Réunion finale BioTempo

Introducing time parameters into Process Hitting with classes of priorities

Maxime FOLSCHETTE

MeForBio / IRCCyN / École Centrale de Nantes (Nantes, France)
maxime.folschette@irccyn.ec-nantes.fr
http://www.irccyn.ec-nantes.fr/~folschet/

Joint work with:
Loïc PAULEVÉ, Morgan MAGNIN, Olivier ROUX

Context and Aims

MeForBio team: Algebraic modeling to study large dynamical biological systems

→ Contribution: the **Process Hitting** framework
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]
[Paulevé *et al.* in Mathematical Structures in Computer Science, 2012]

- A restriction of synchronous automata networks
- Special form for the actions ⇒ more atomistic than Interaction Graphs
- Efficient reachability analysis

→ Introduction of temporal features:

1) Stochastic parameters
   [Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]
2) Priorities
   [Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]
3) Neutralizing edges

The Process Hitting modeling

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

Sorts: components a, b, z

The Process Hitting modeling

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

Sorts: components a, b, z

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

The Process Hitting modeling

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

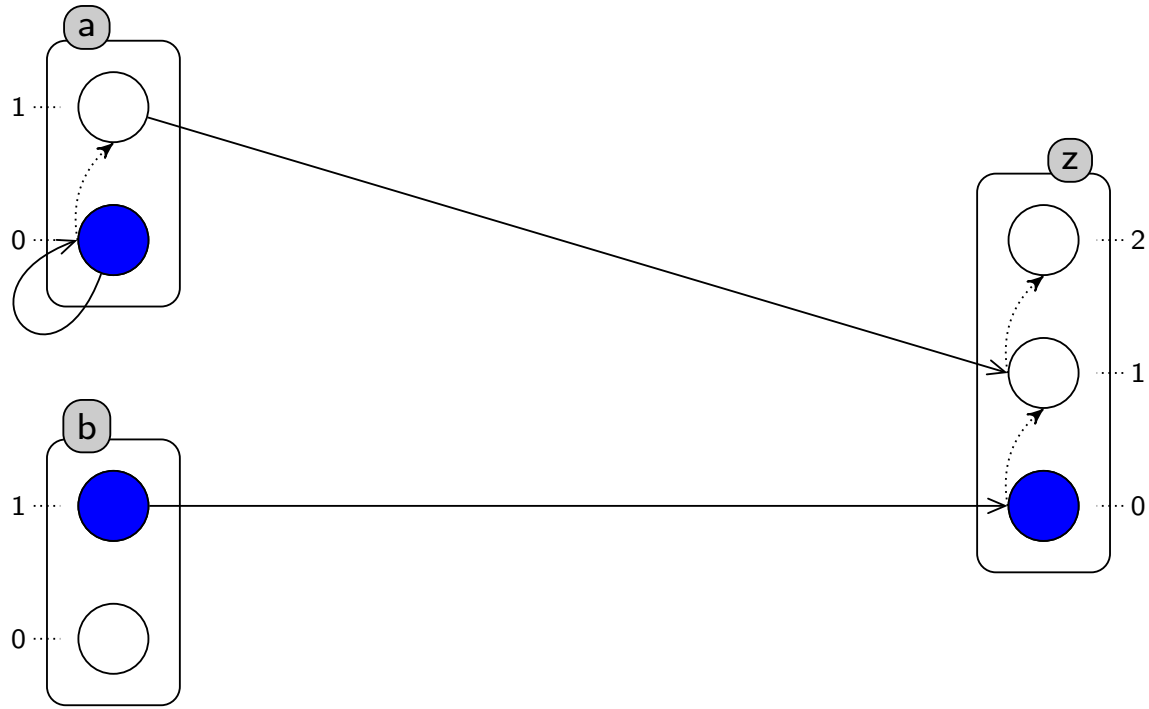
Sorts: components a, b, z

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

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

The Process Hitting modeling

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



Sorts: components a, b, z

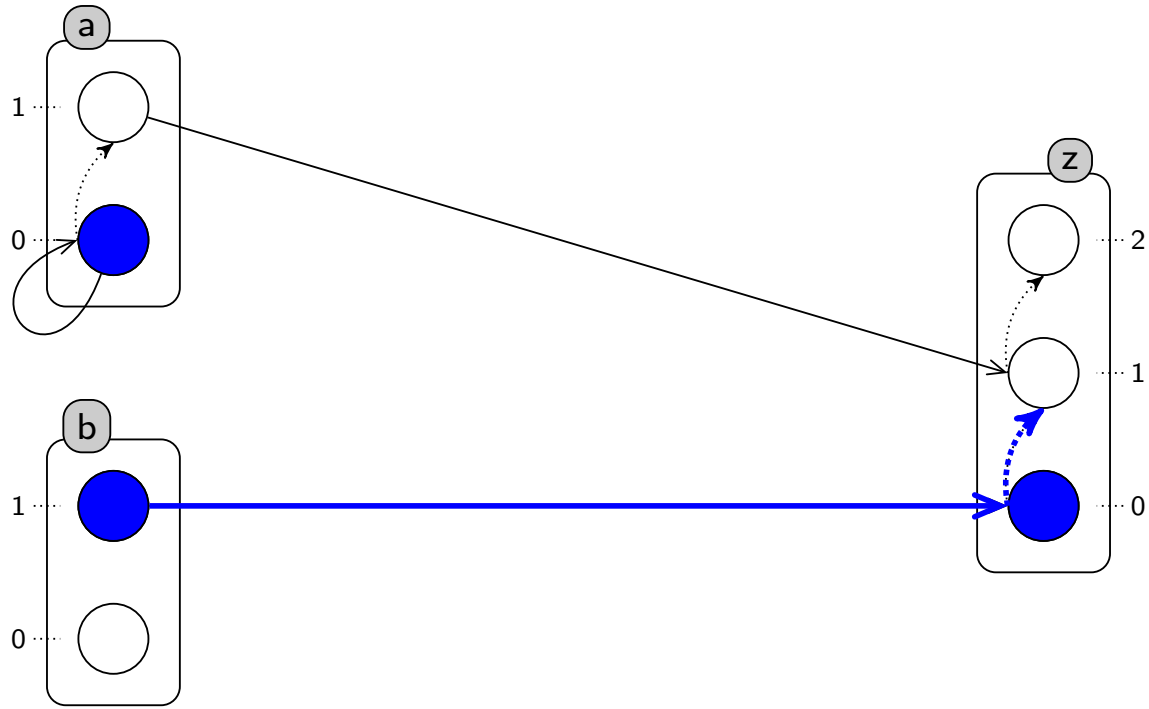
Processes: local states / levels of expression z_0, z_1, z_2

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

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

The Process Hitting modeling

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



Sorts: components a, b, z

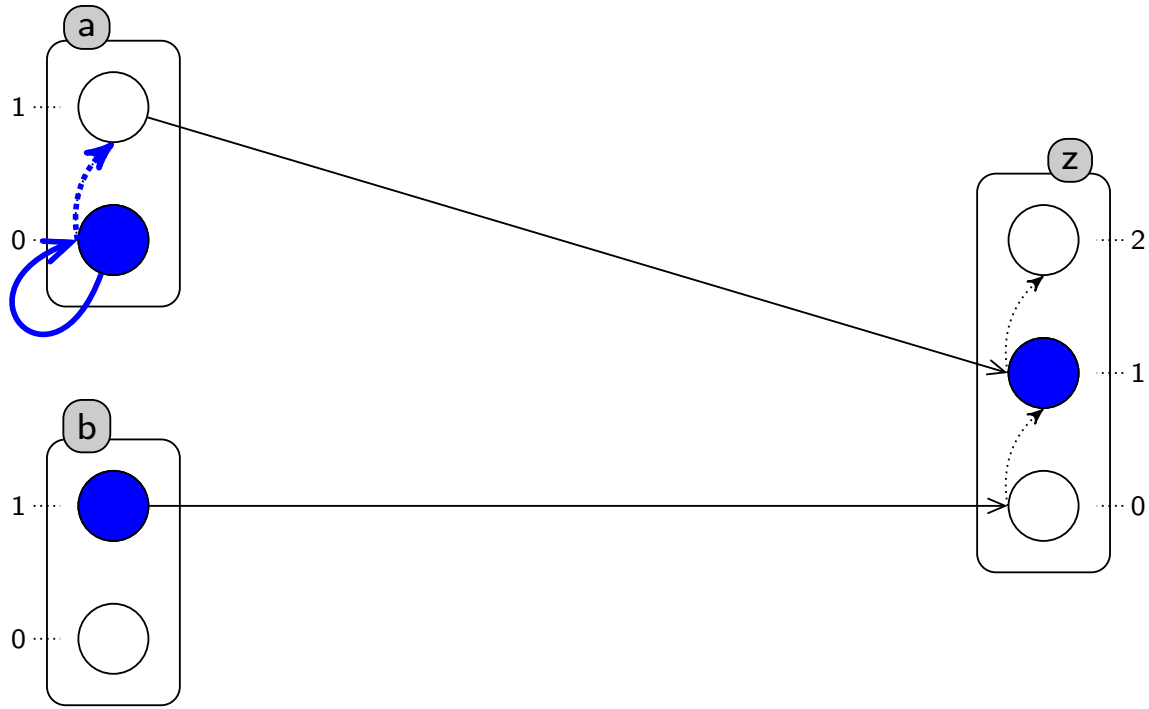
Processes: local states / levels of expression z_0, z_1, z_2

