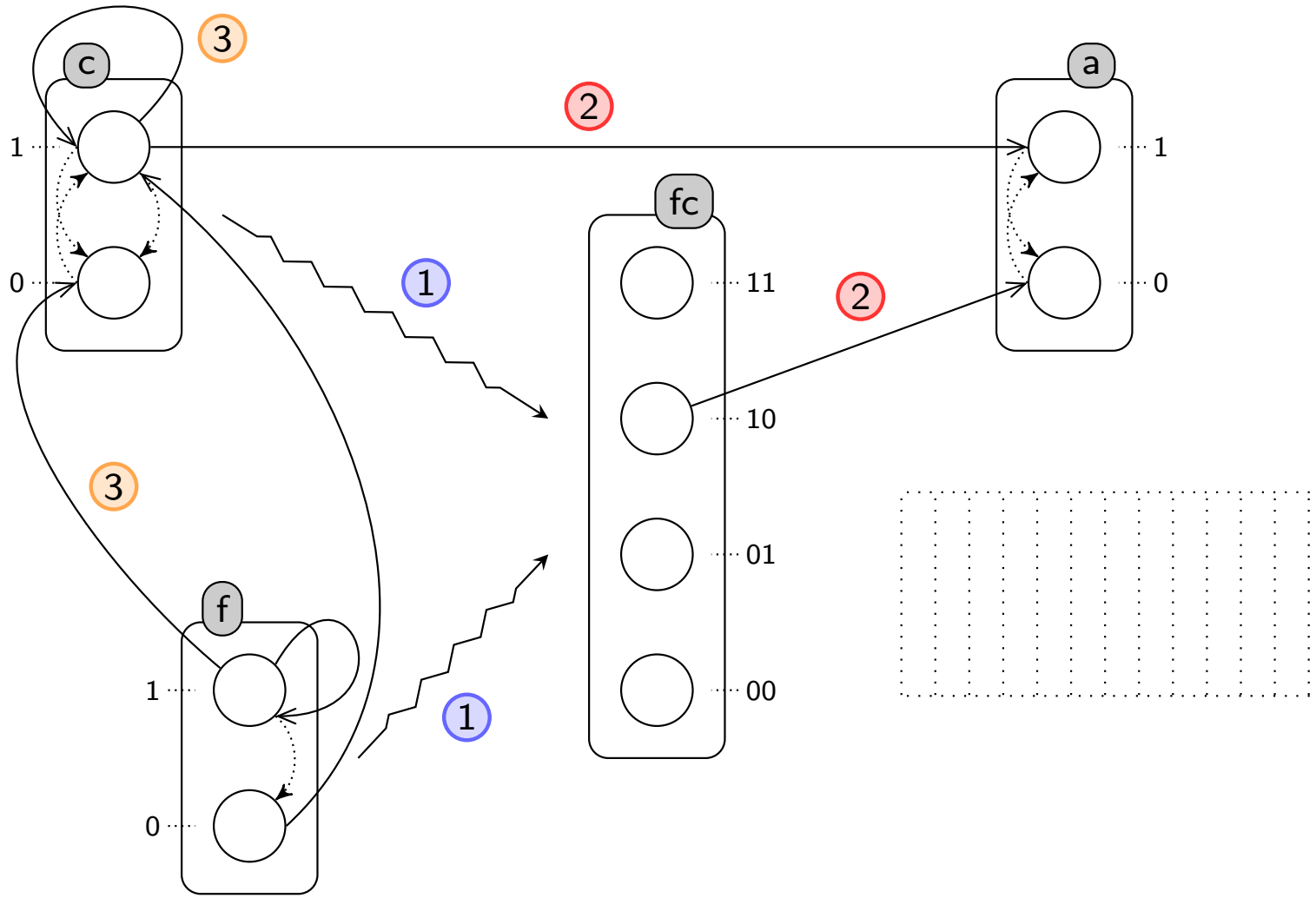
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



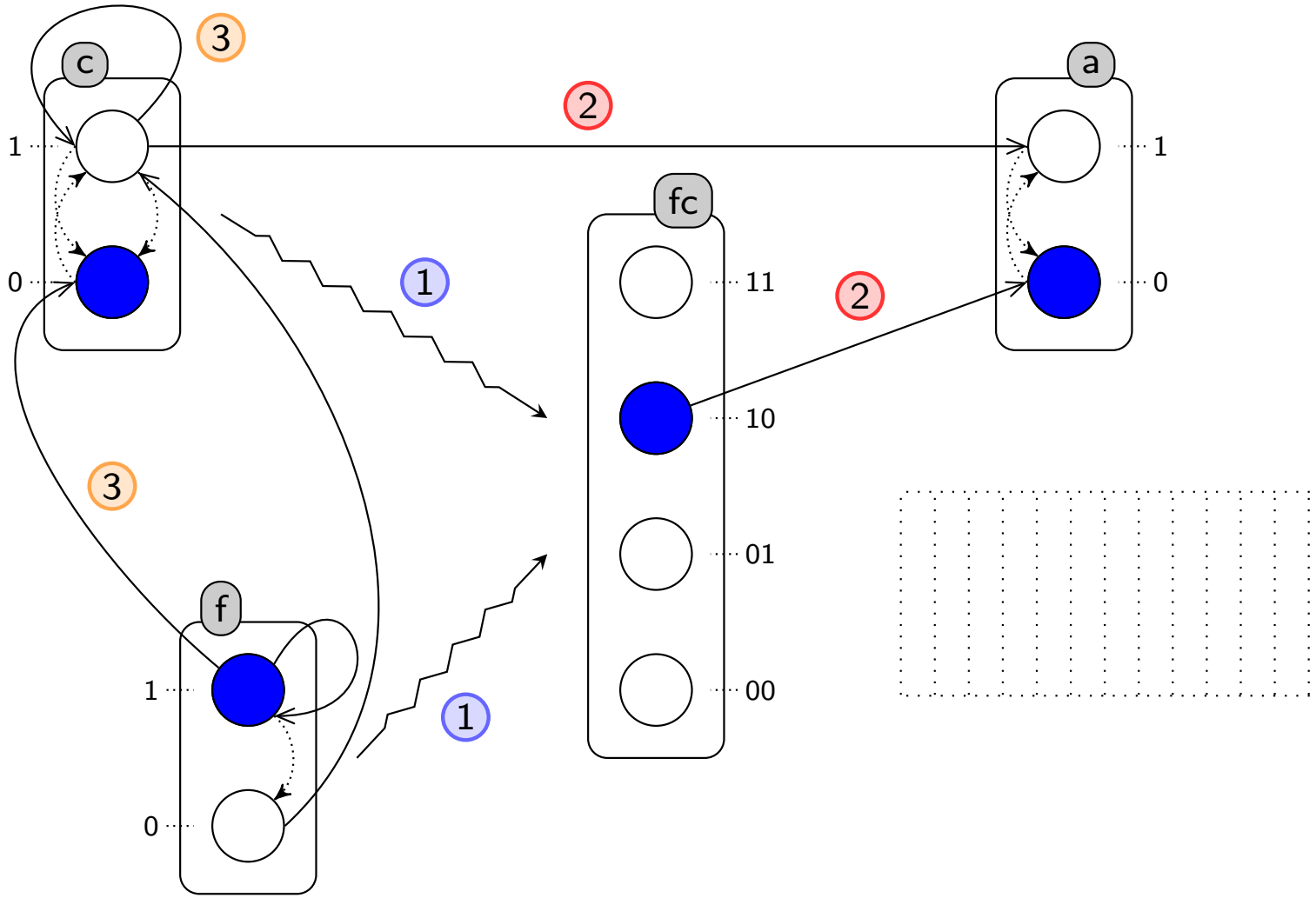
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



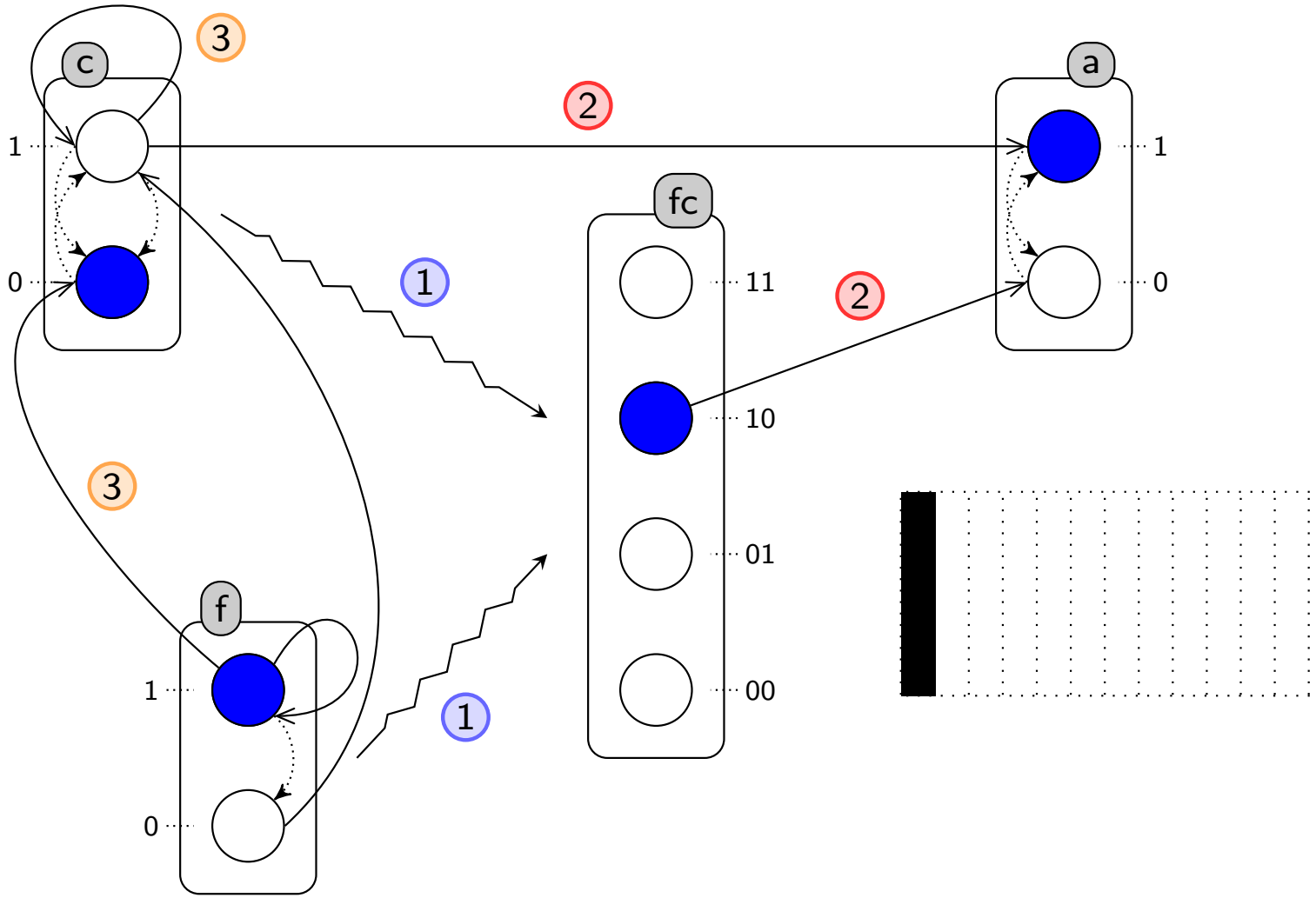
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



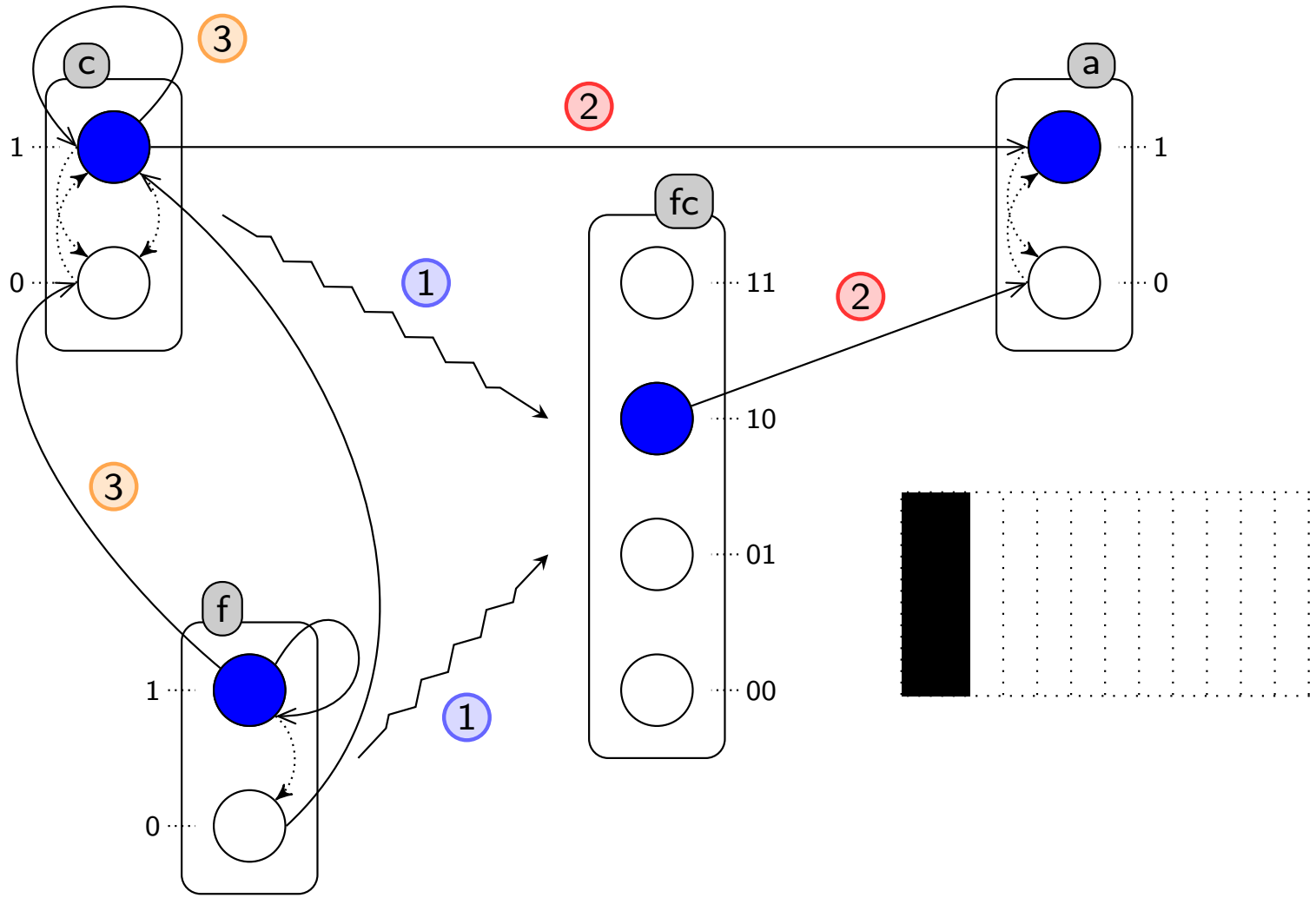
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



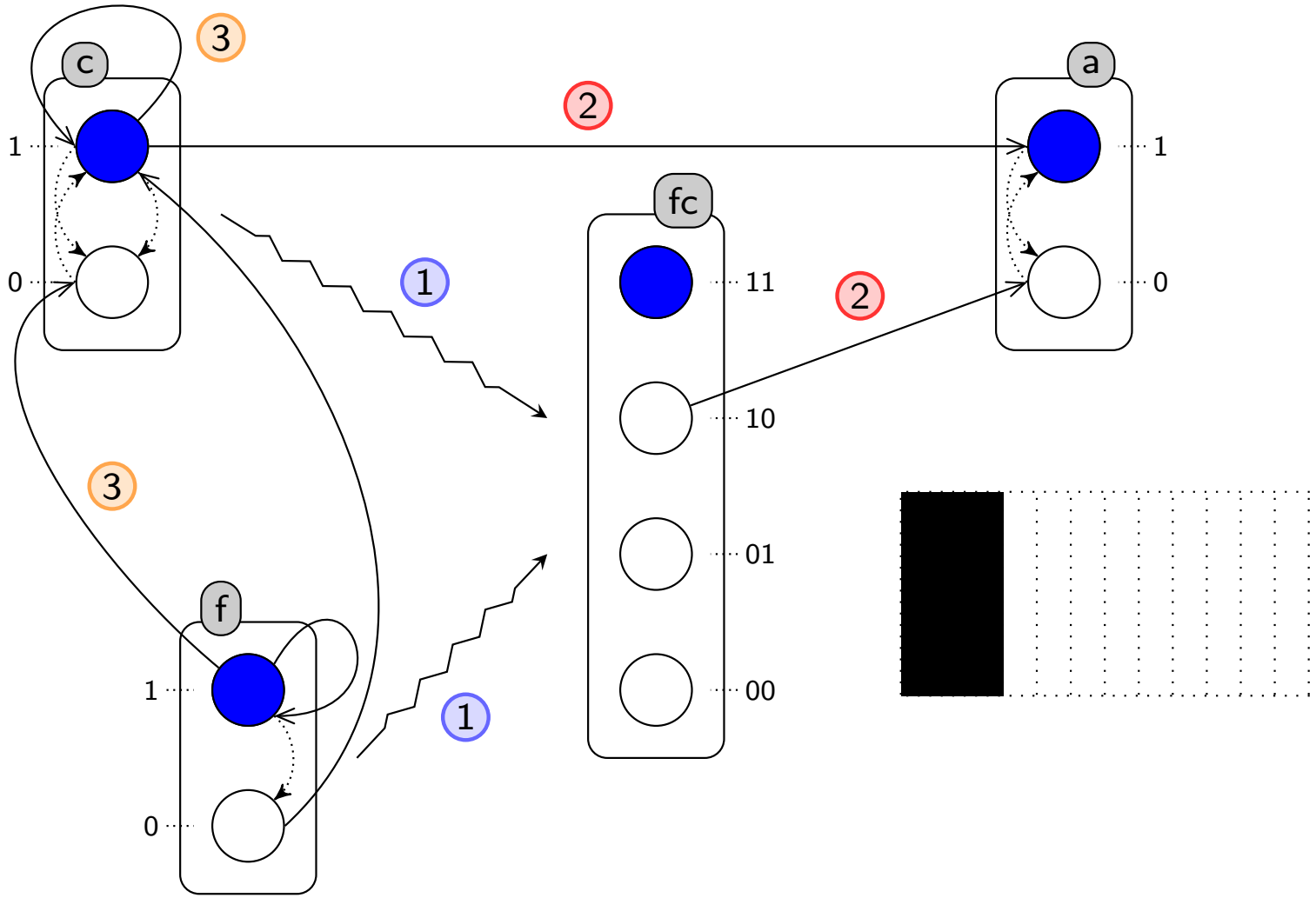
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