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

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

The Process Hitting modeling

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



Sorts: components a, b, z

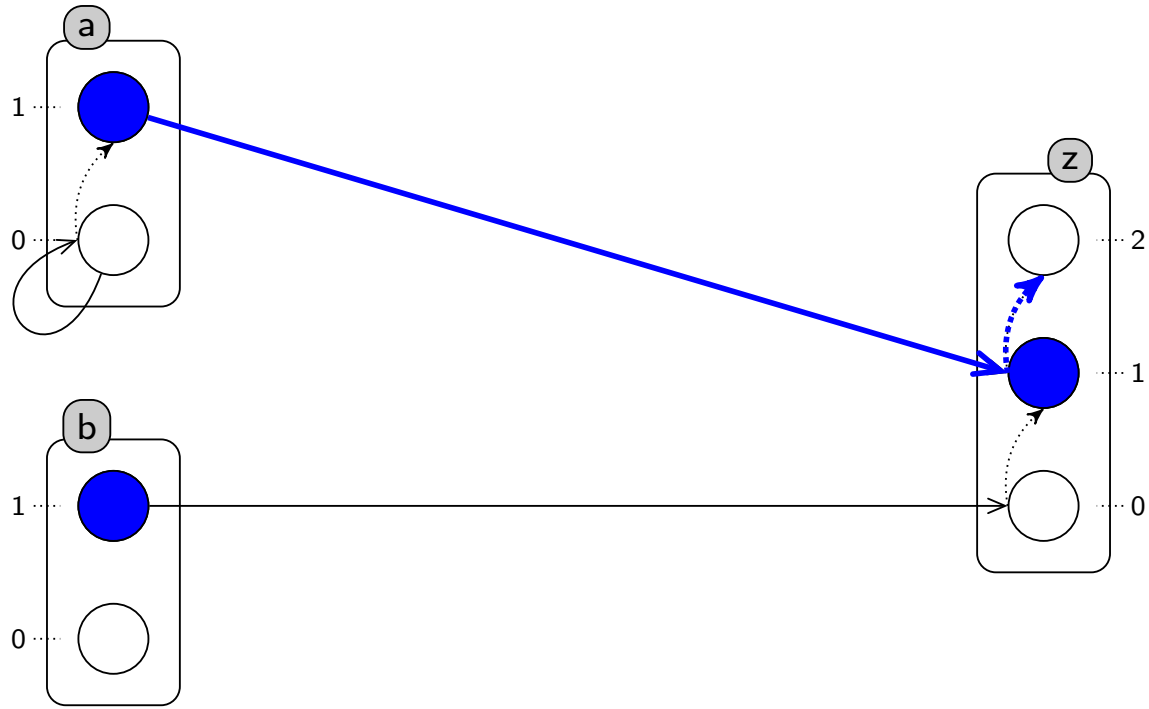
Processes: local states / levels of expression z_0, z_1, z_2

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

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

The Process Hitting modeling

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



Sorts: components a, b, z

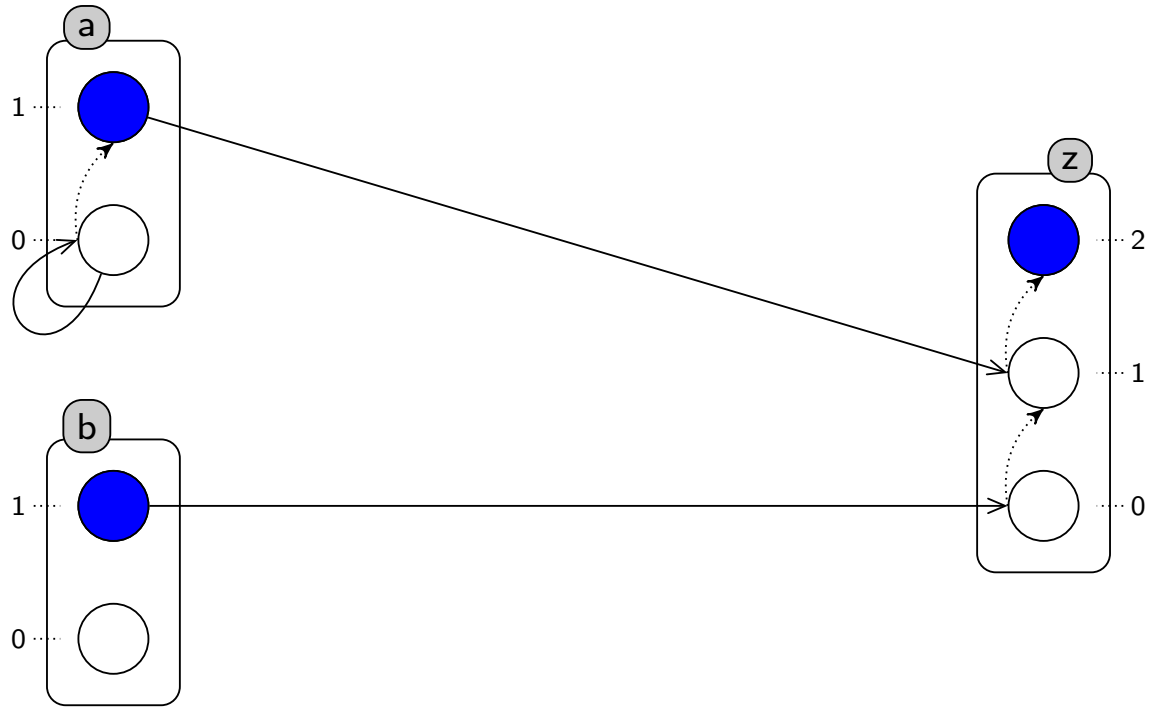
Processes: local states / levels of expression z_0, z_1, z_2

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

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

The Process Hitting modeling

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



Sorts: components a, b, z

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

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

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

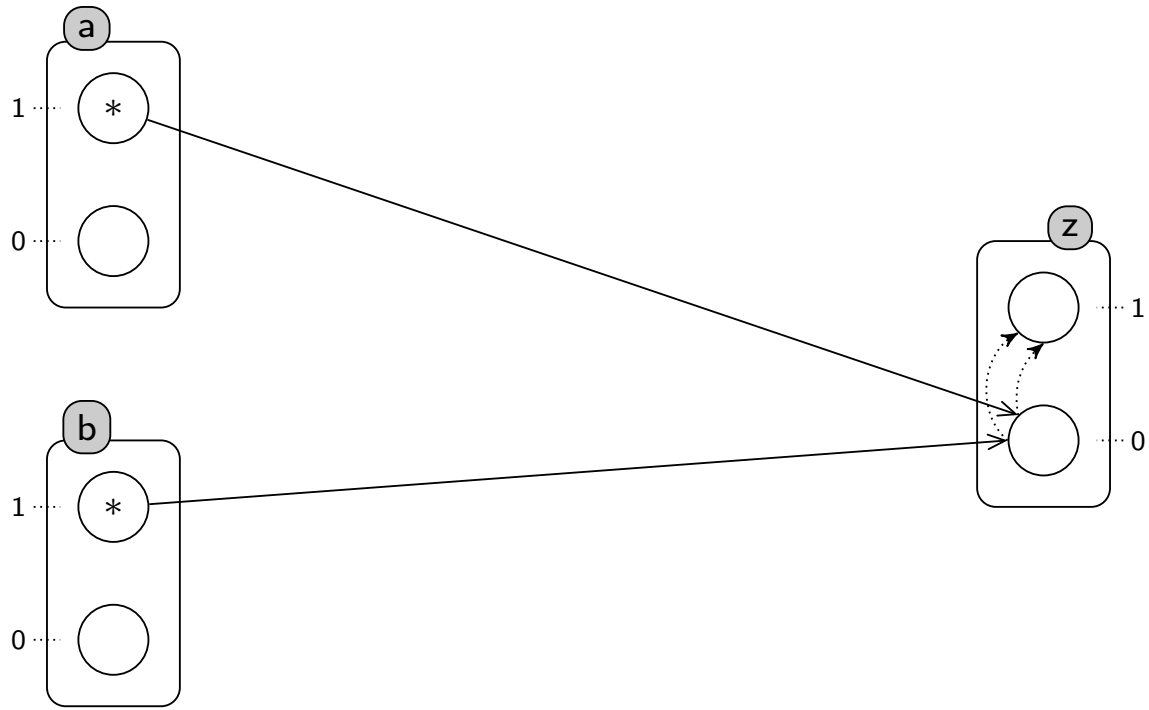
Adding cooperations

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

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

Adding cooperations

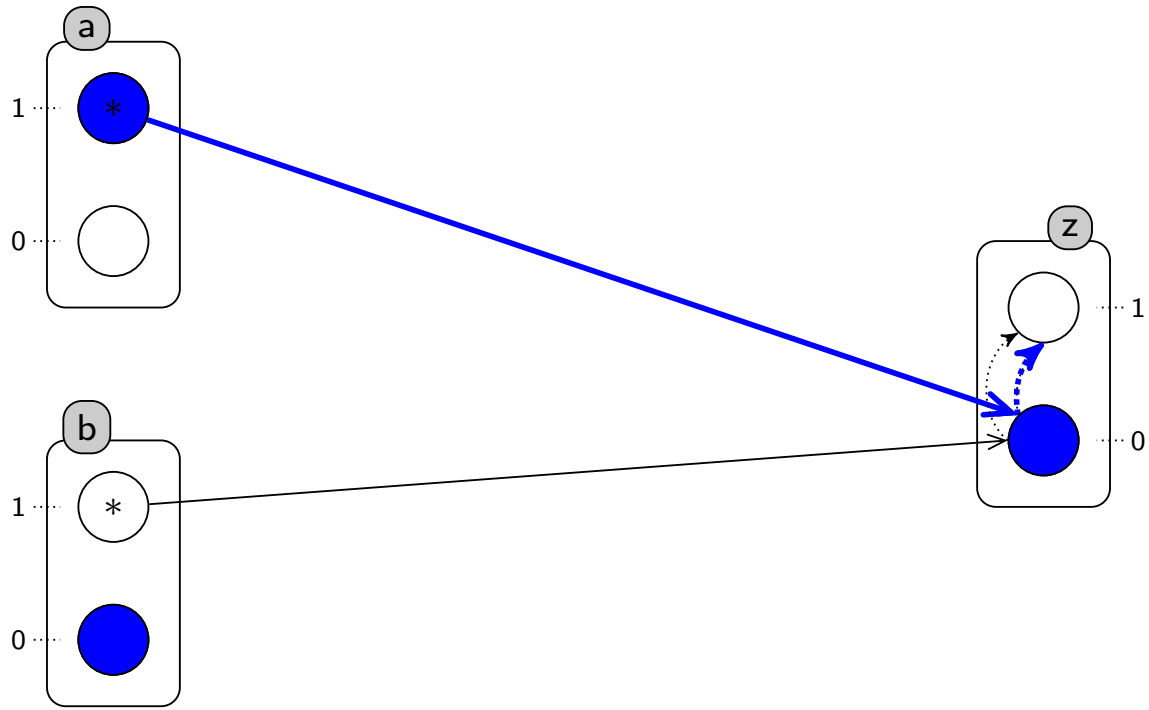
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Adding cooperations

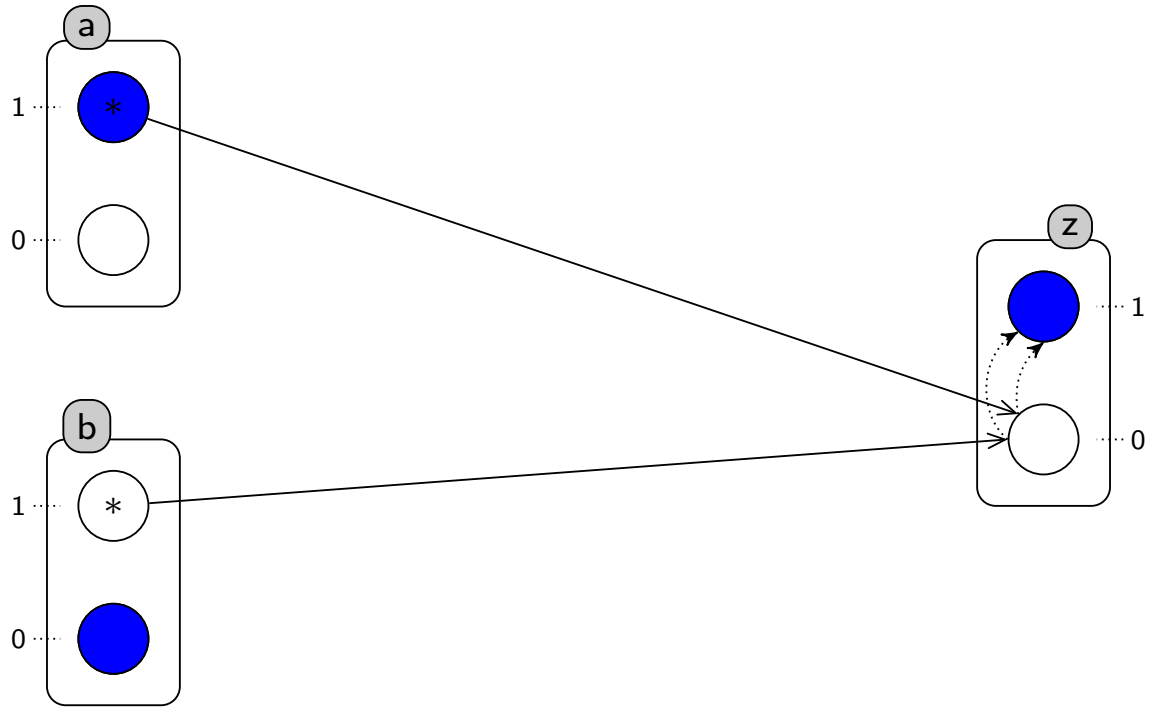
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Adding cooperations

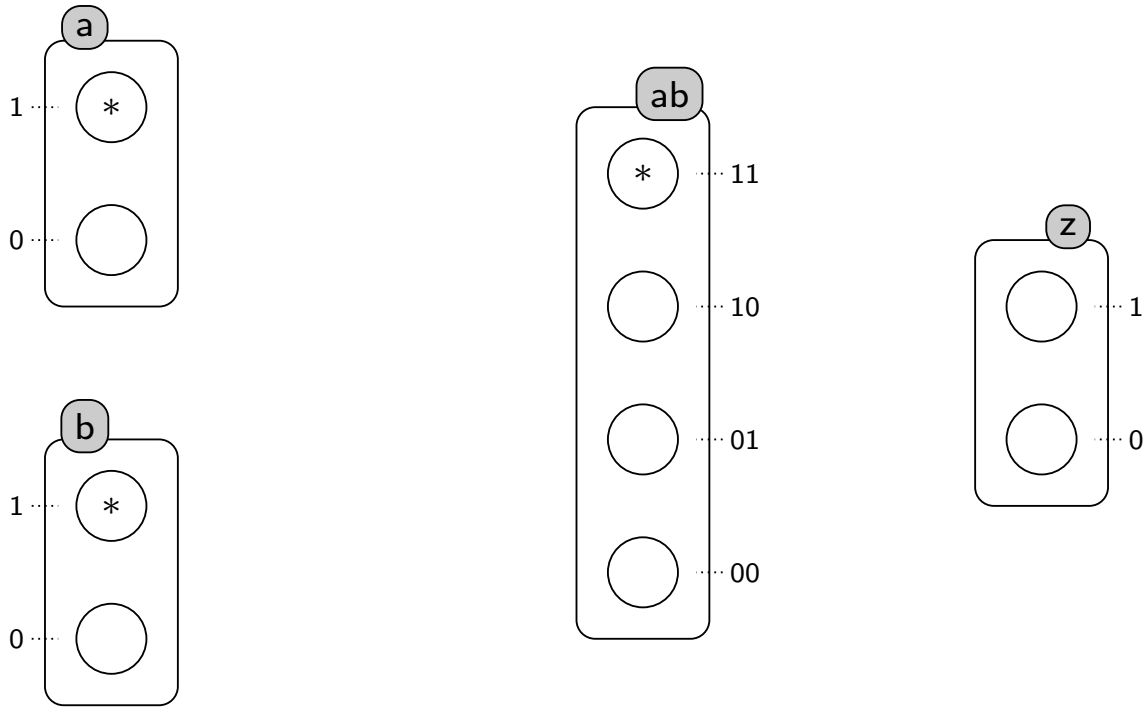
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Adding cooperations