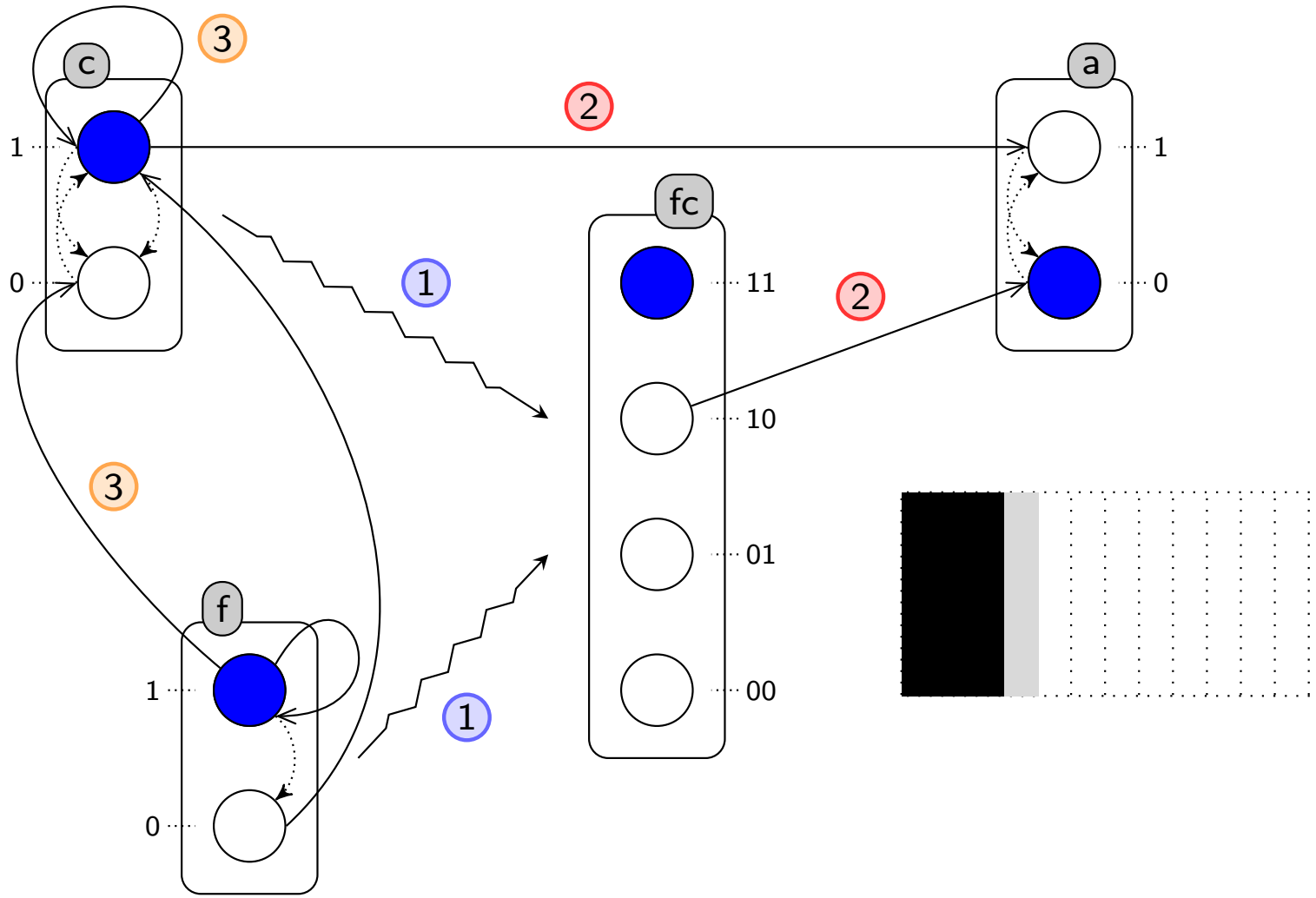
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



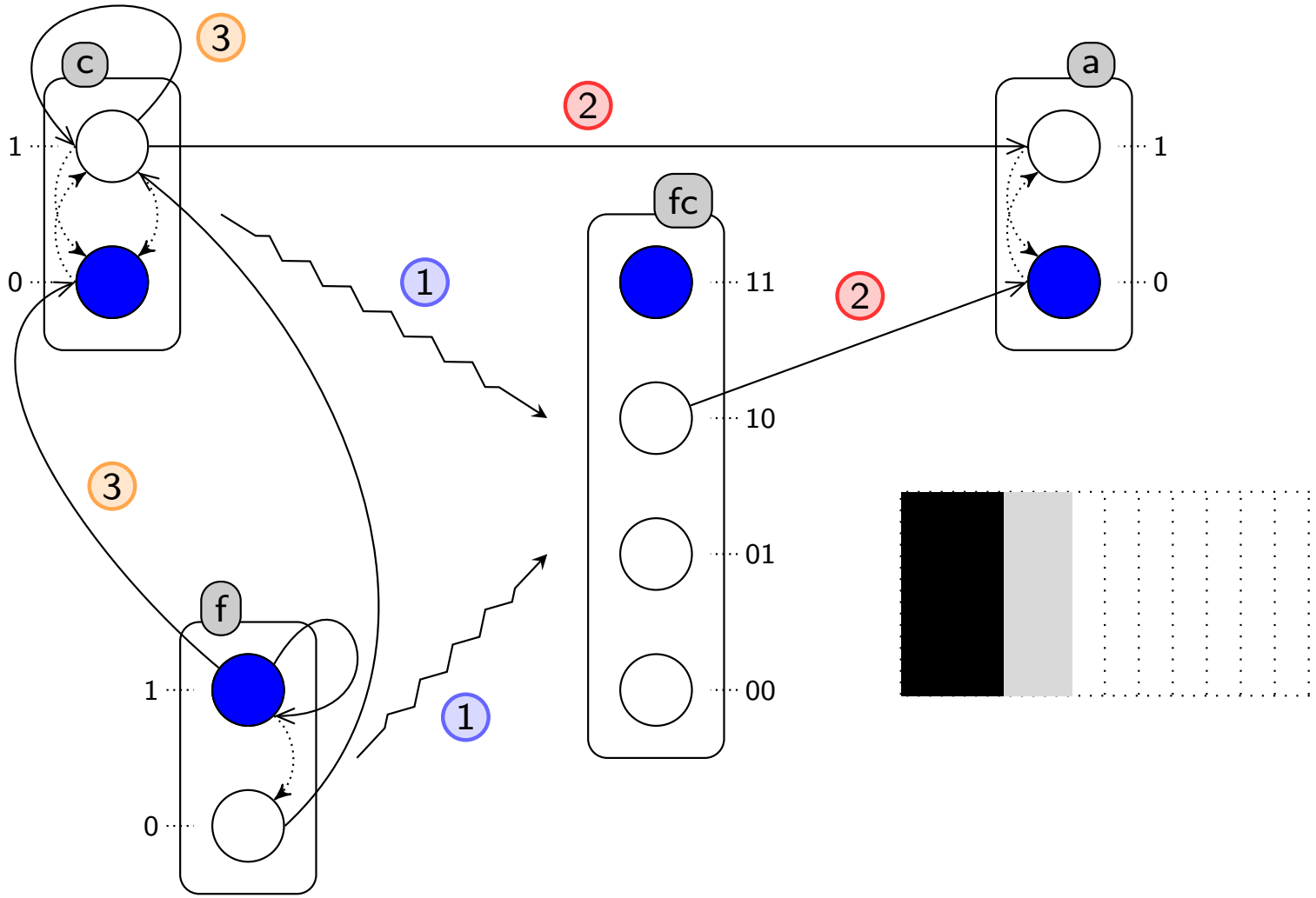
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



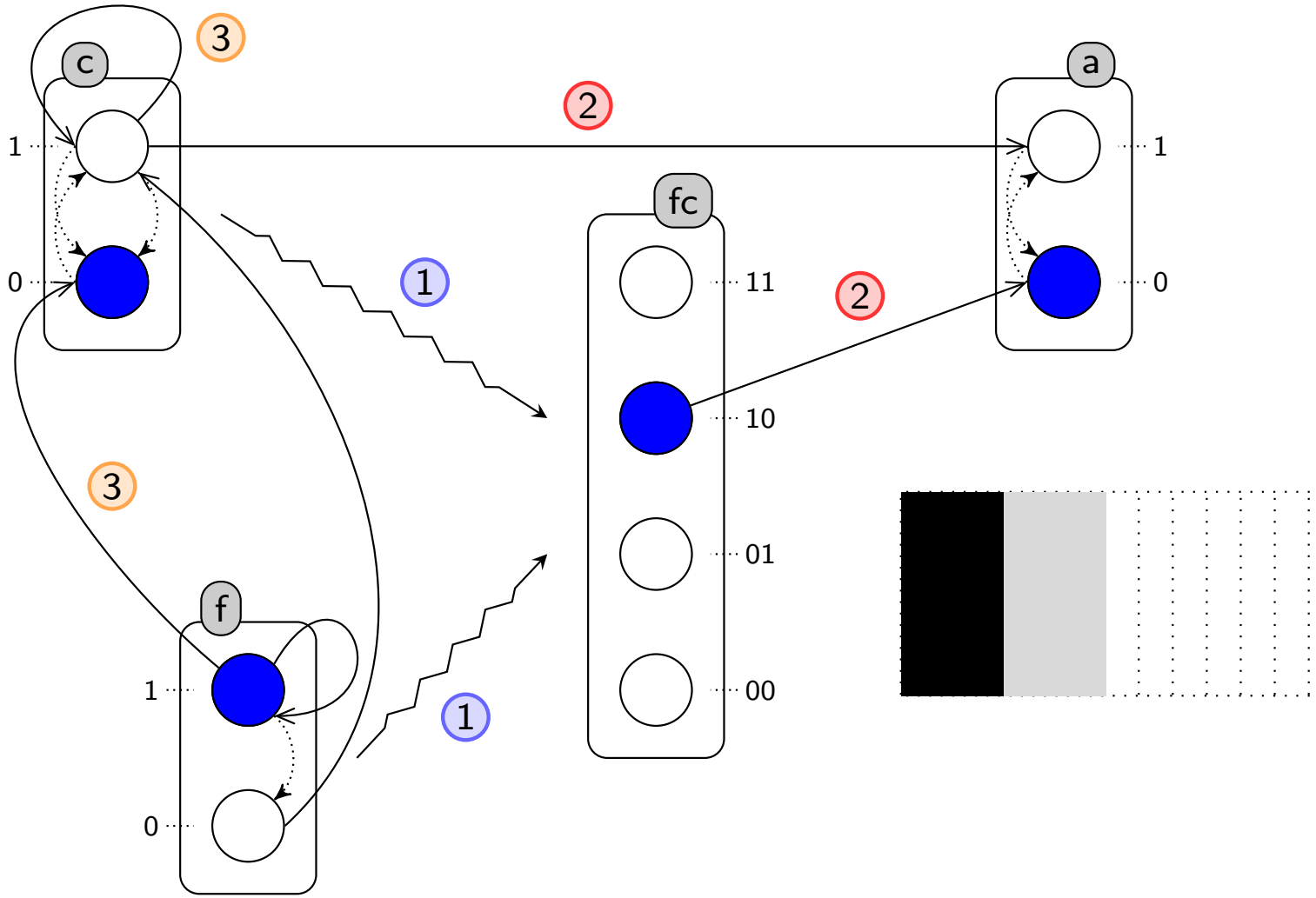
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



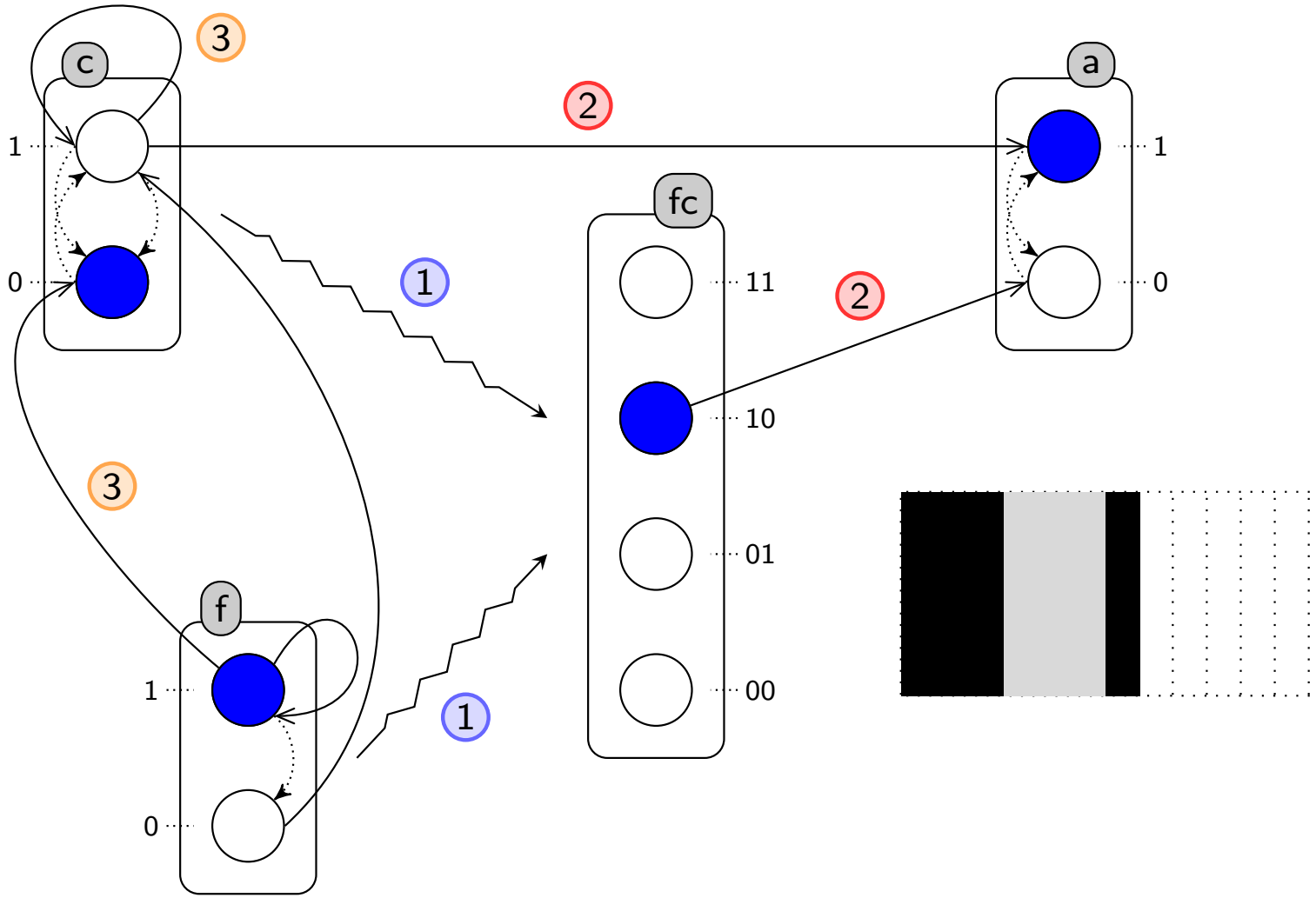
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



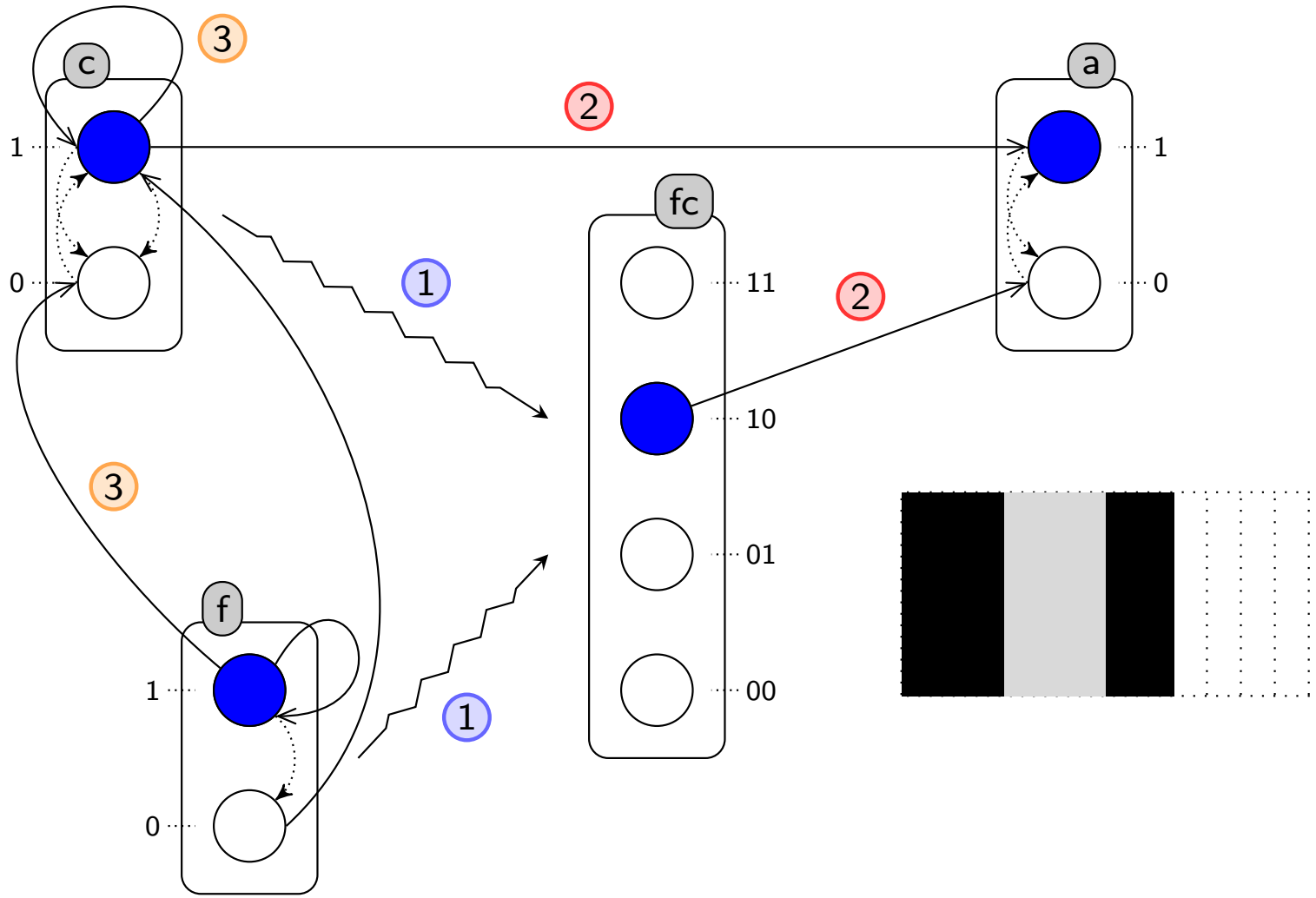
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



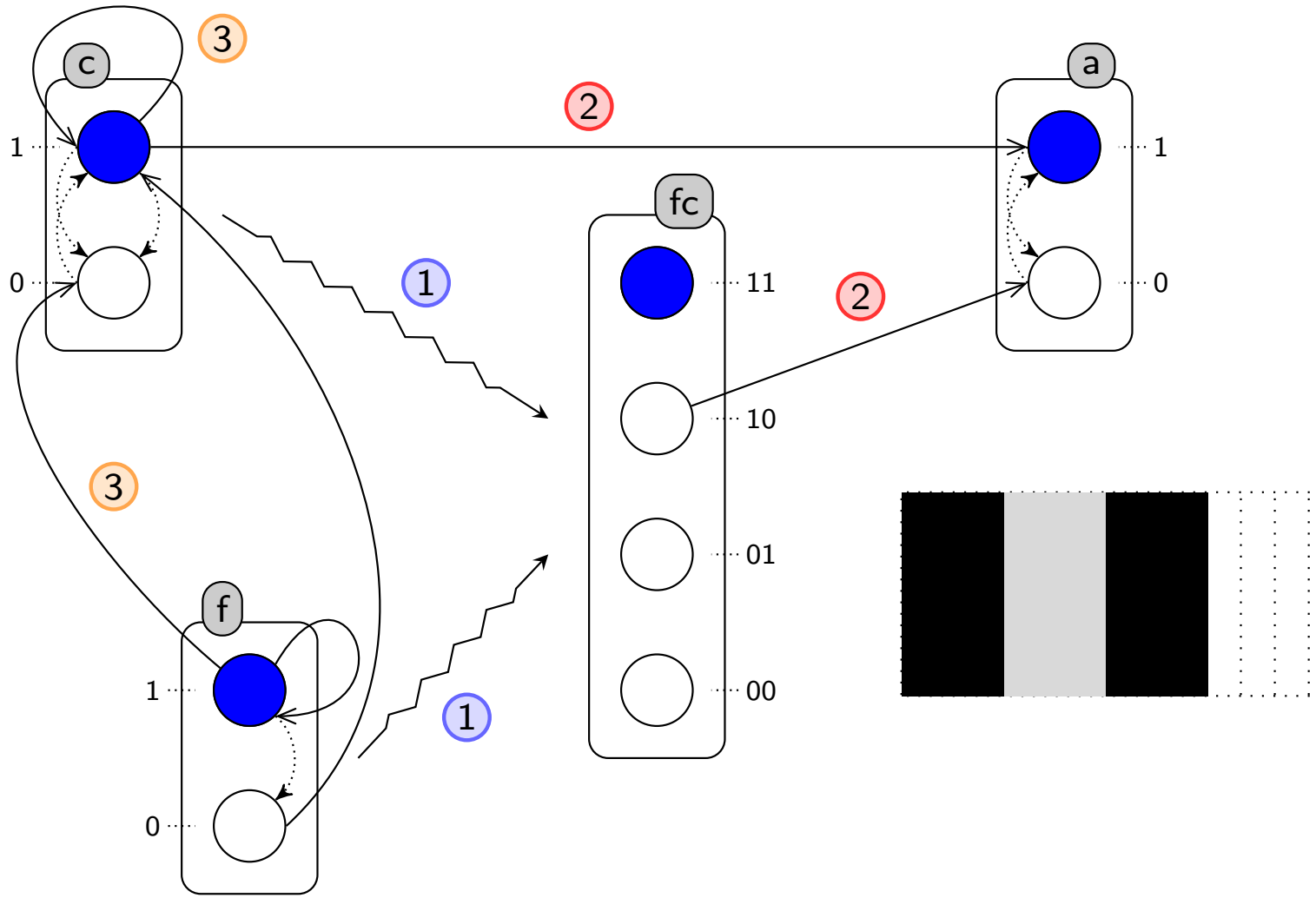
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



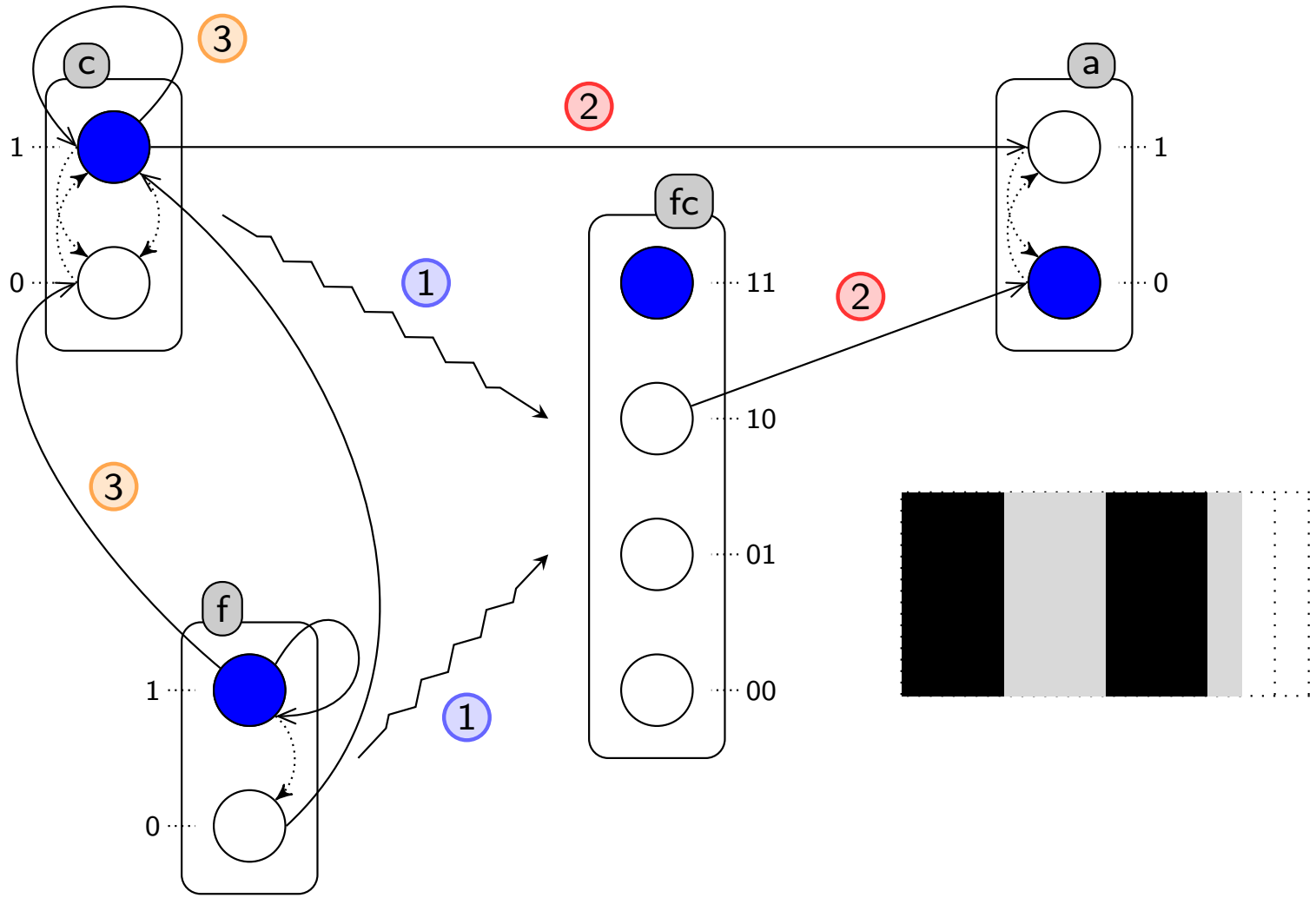
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



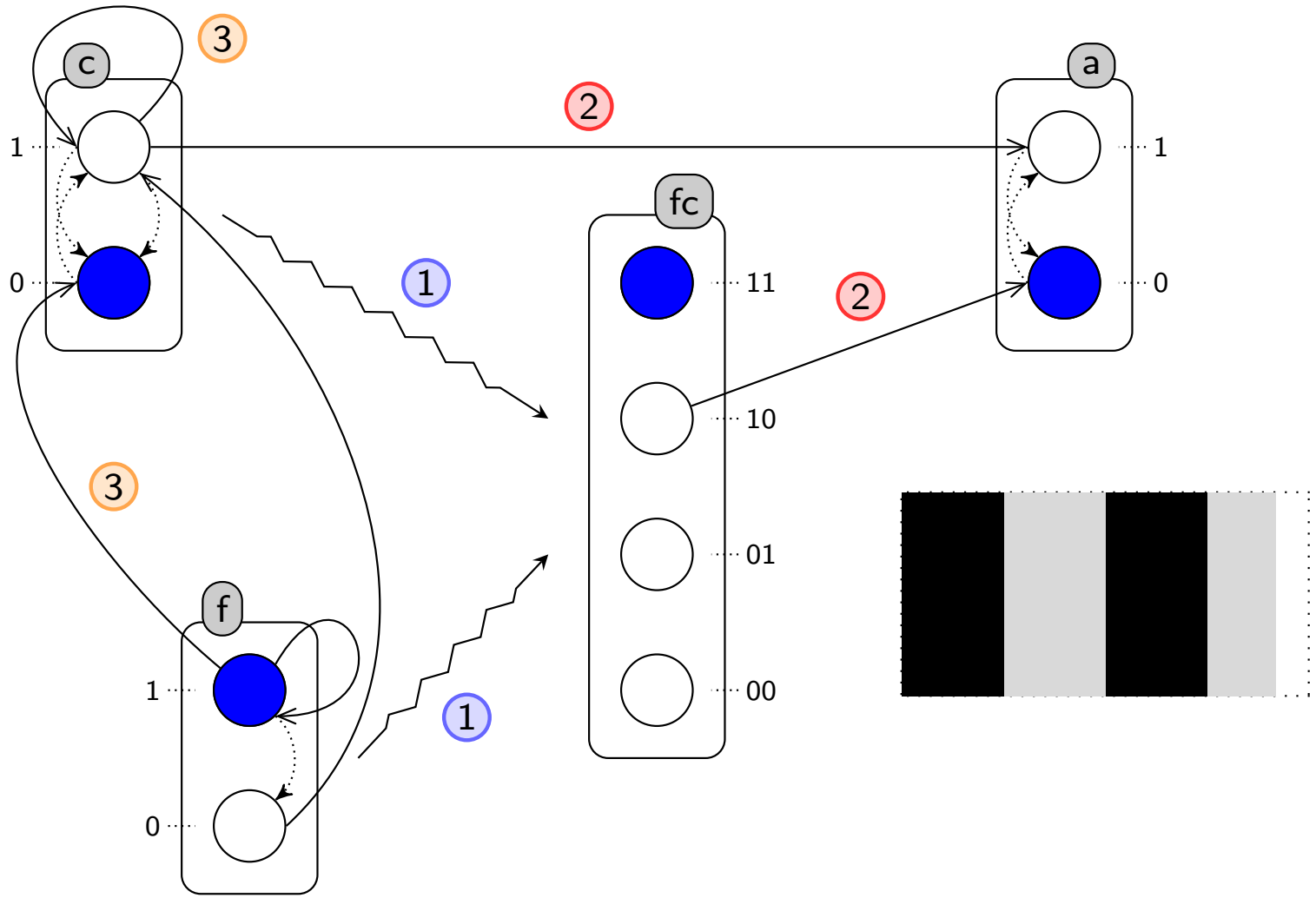
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



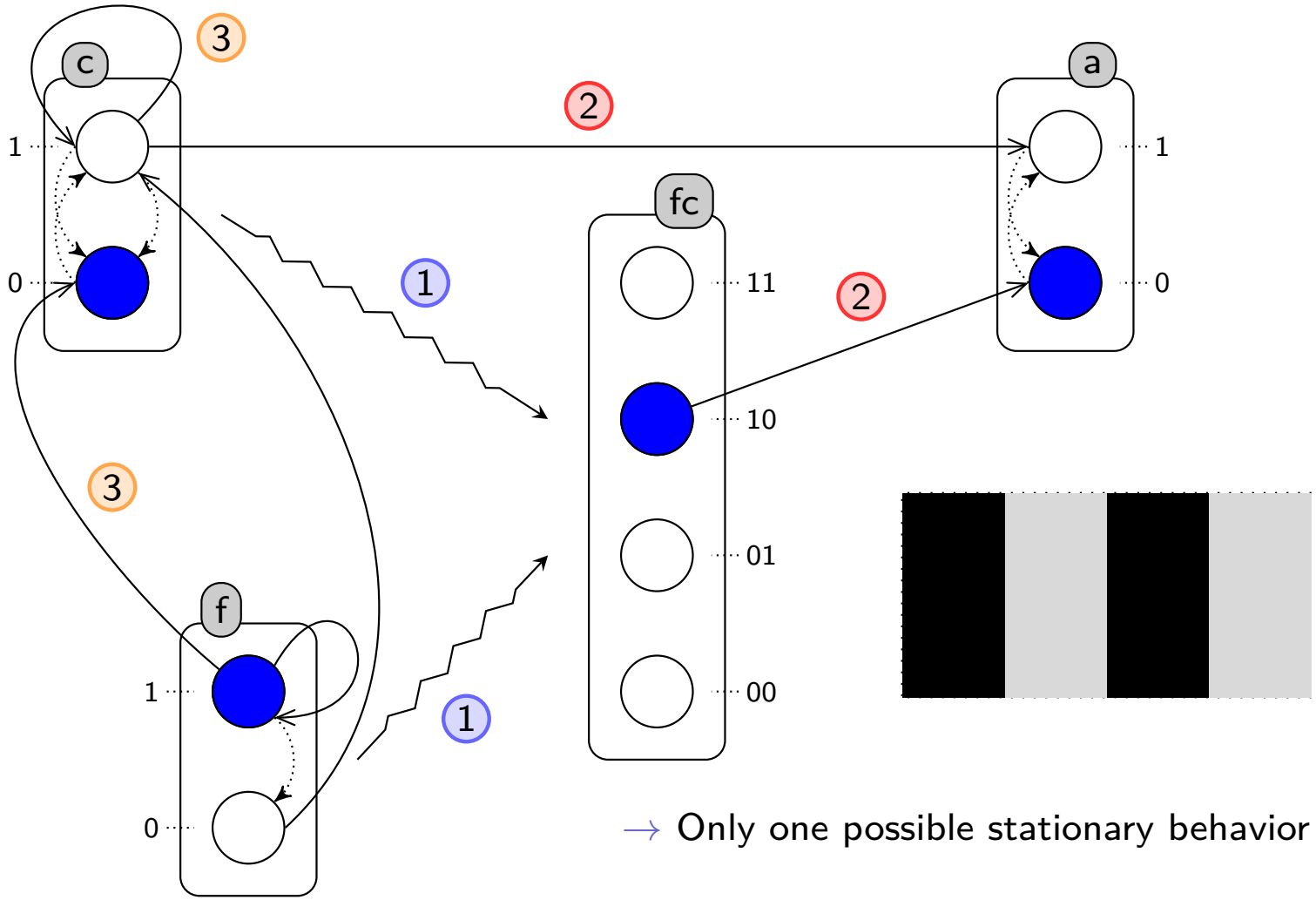
Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Use of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