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

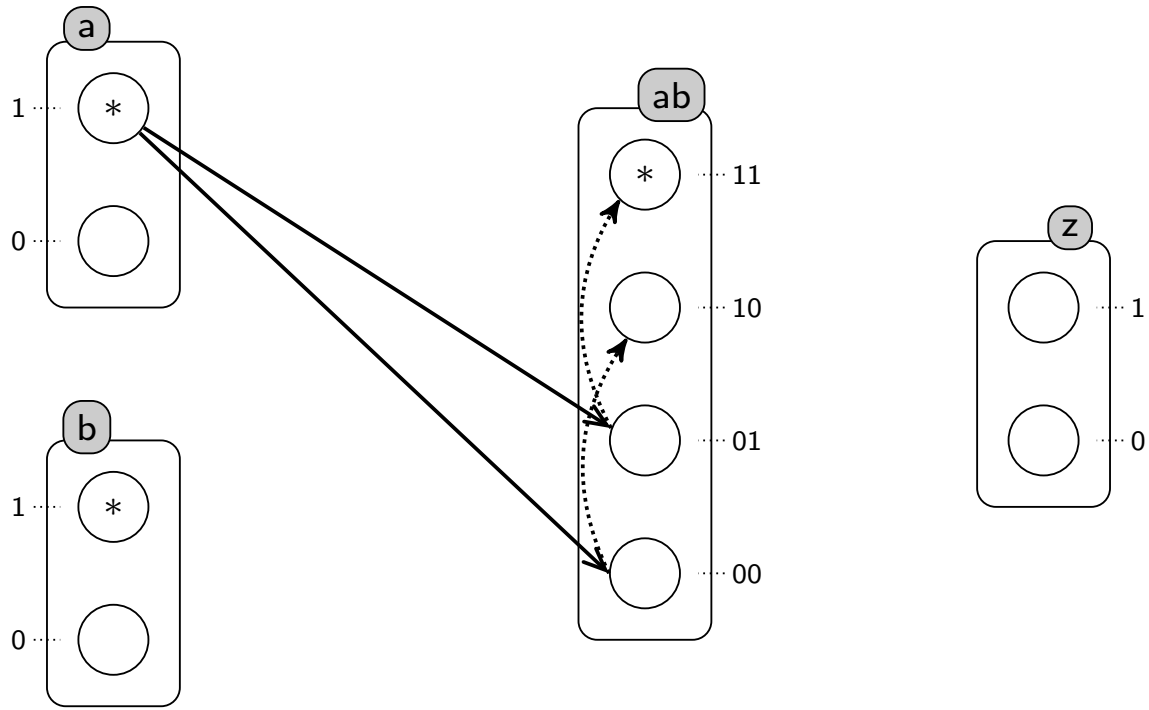


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

Adding cooperations

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

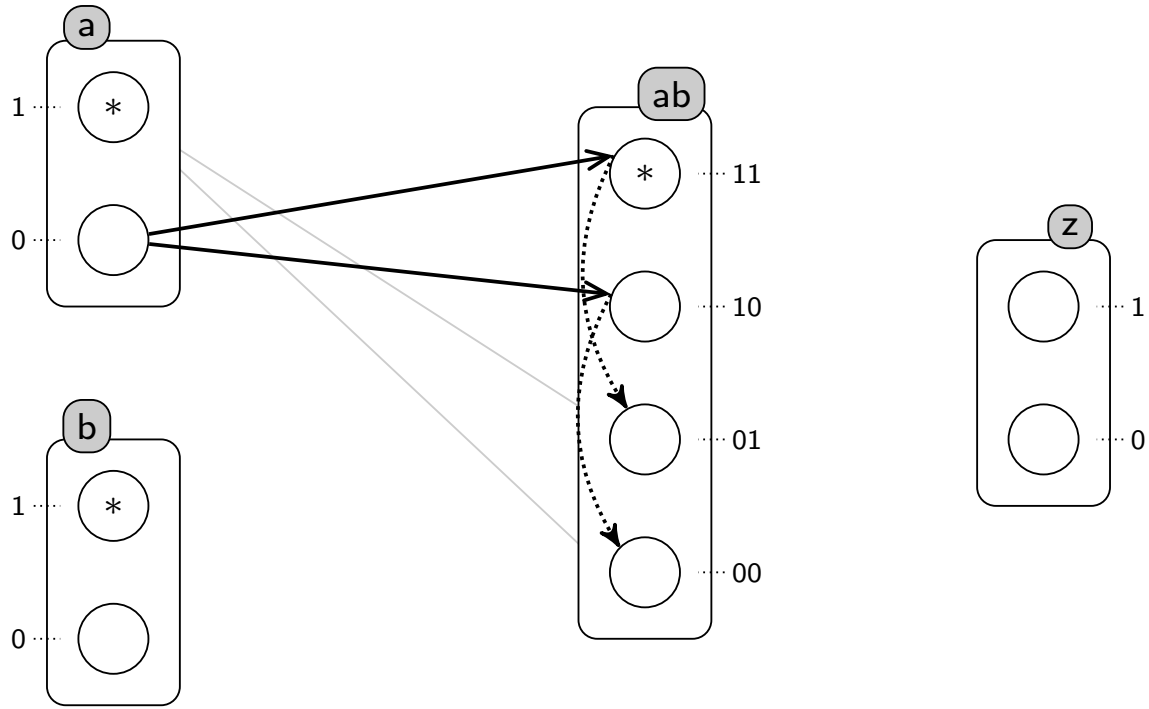


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

Adding cooperations

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

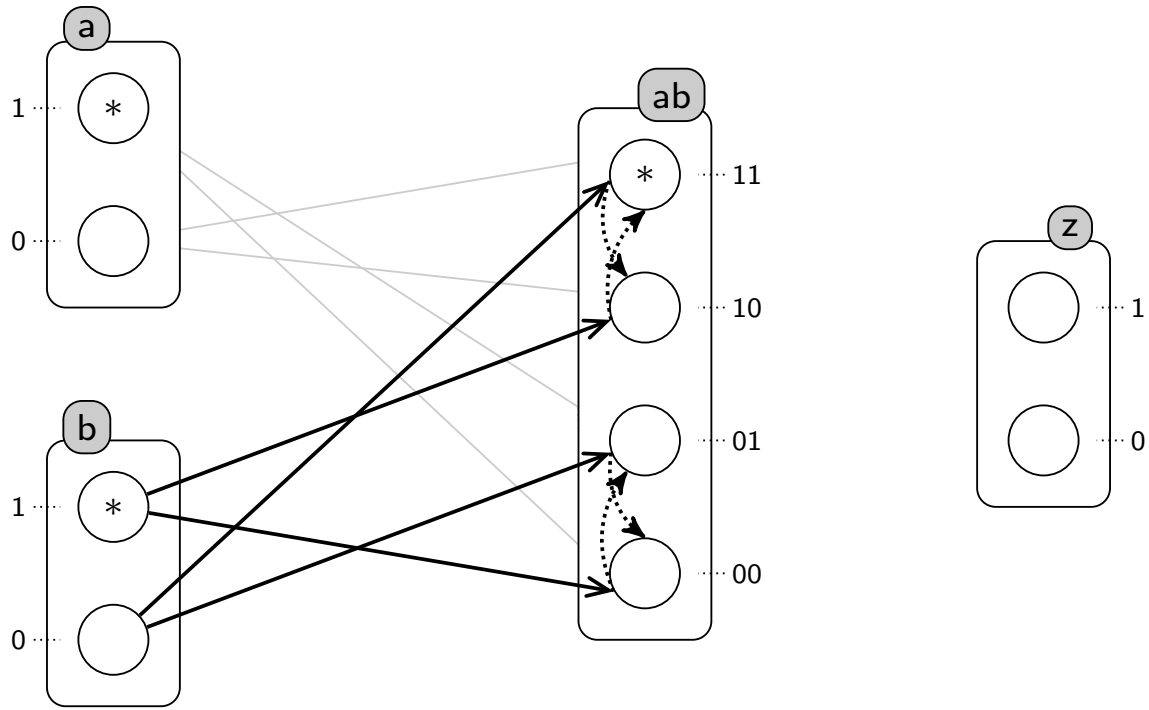


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

Adding cooperations

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

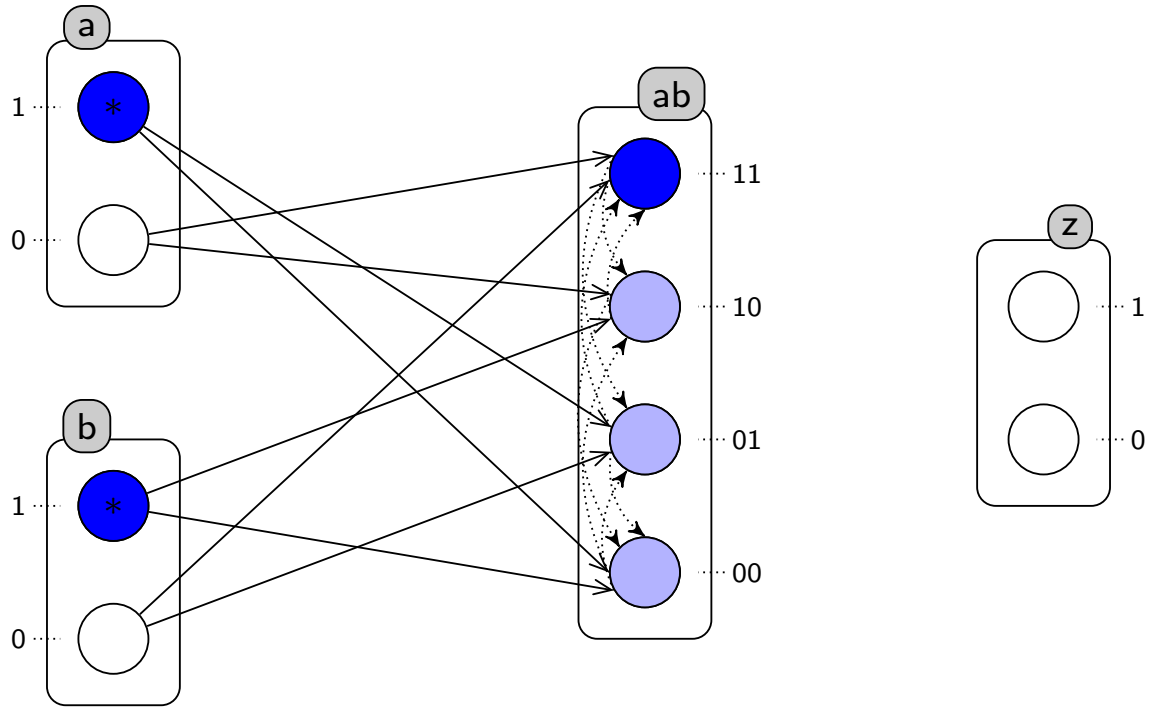


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