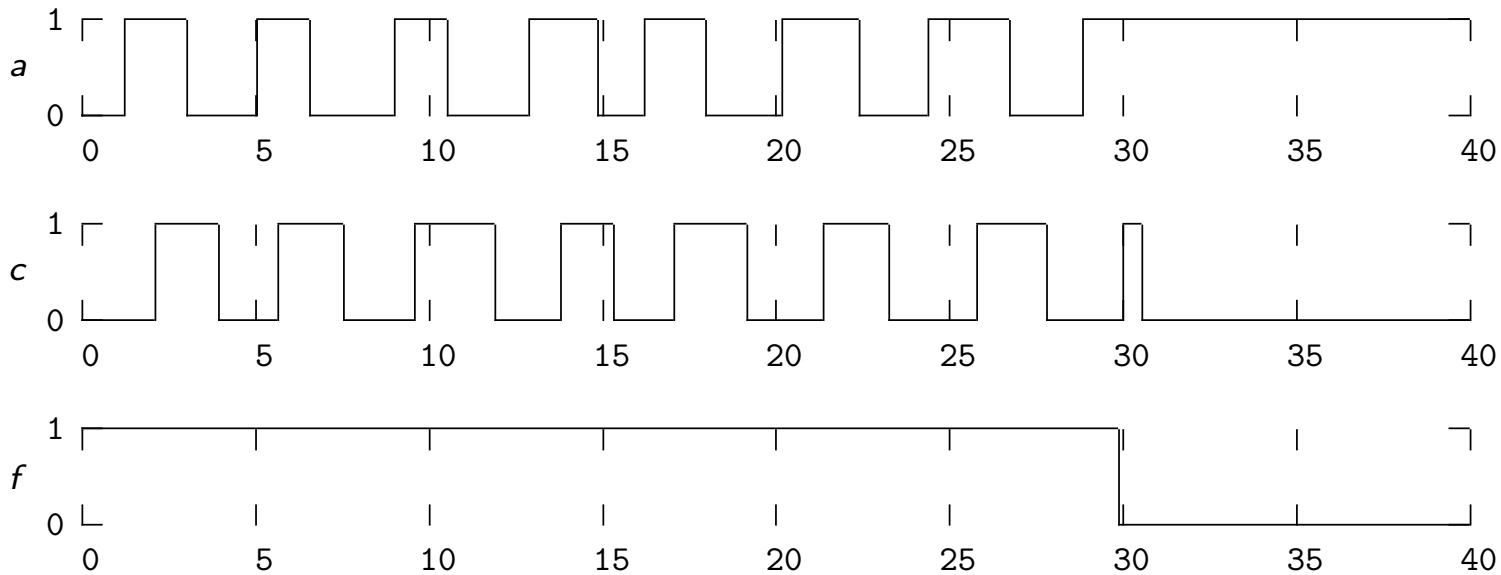


→ Only one possible stationary behavior

Abstraction of Temporal Parameters

[Paulevé (PhD thesis), 2011]

- Simulation with stochastic parameters:

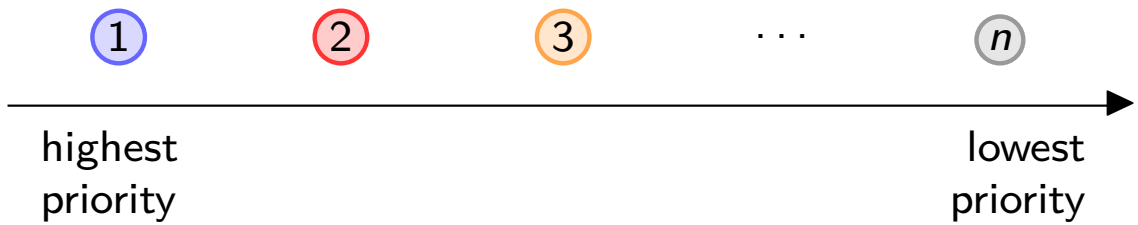


- Other possible analysis: stochastic model checkers (PRISM)
  → But combinatoric explosion: PRISM fails for more than 5 components

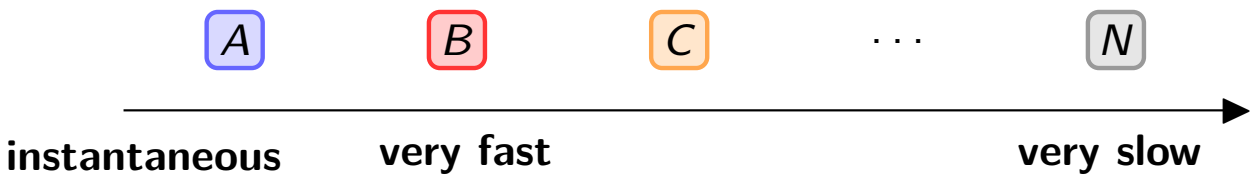
Addition of classes of priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

- Each action is associated to a discrete priority
- An action is playable only if no other action with higher priority is playable

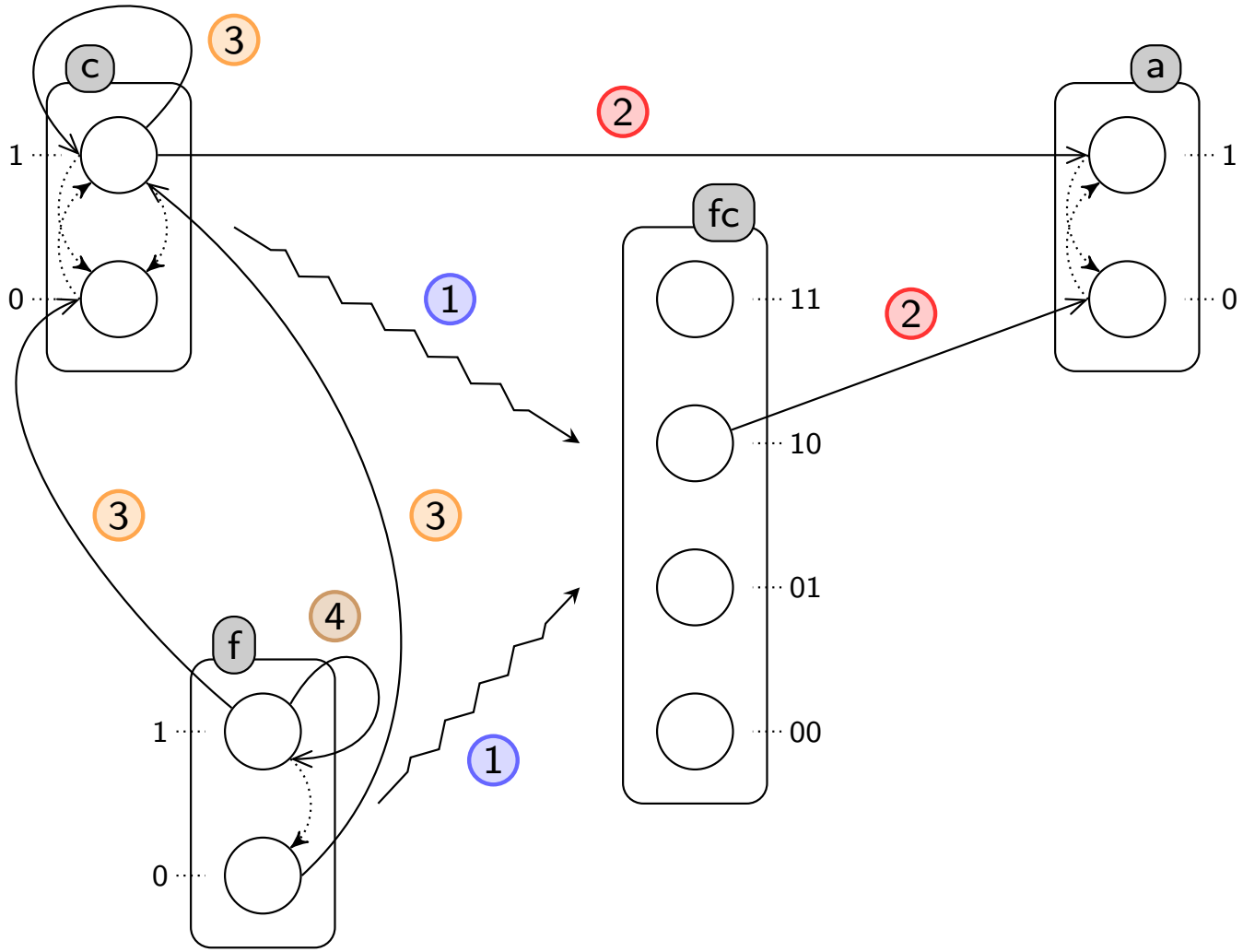


- Allow to model classes of actions with similar speeds or temporal parameters



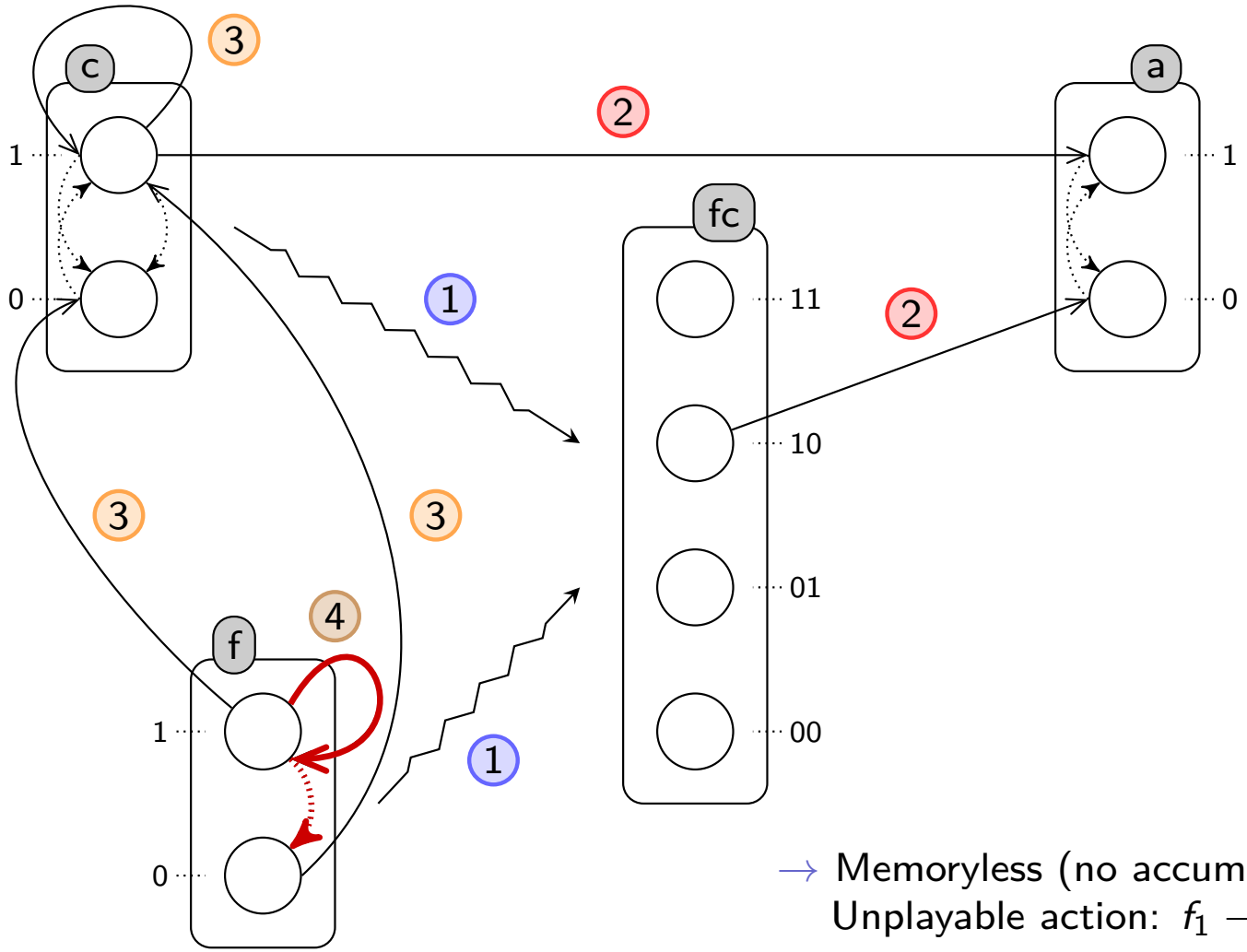
Limitation of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



Limitation of the Classes of Priorities

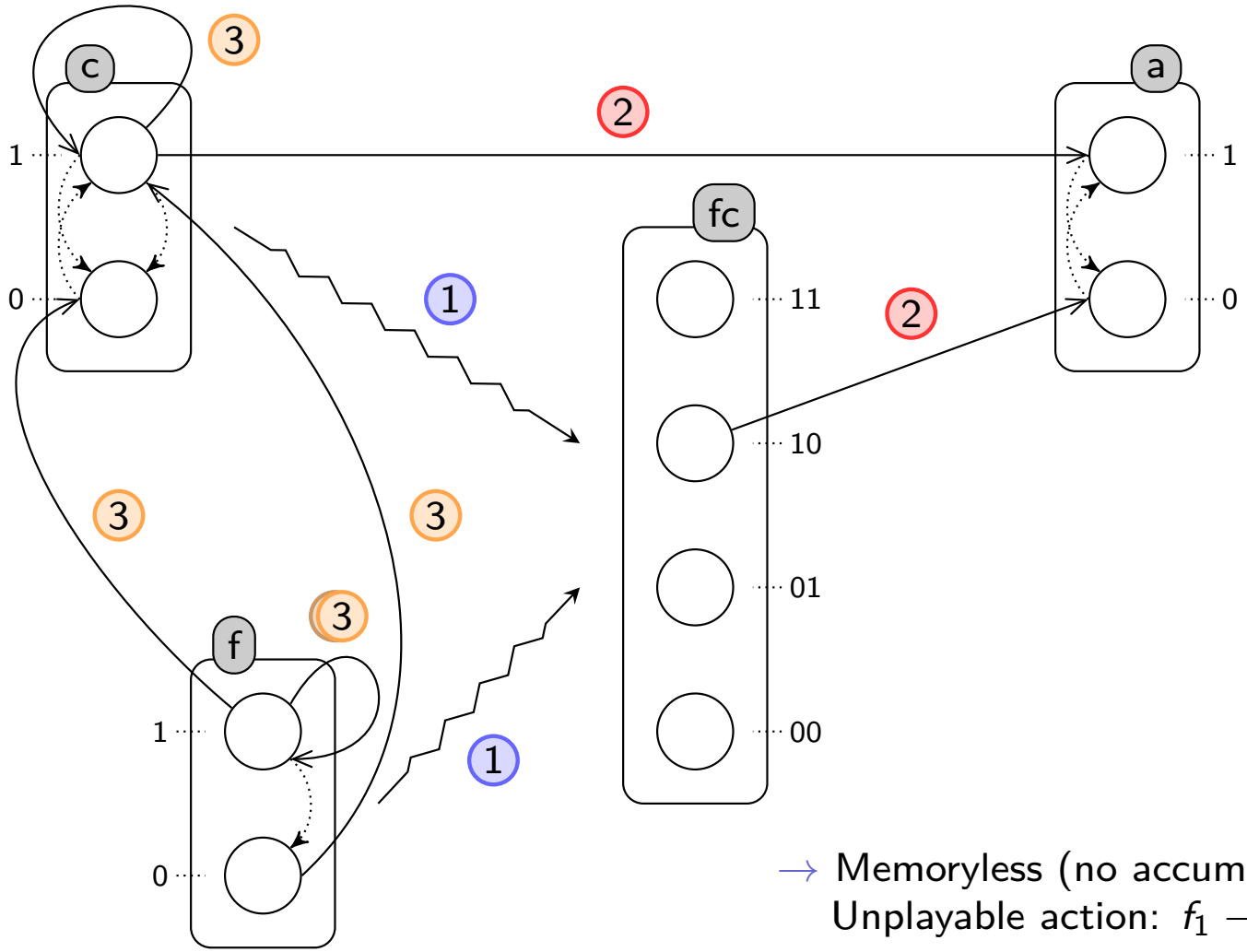
[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]



→ Memoryless (no accumulation)
Unplayable action: $f_1 \rightarrow f_1 \not\vdash f_0$

Limitation of the Classes of Priorities

[Folschette *et al.* in *Workshop on Interactions between Computer Science and Biology*, 2013]

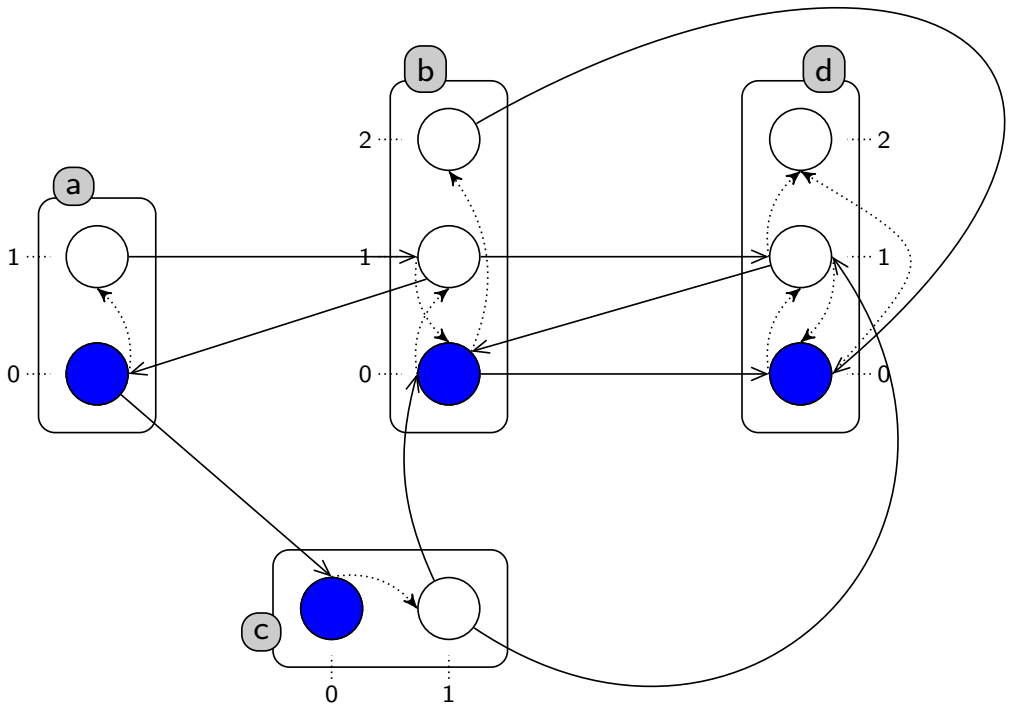


→ Memoryless (no accumulation)
Unplayable action: $f_1 \rightarrow f_1 \not\vdash f_0$

Static analysis: successive reachability

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

Successive reachability of processes:



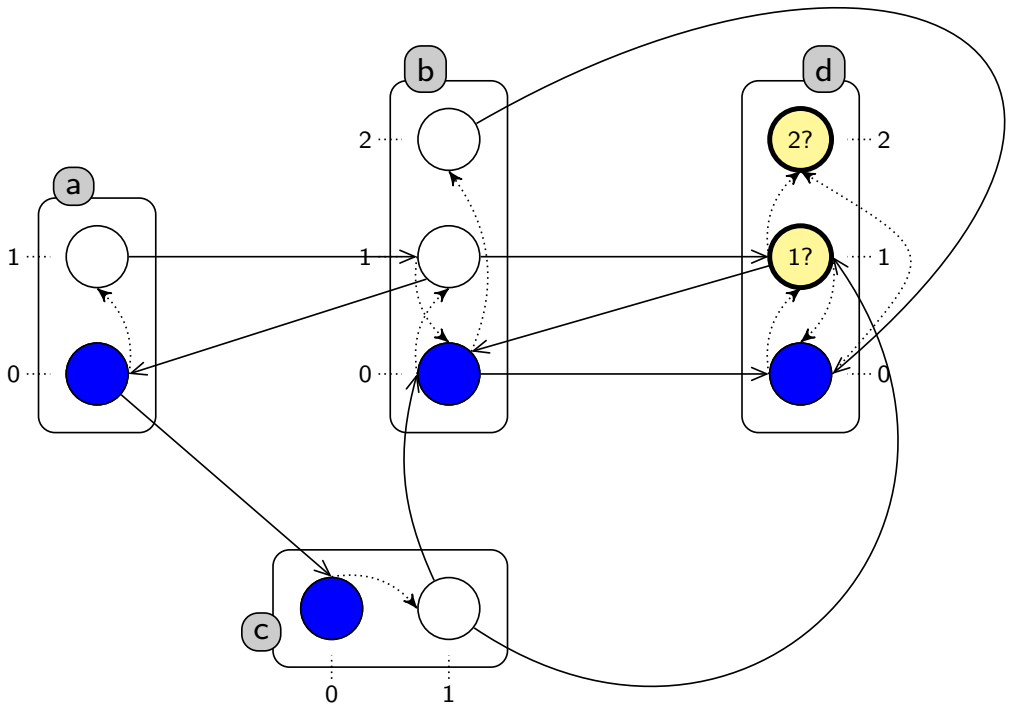
- Initial state

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

Static analysis: successive reachability

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

Successive reachability of processes:

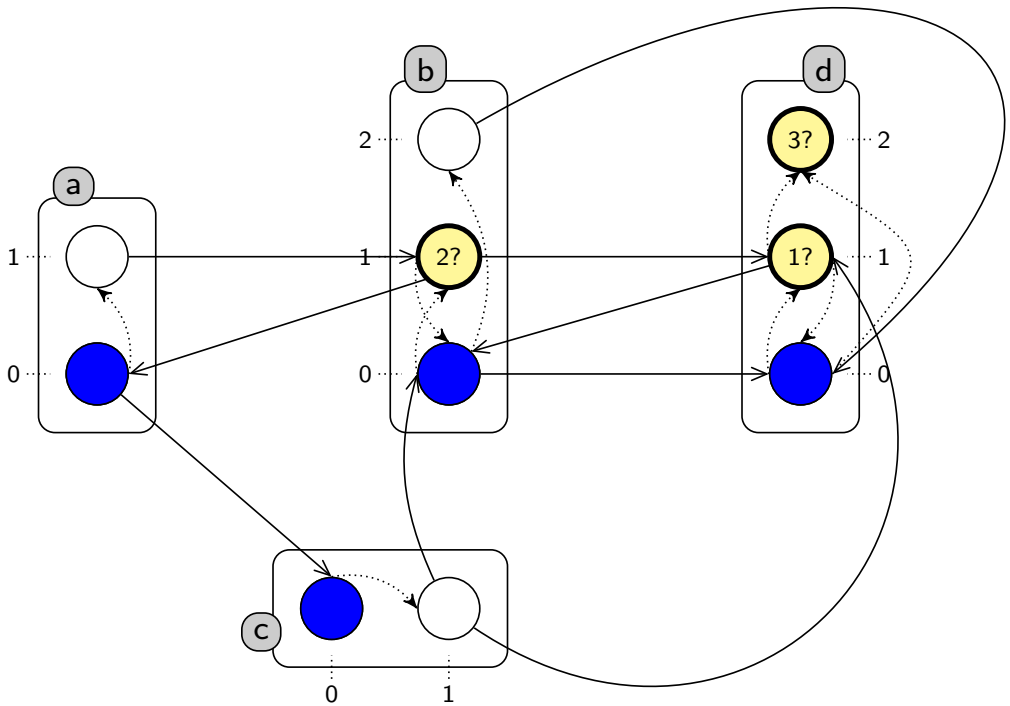


- Initial state $\langle a_1, b_0, c_0, d_0 \rangle$
- Objectives $[\ \Rsh d_1 \ :: \Rsh d_2\ ]$

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

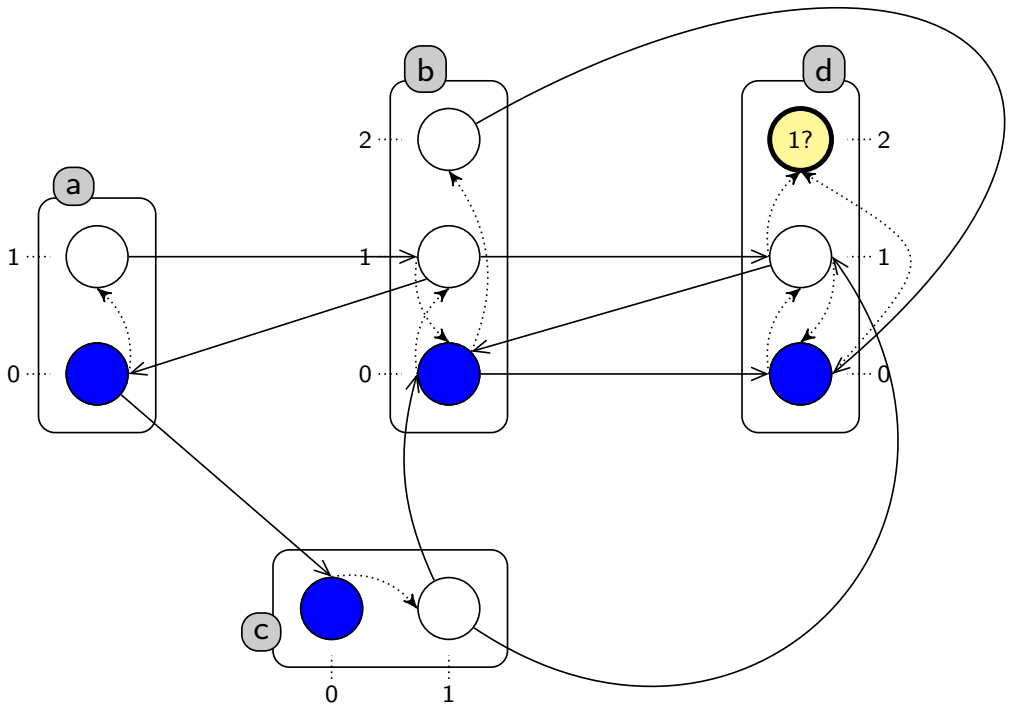
$$[\ \Rsh\ d_1\ \ ::\ \Rsh\ d_2\ ]$$

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

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

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

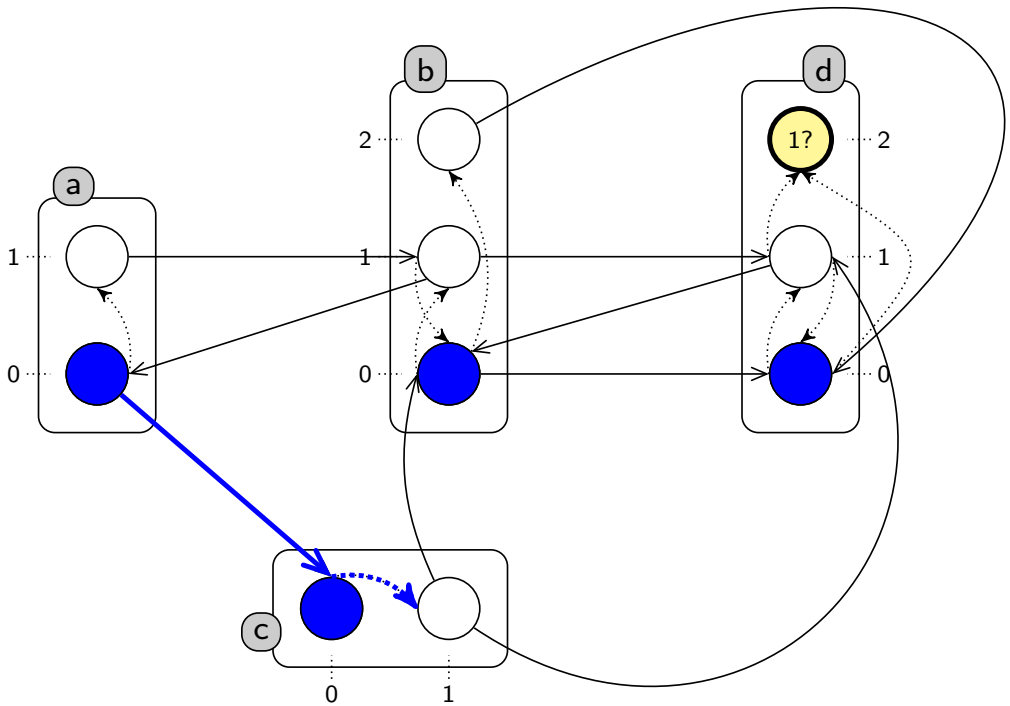
$$[\ \Rsh\ d_1 :: \Rsh\ b_1 :: \Rsh\ d_2\ ]$$

$$[\ \Rsh\ d_2\ ]$$

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

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

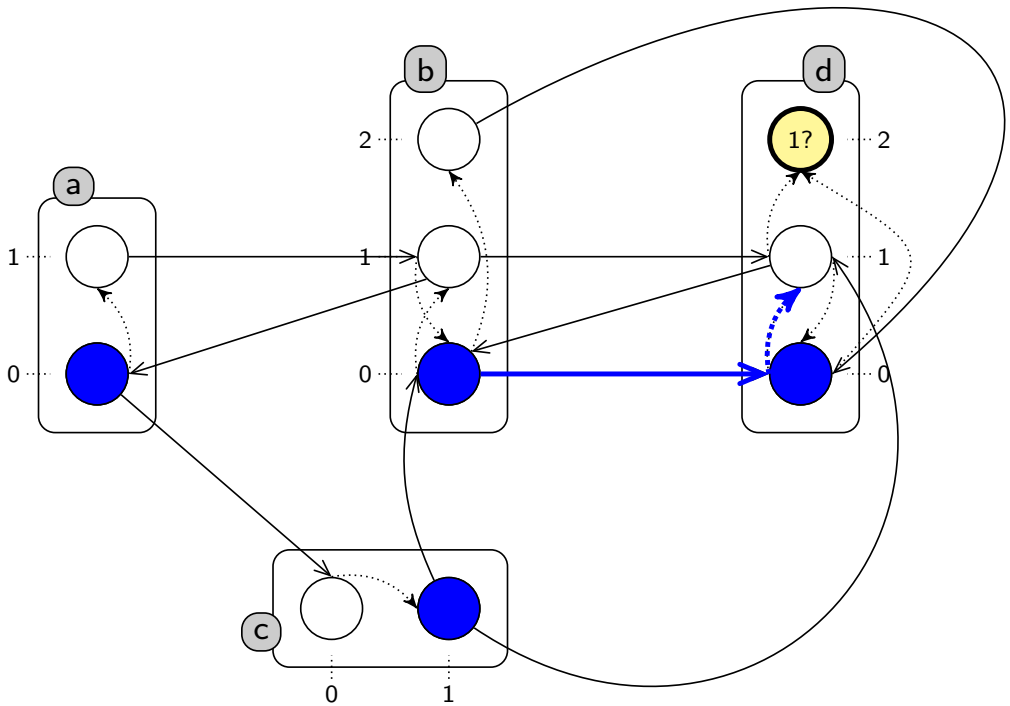
$\rightarrow$ Concretization of the objective = scenario

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

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

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

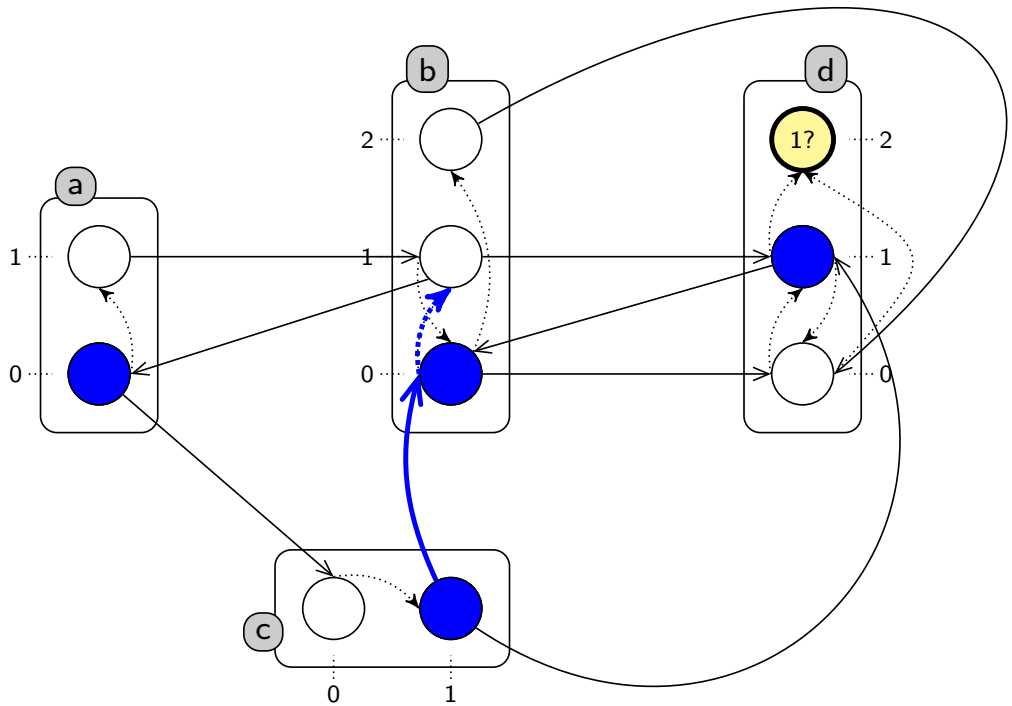
$\rightarrow$ Concretization of the objective = scenario