Adding cooperations

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



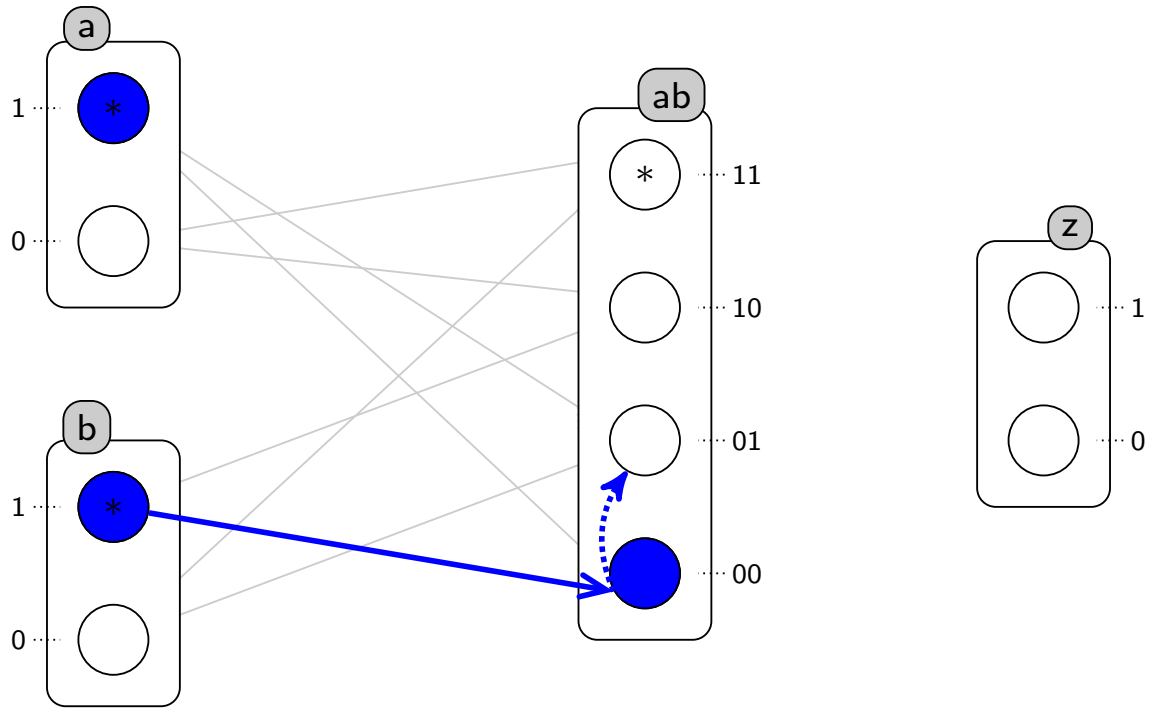
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

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

Adding cooperations

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



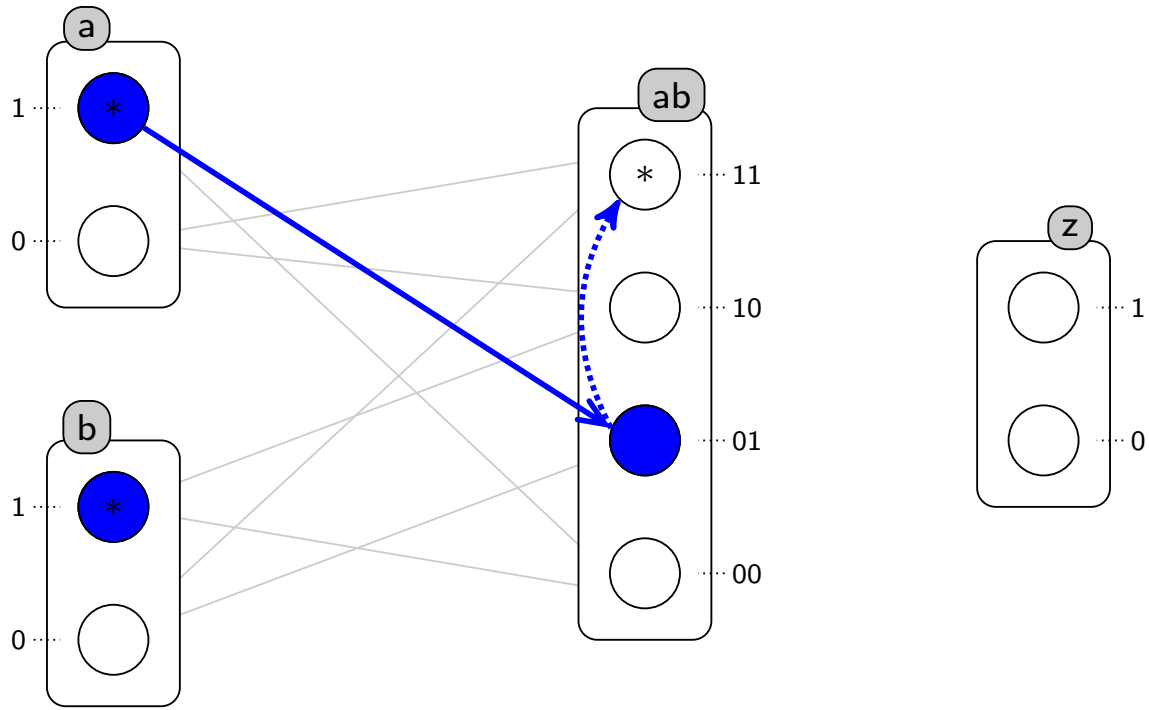
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

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

Adding cooperations

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



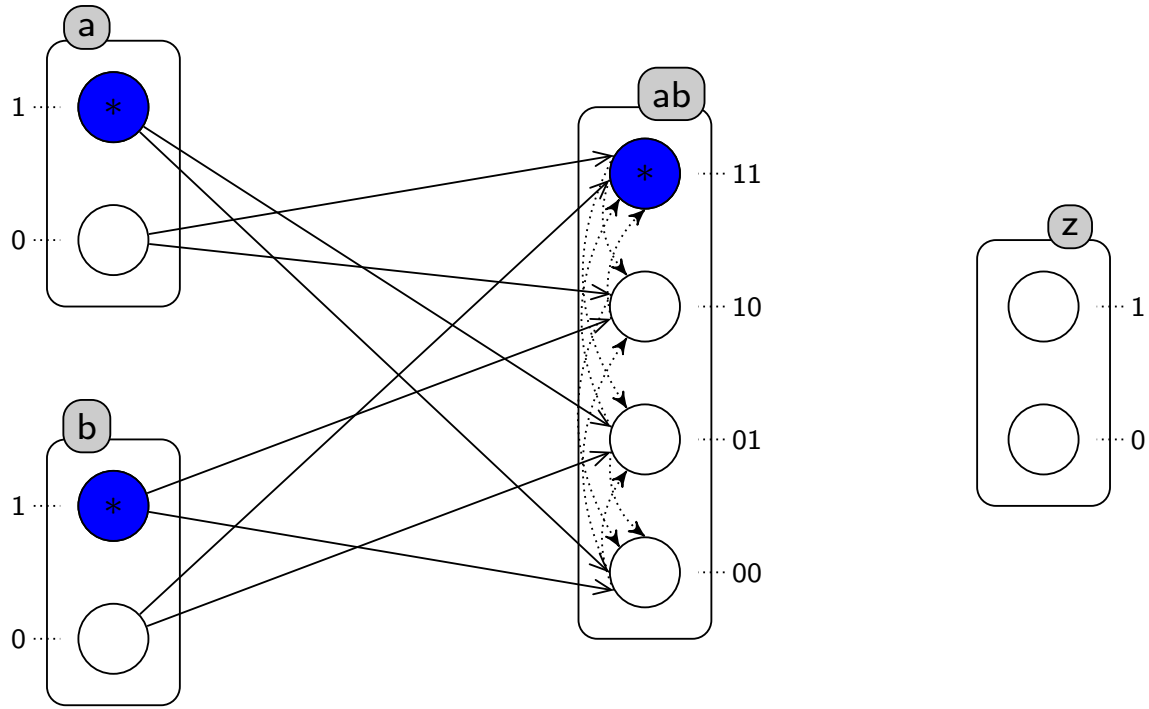
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