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

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

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

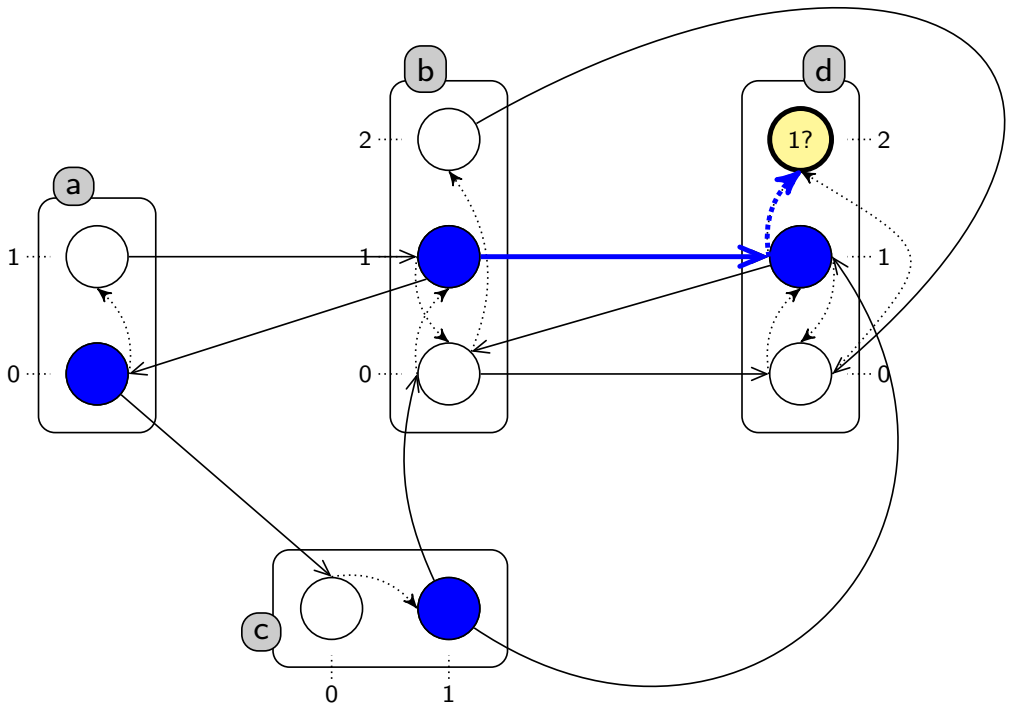
$\rightarrow$ Concretization of the objective = scenario

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

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

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

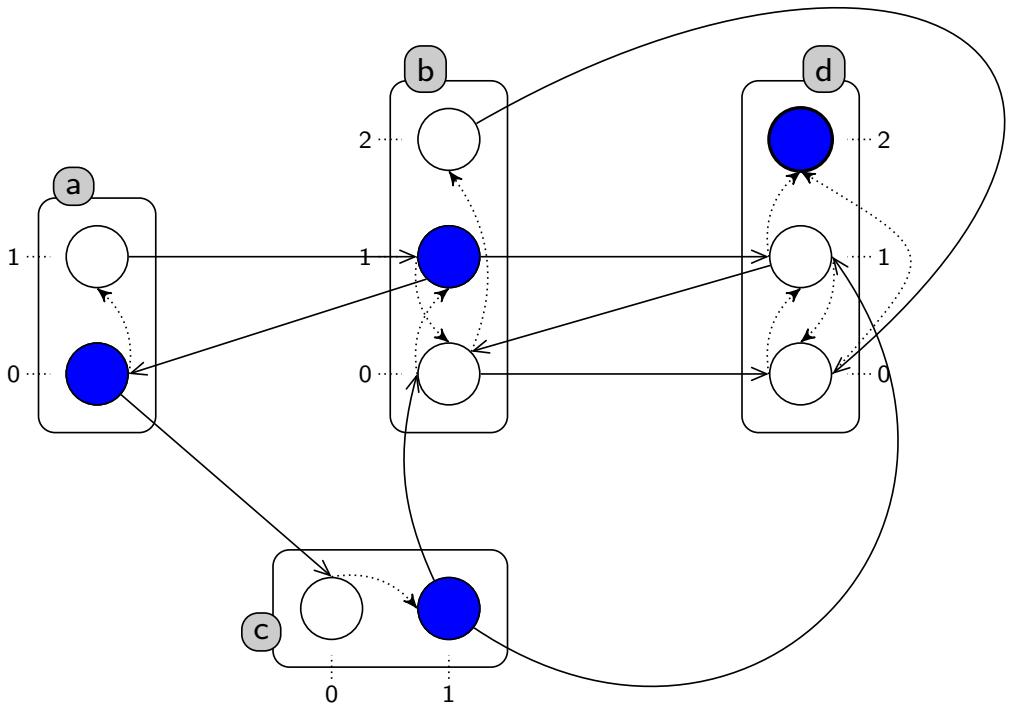
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

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

Successive reachability of processes:



- Initial state

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

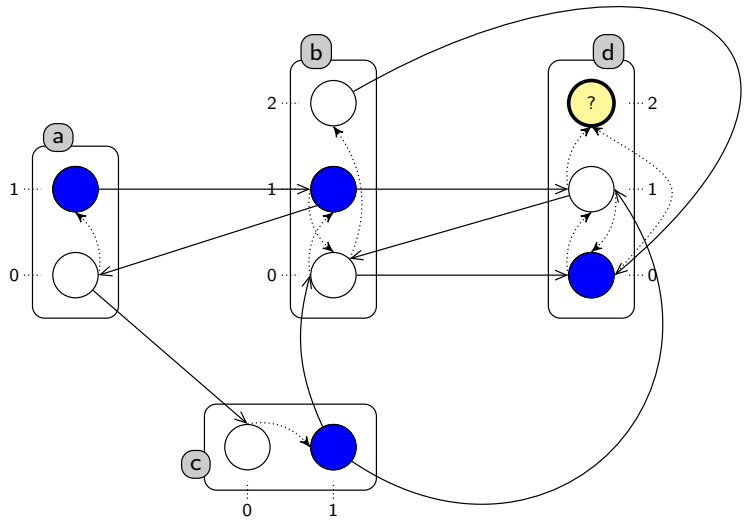
- Objectives

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

$\rightarrow$ Concretization of the objective = scenario

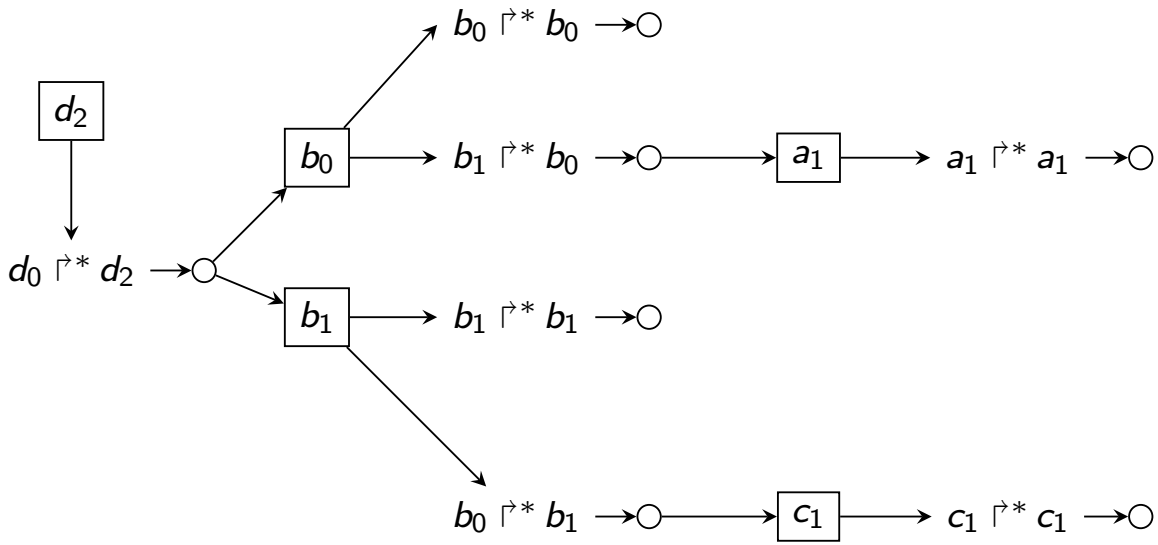
$$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$$

Under-approximation



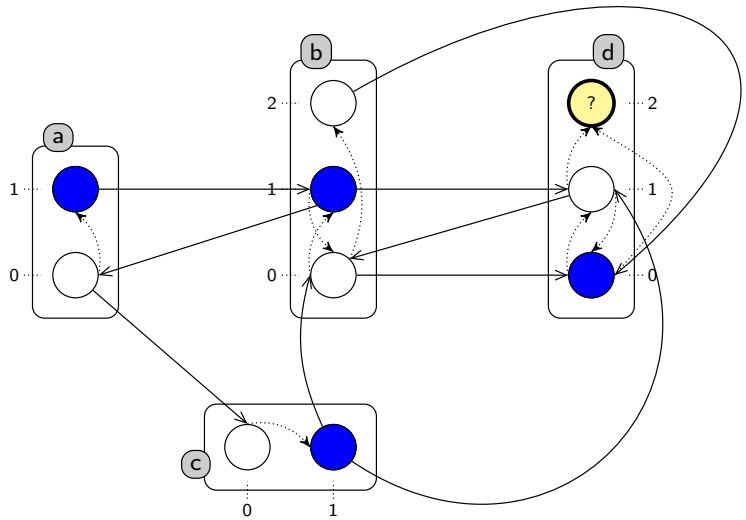
Sufficient condition:

- no cycle
- each objective has a solution



d_2	Required process
$d_0 \Gamma^* d_2$	Objective
○	Solution to an objective

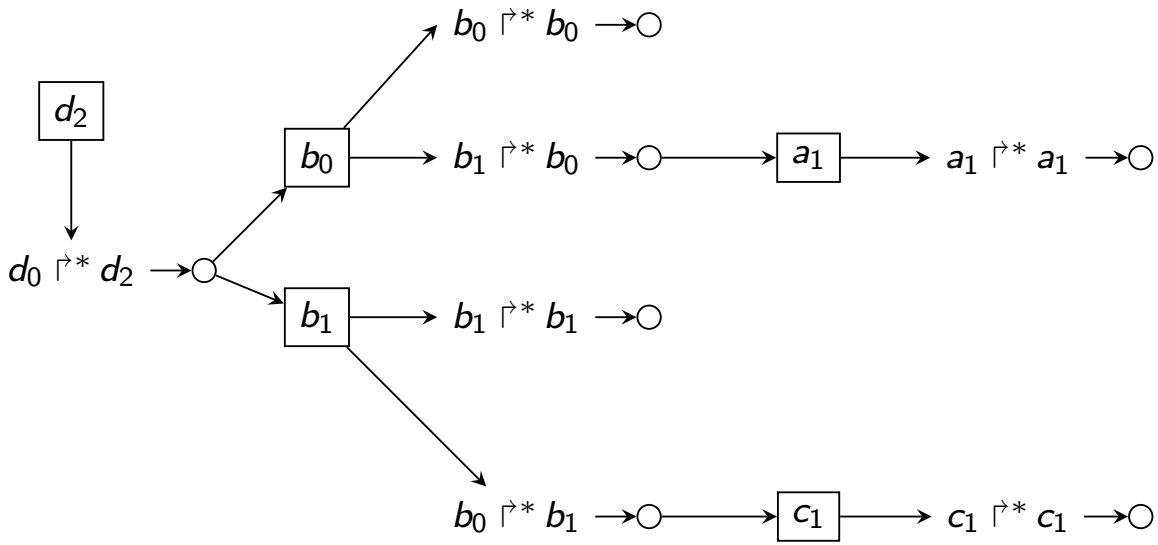
Under-approximation



Sufficient condition:

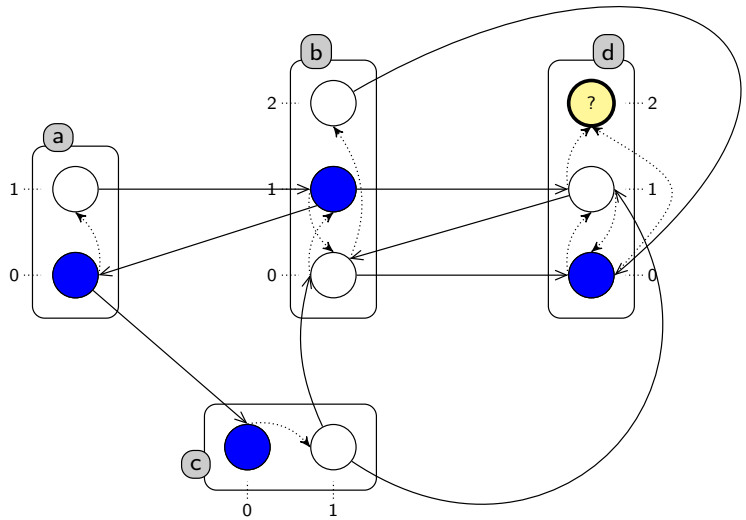
- no cycle
- each objective has a solution

R is **true**



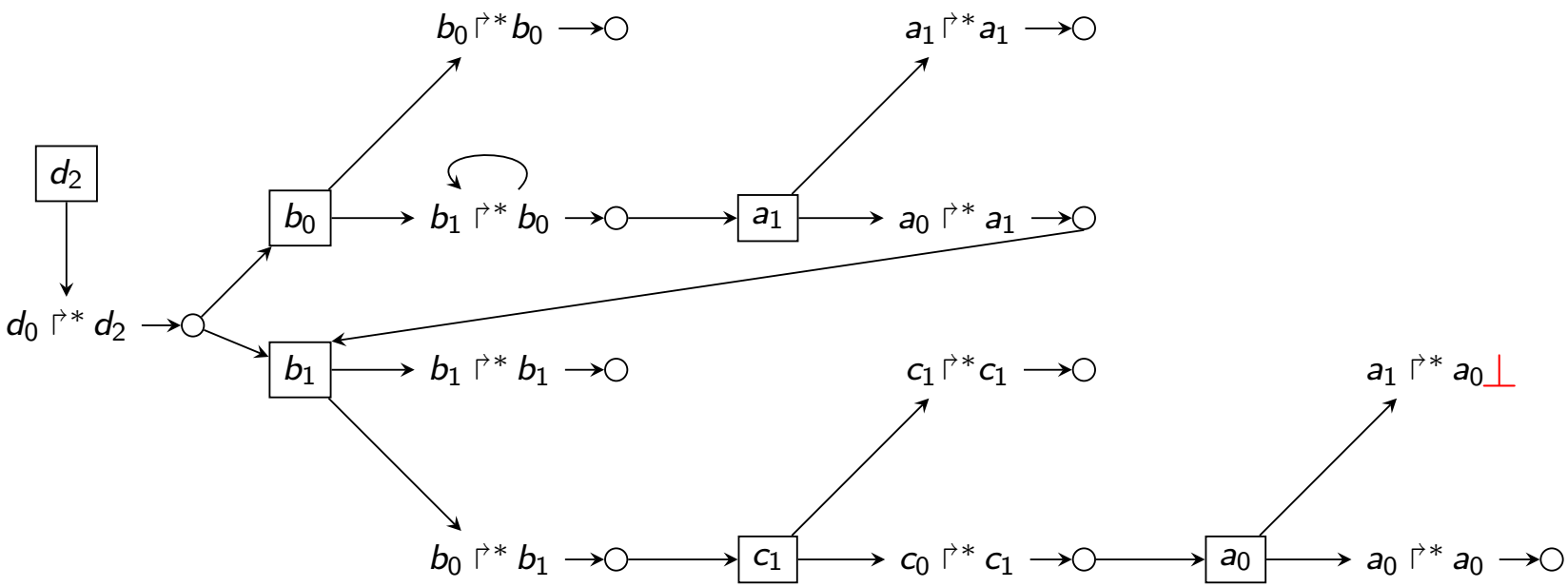
d_2	Required process
$d_0 \Gamma^* d_2$	Objective
○	Solution to an objective

Under-approximation

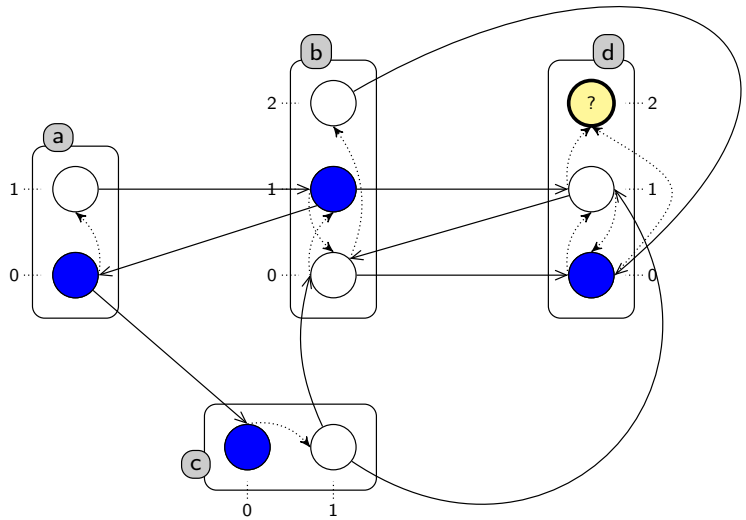


Sufficient condition:

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



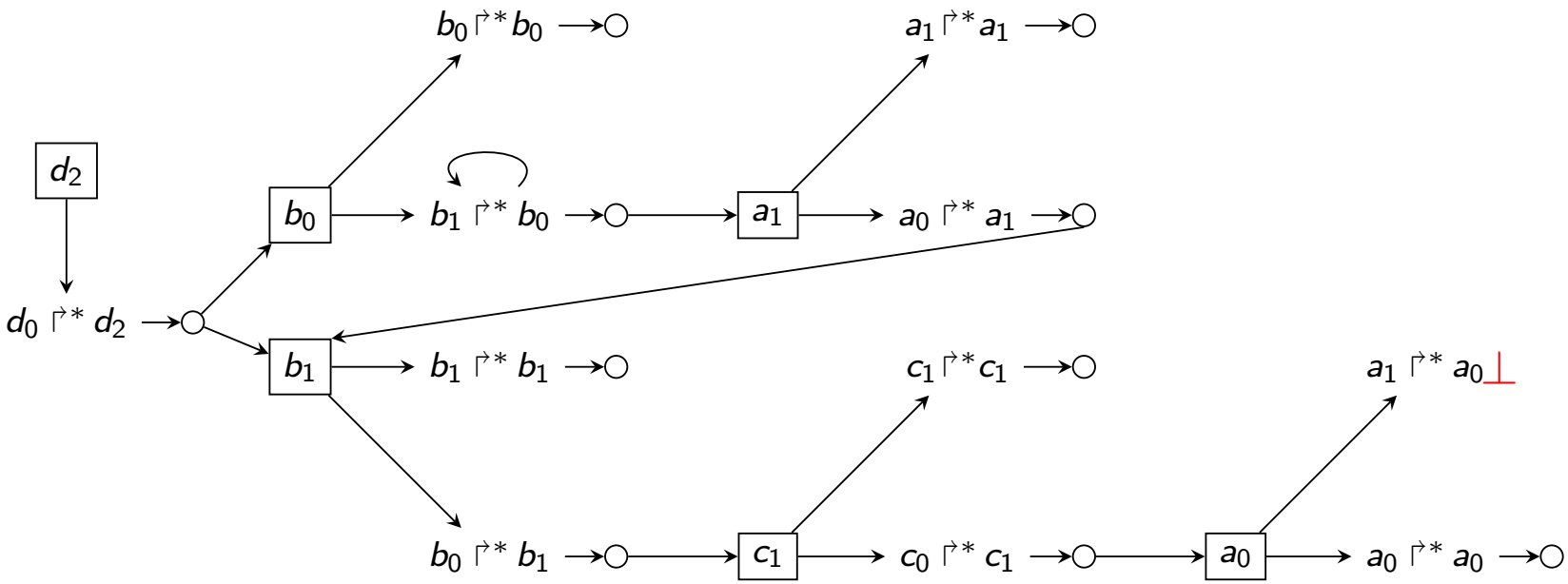
Under-approximation



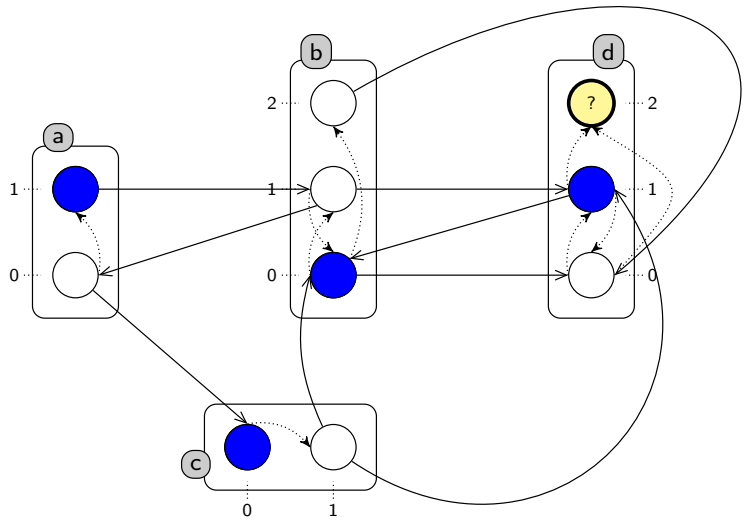
Sufficient condition:

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

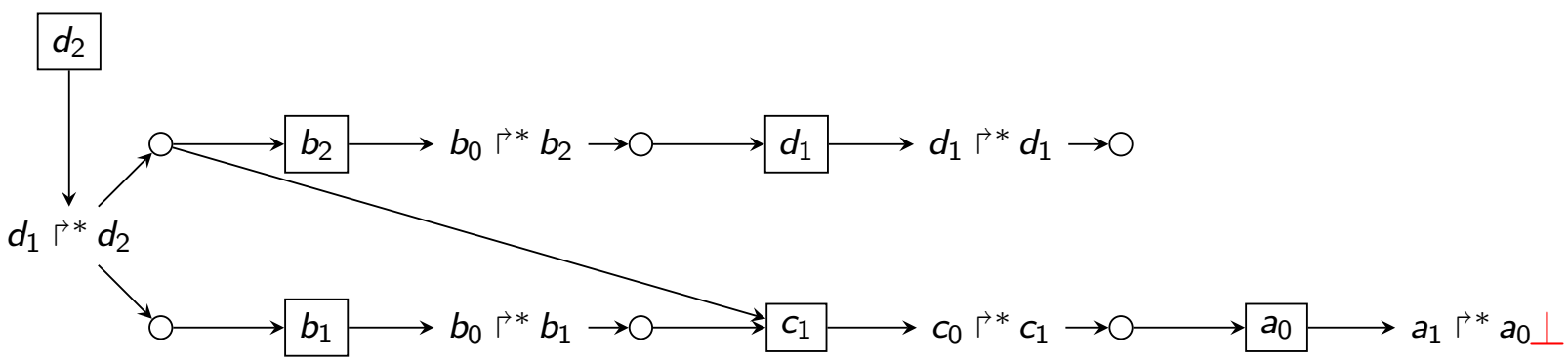
Inconclusive



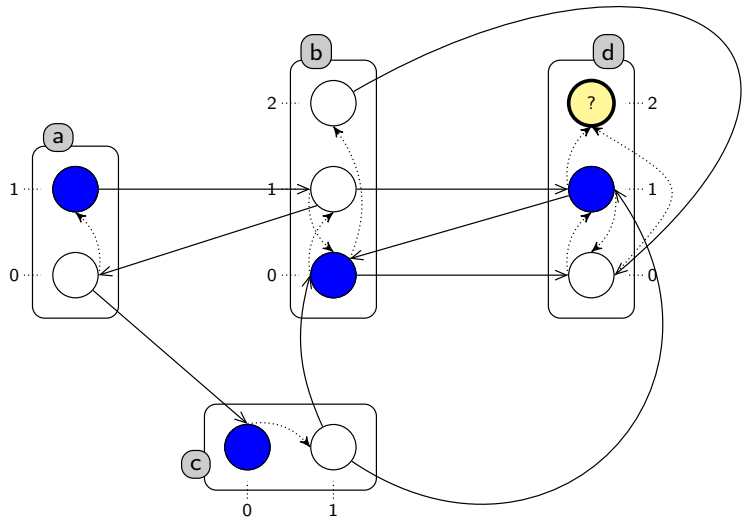
Over-approximation



Necessary condition:



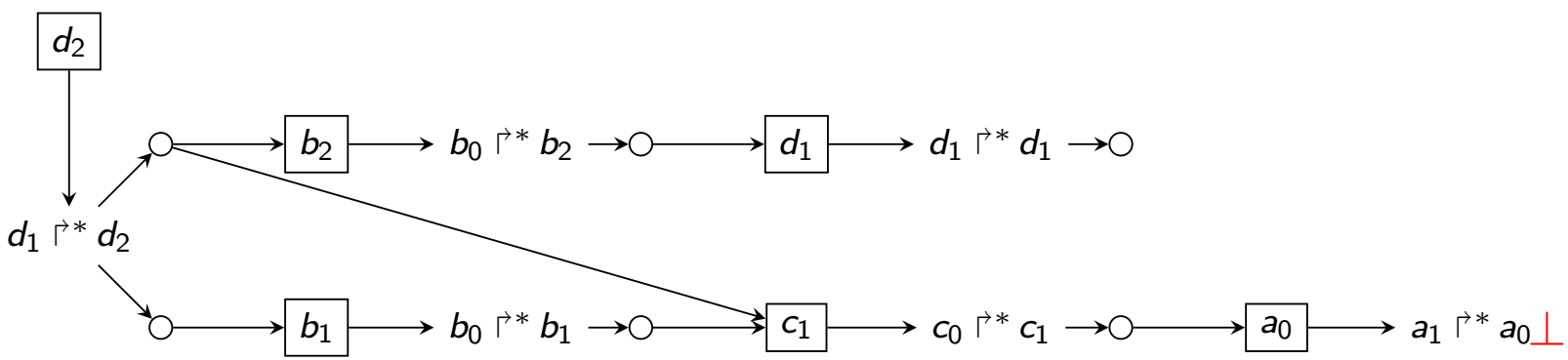
Over-approximation



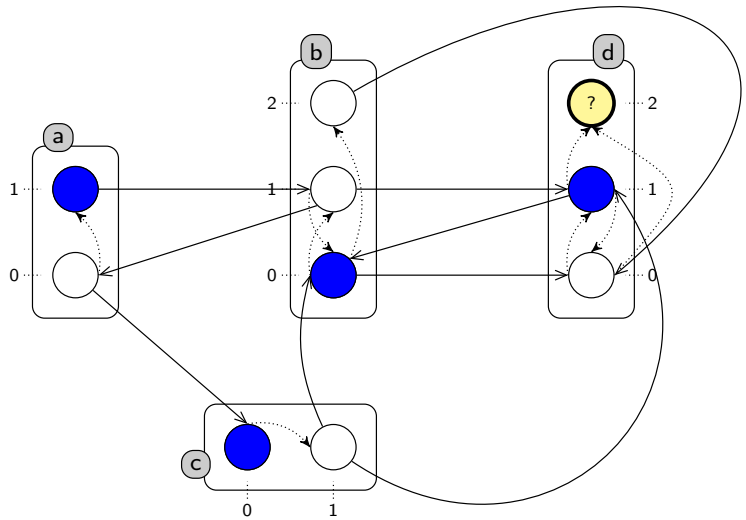
Necessary condition:

There exists a traversal with no cycle

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



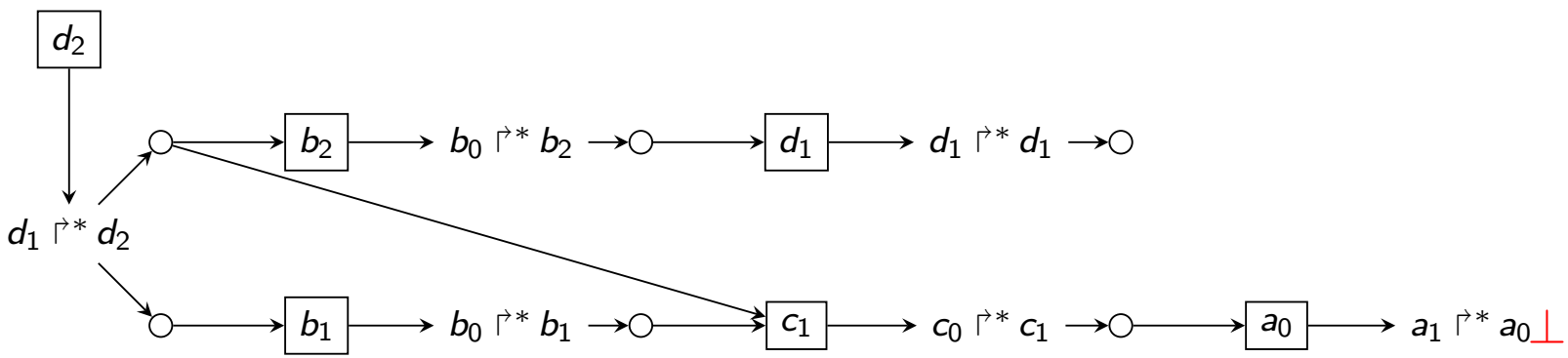
Over-approximation



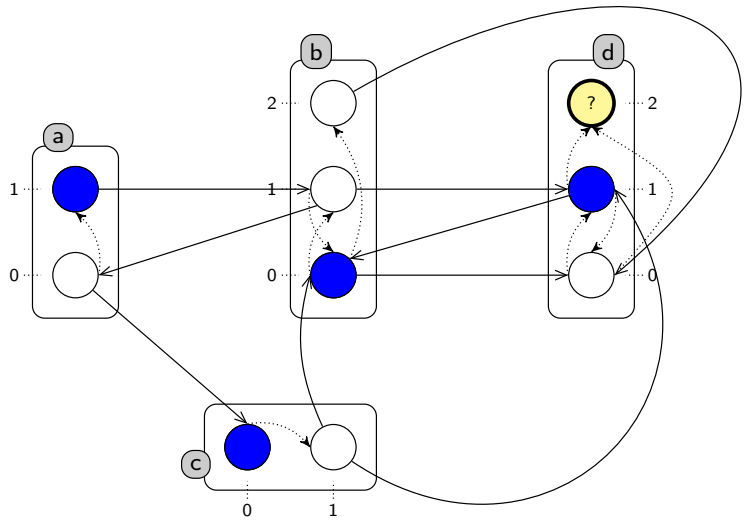
Necessary condition:

~~There exists a traversal~~ with no cycle

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



Over-approximation

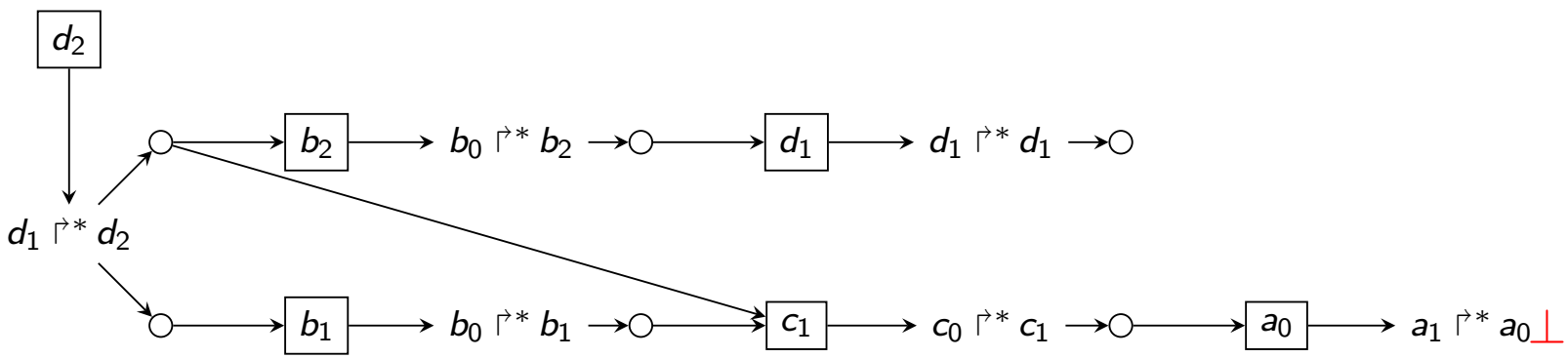


Necessary condition:

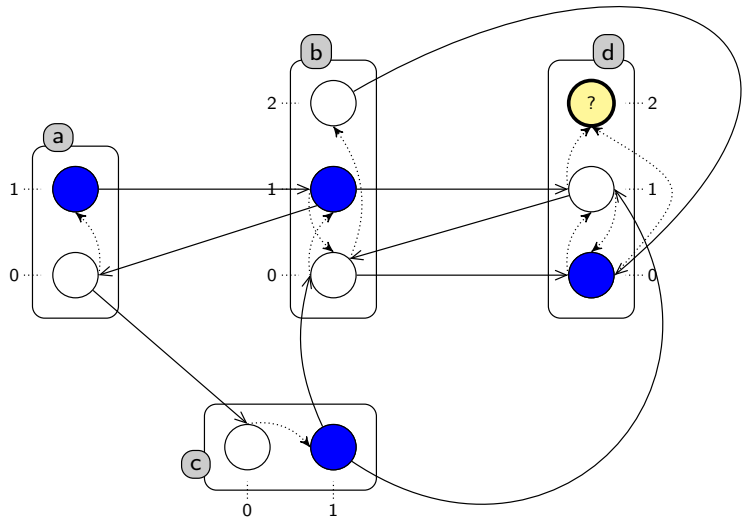
~~There exists a traversal~~ with no cycle

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

R is **false**



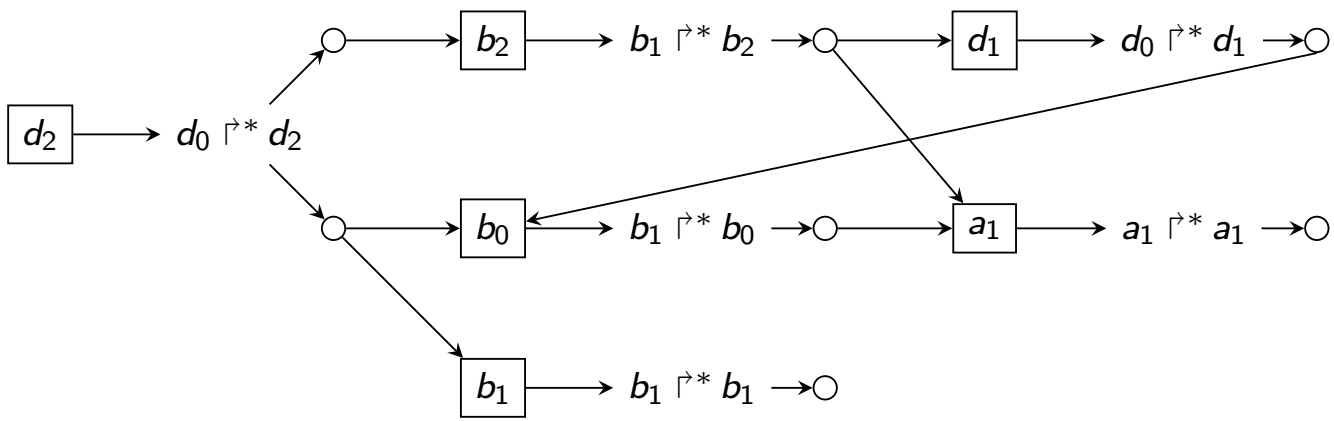
Over-approximation



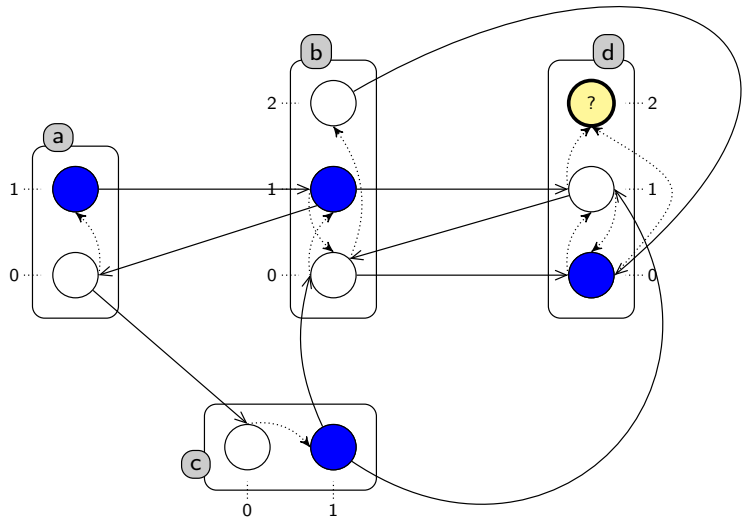
Necessary condition:

There exists a traversal with no cycle

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



Over-approximation

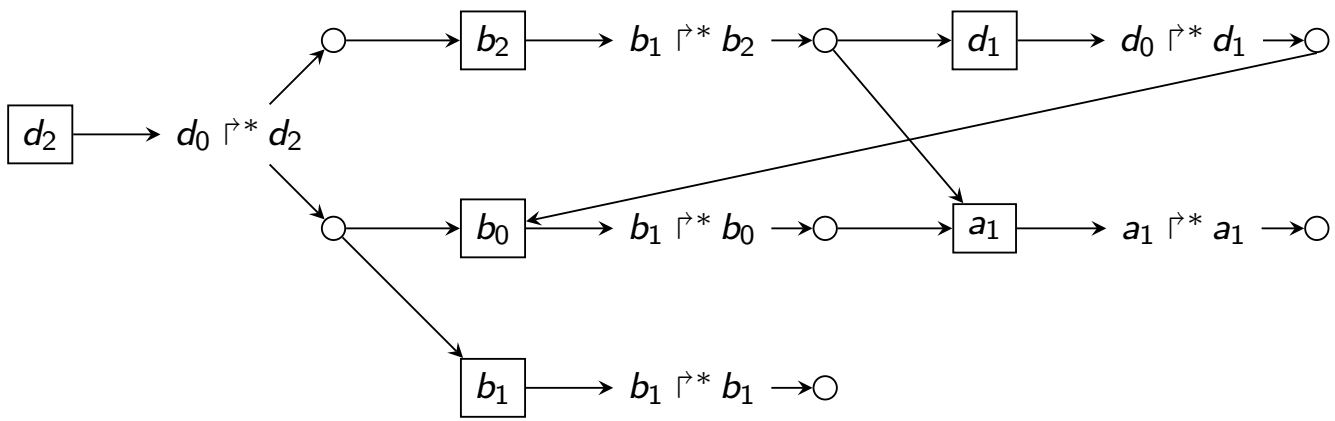


Necessary condition:

There exists a traversal with no cycle

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

Inconclusive

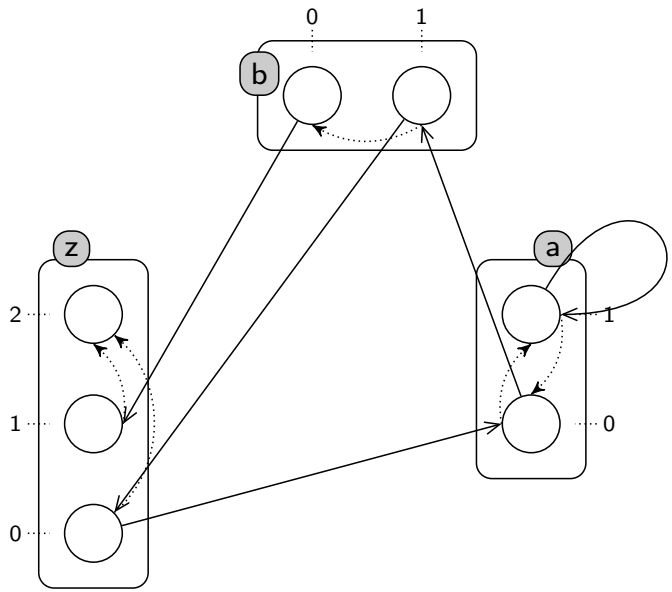


Static Analysis: Fixed Points

[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



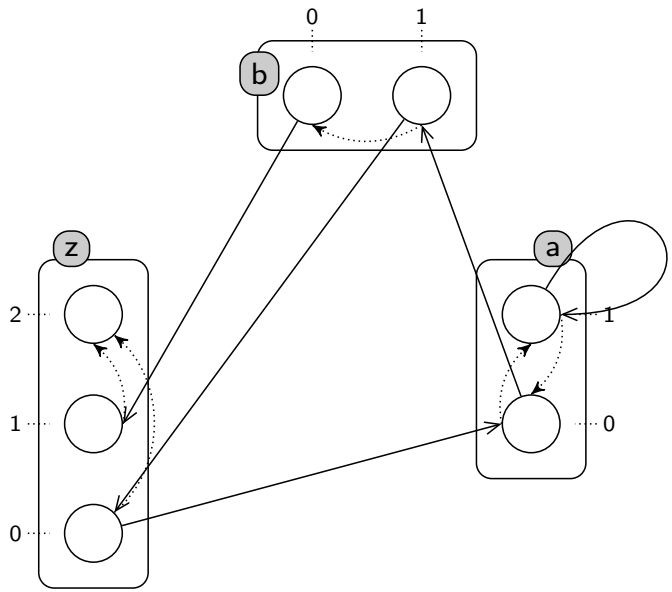
Static Analysis: Fixed Points

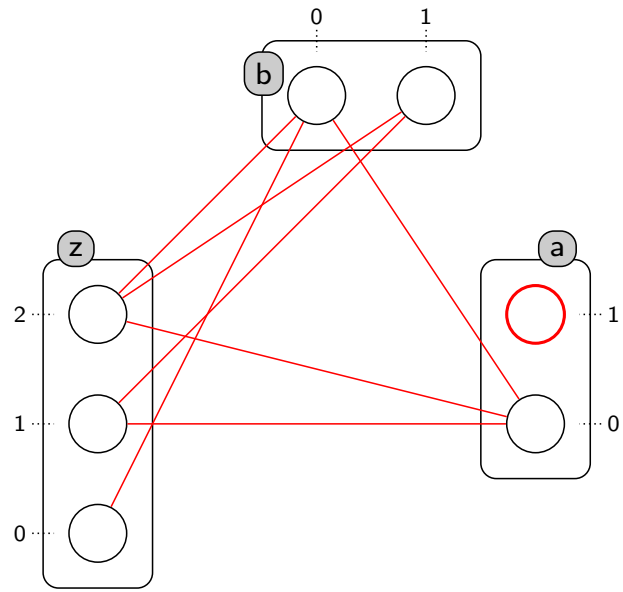
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

→ Hitless Graph





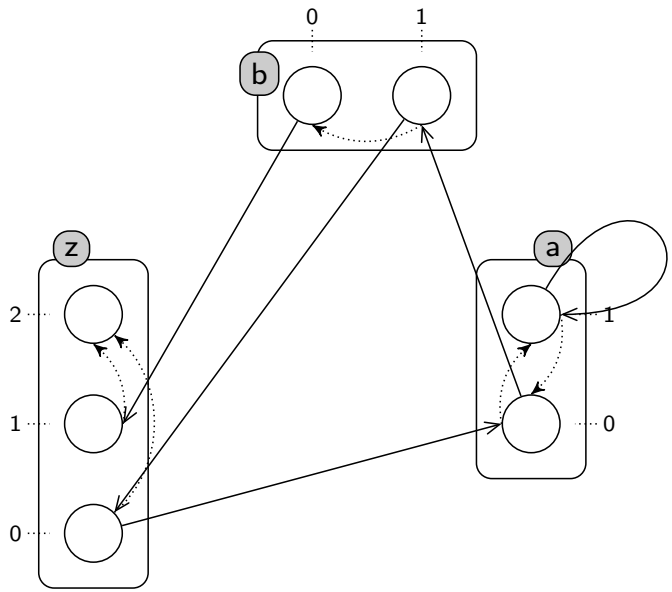
Static Analysis: Fixed Points

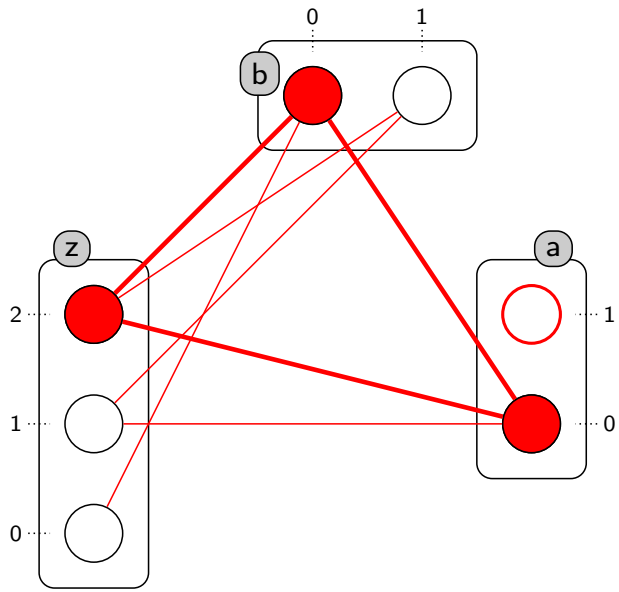
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

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





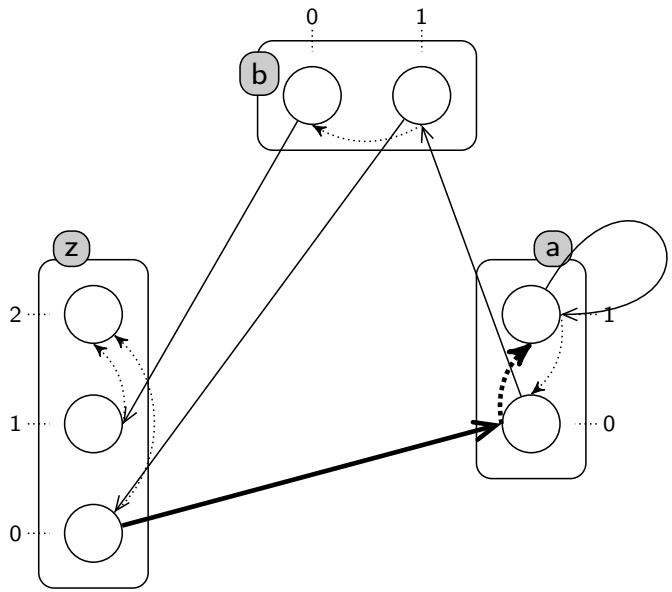
Static Analysis: Fixed Points

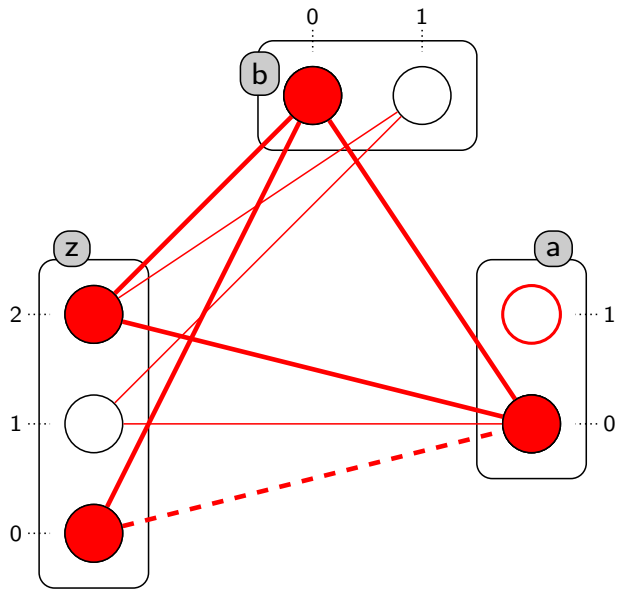
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

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





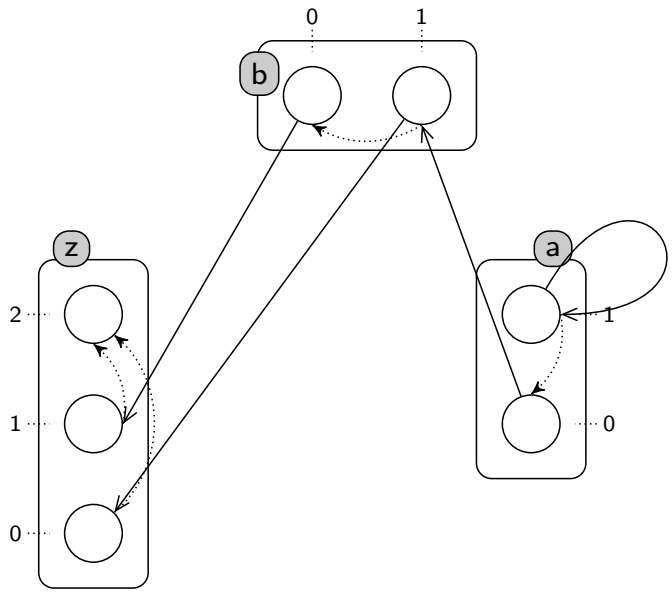
Static Analysis: Fixed Points

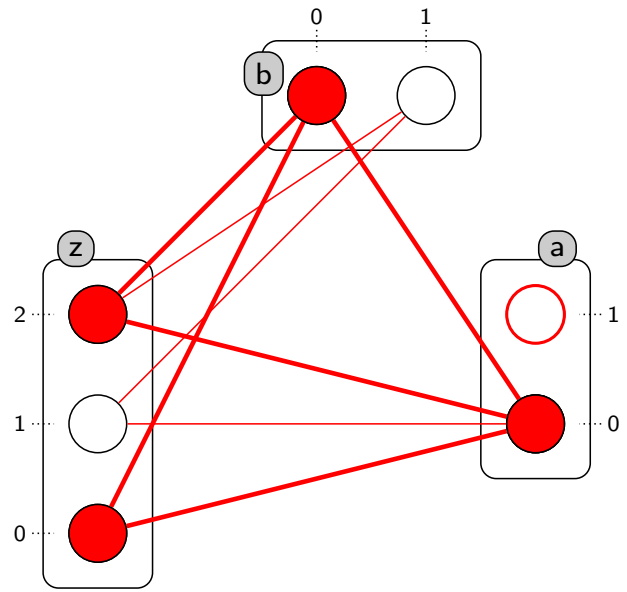
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

Fixed point = state where no action can be fired

→ avoid couples of processes bounded by an action

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





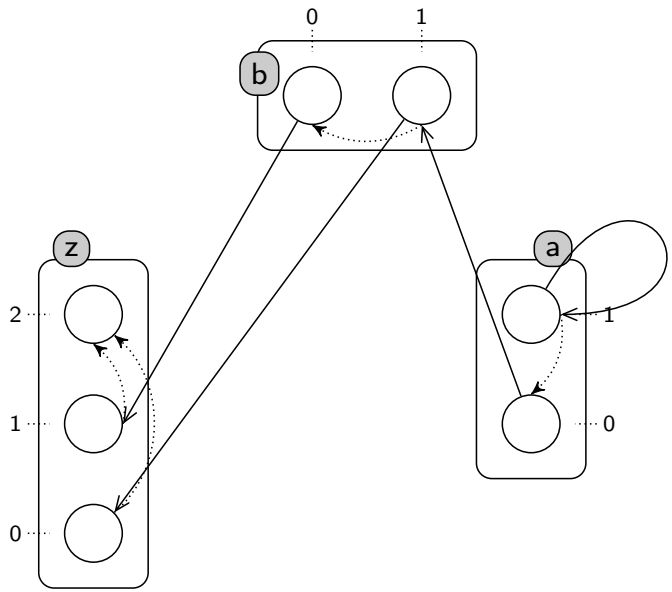
Static Analysis: Fixed Points

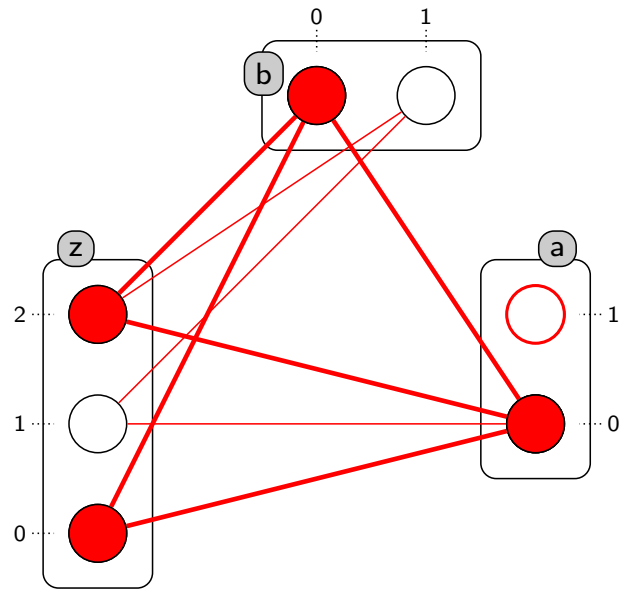
[Paulevé *et al.* in *Transactions on Computational Systems Biology*, 2011]

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