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

Adding cooperations

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



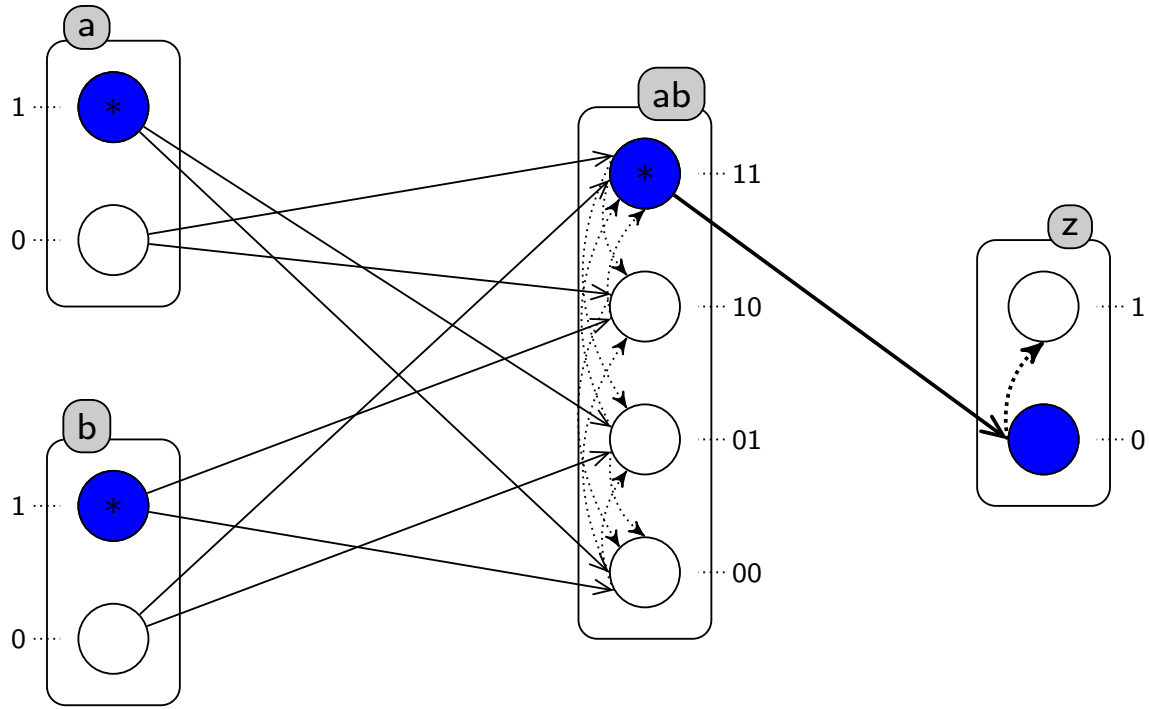
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

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

Adding cooperations

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



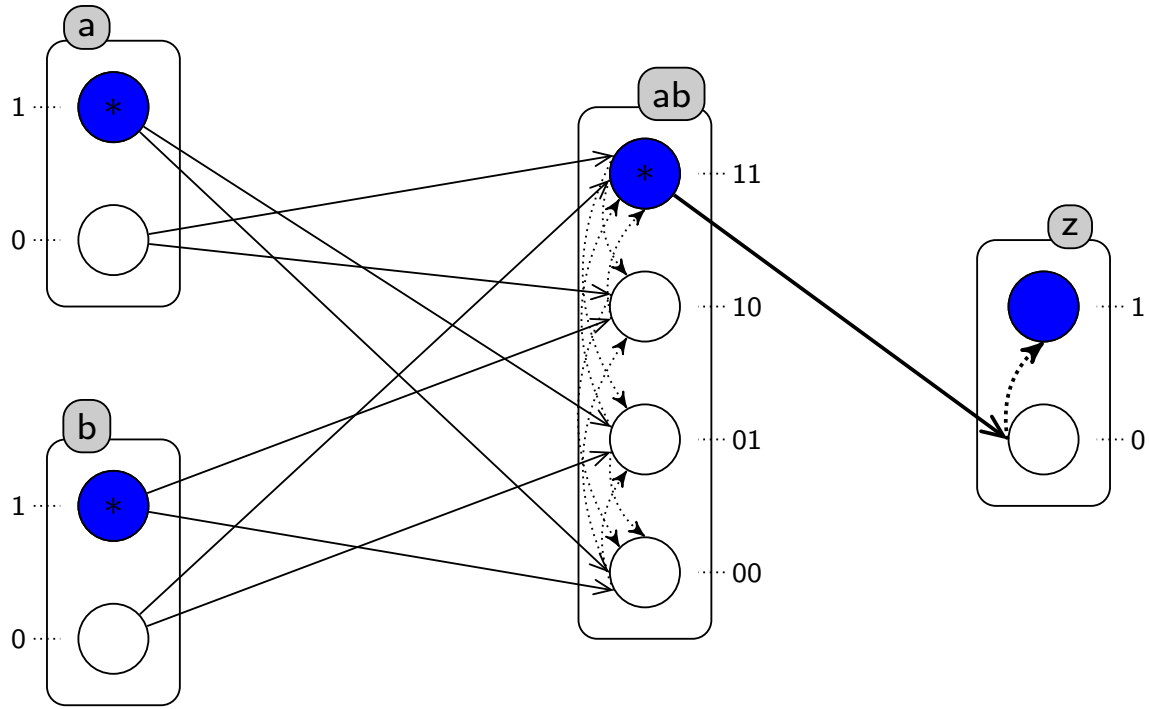
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \nearrow z_1$

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

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

Adding cooperations

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



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

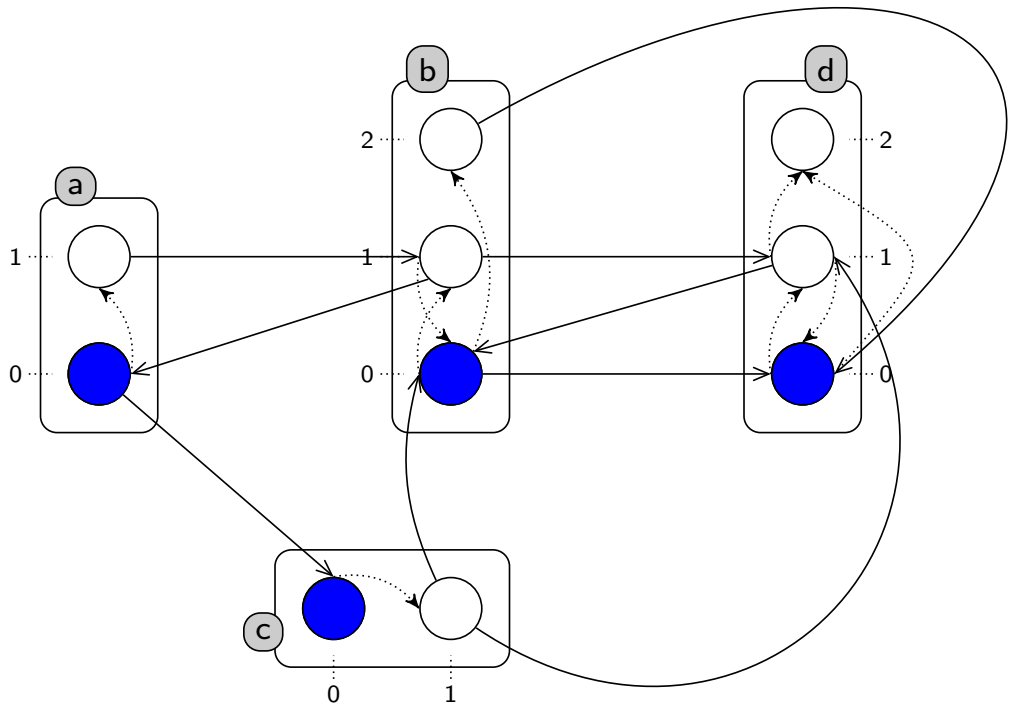
Solution: a **cooperative sort** ab to express $\underline{a_1 \wedge b_1}$

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

Static analysis: successive reachability

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

Successive reachability of processes:



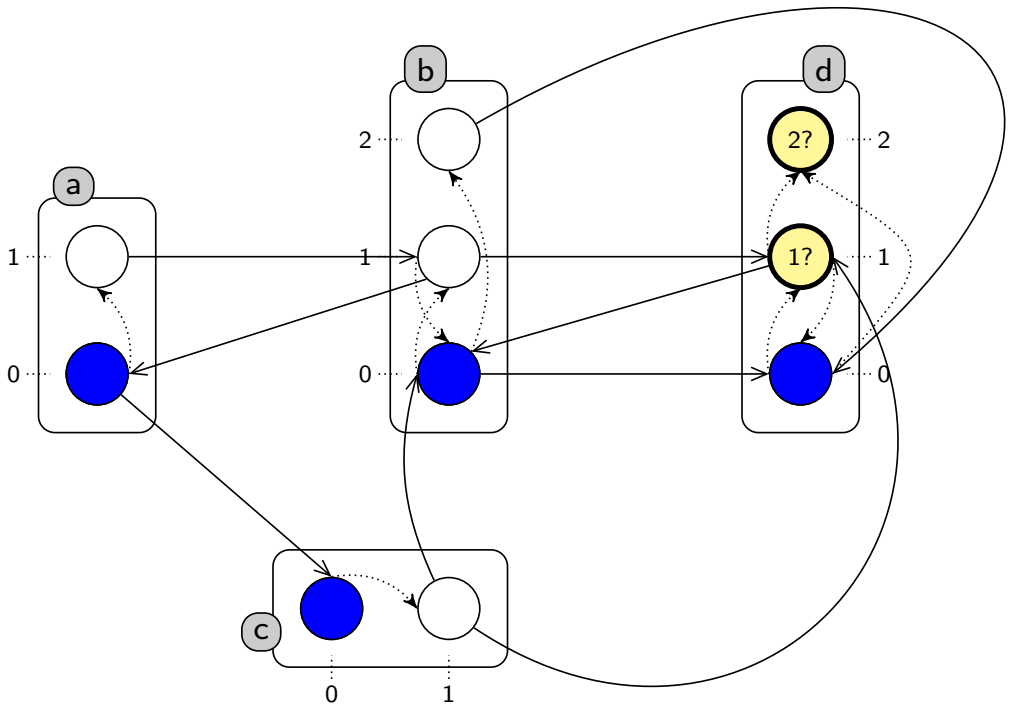
- Initial state

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

Static analysis: successive reachability

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

Successive reachability of processes:



- Initial state

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

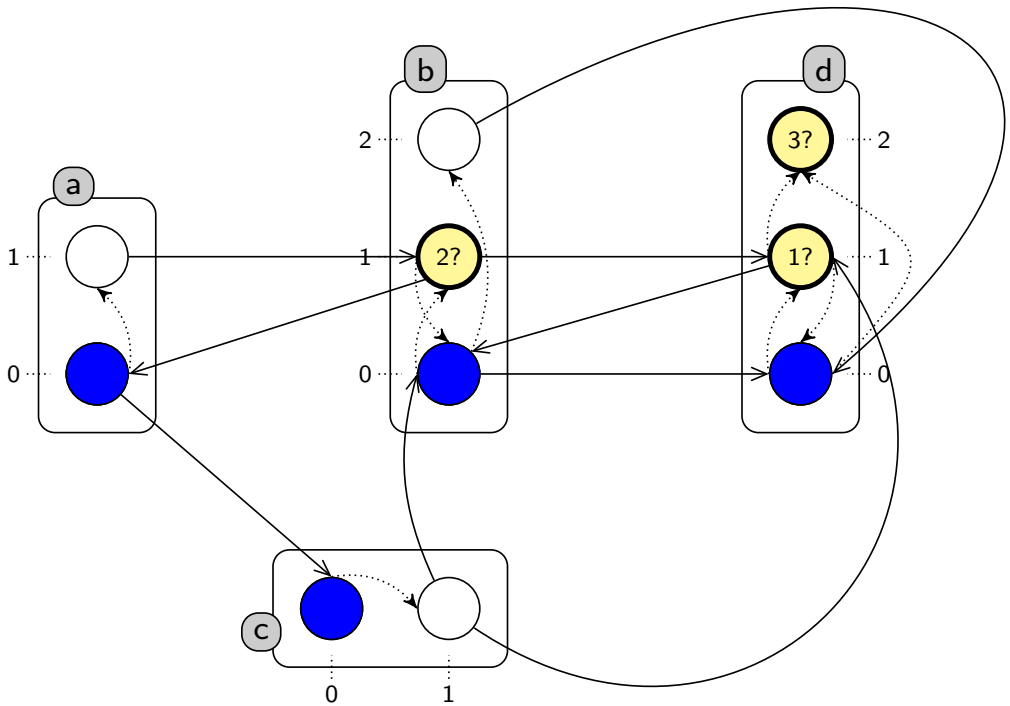
- Objectives

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

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

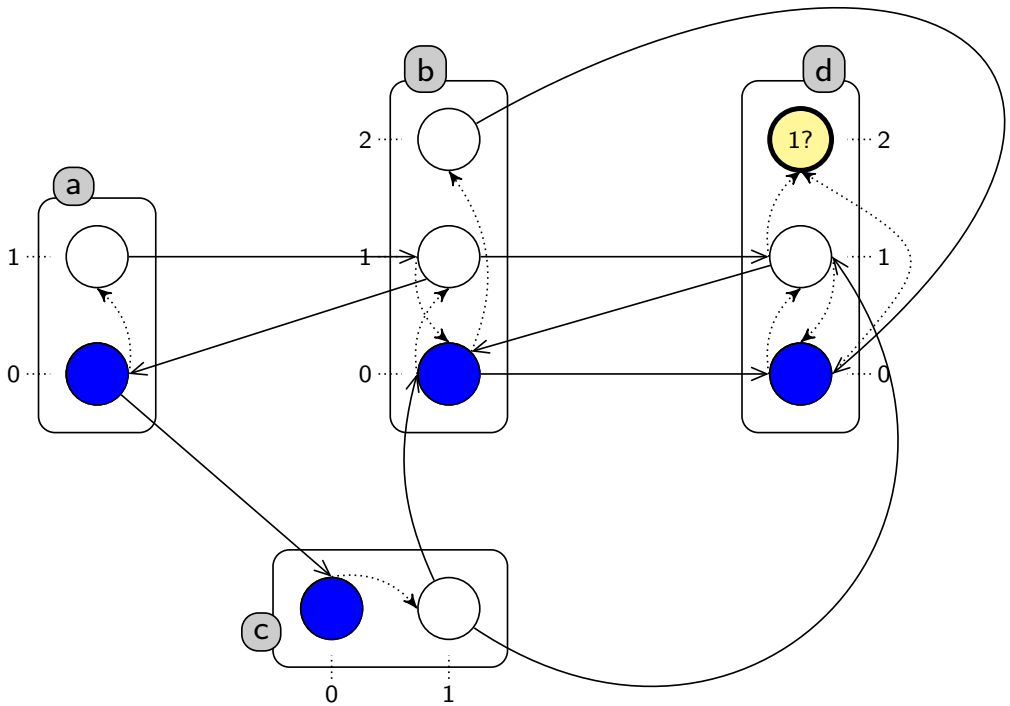
$[\ \Rsh\ d_1\ ::\ \Rsh\ d_2\ ]$

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

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

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

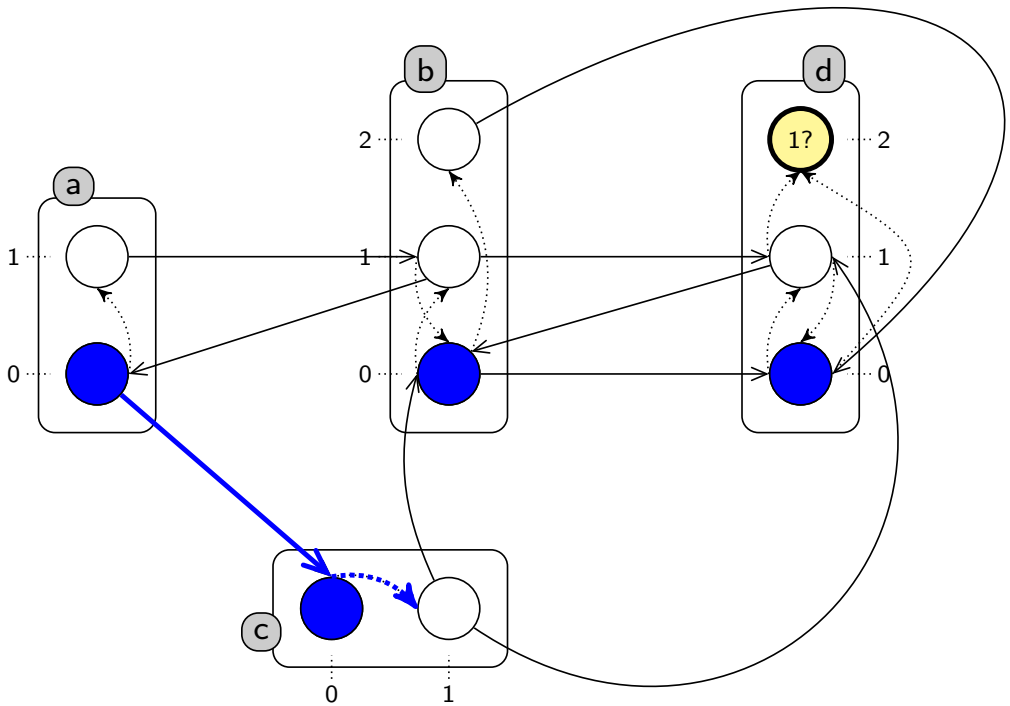
$$[\ \Rsh\ d_1 :: \Rsh\ b_1 :: \Rsh\ d_2\ ]$$

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

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

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

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

$[\ \Rsh\ d_2\ ]$

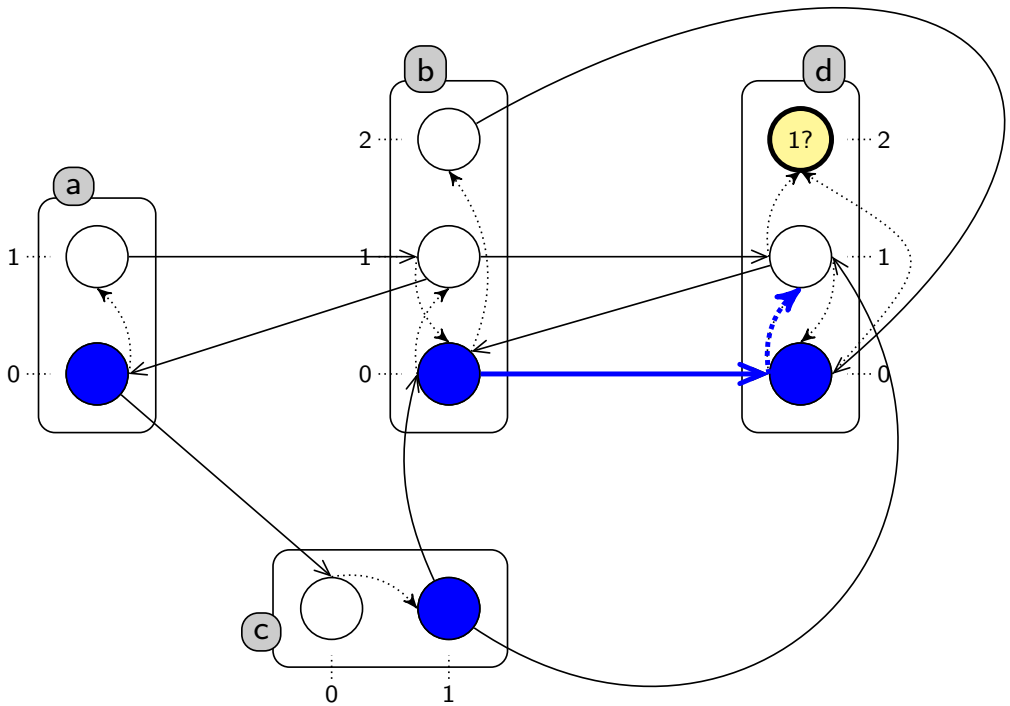
$\rightarrow$ Concretization of the objective = scenario

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

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

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

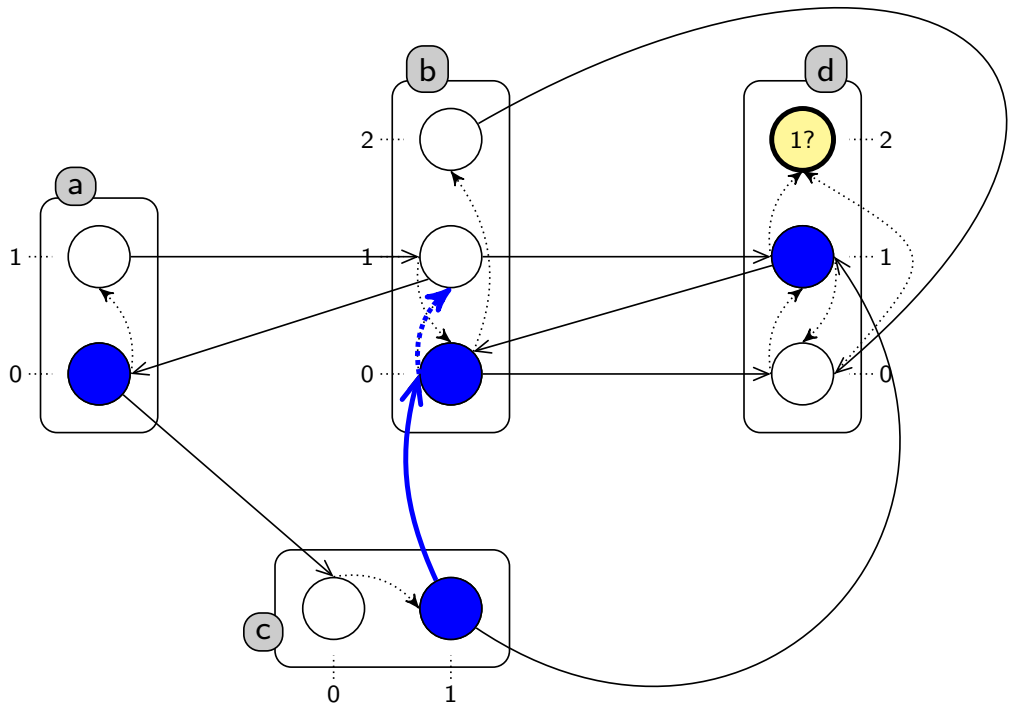
$\rightarrow$ Concretization of the objective = scenario

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

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

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

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

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

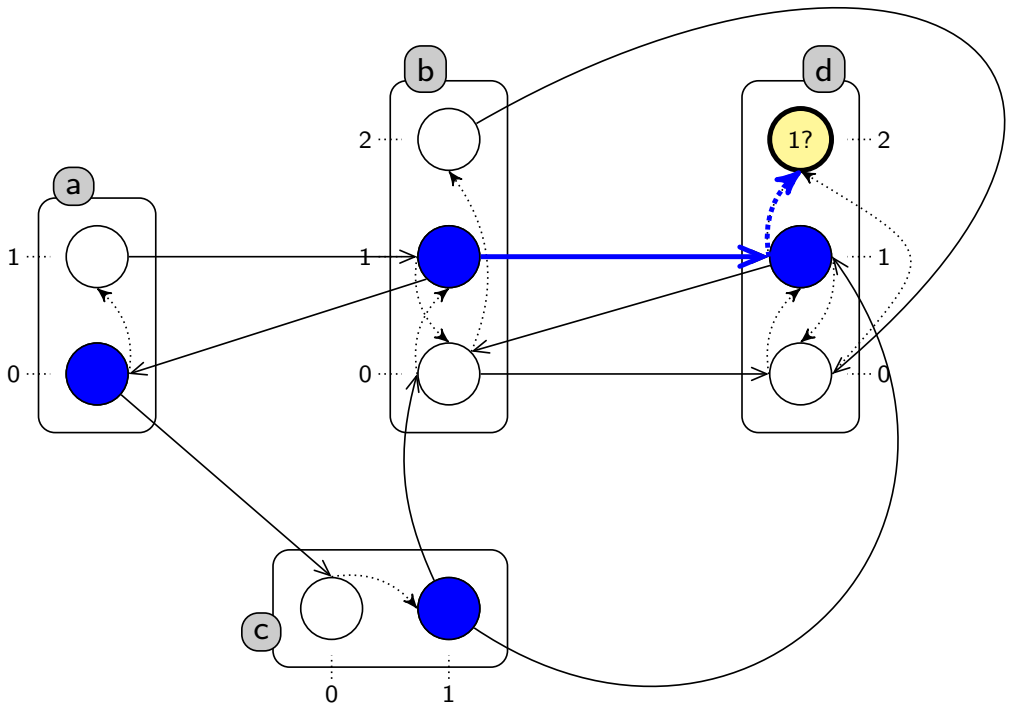
$\rightarrow$ Concretization of the objective = scenario

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

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

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

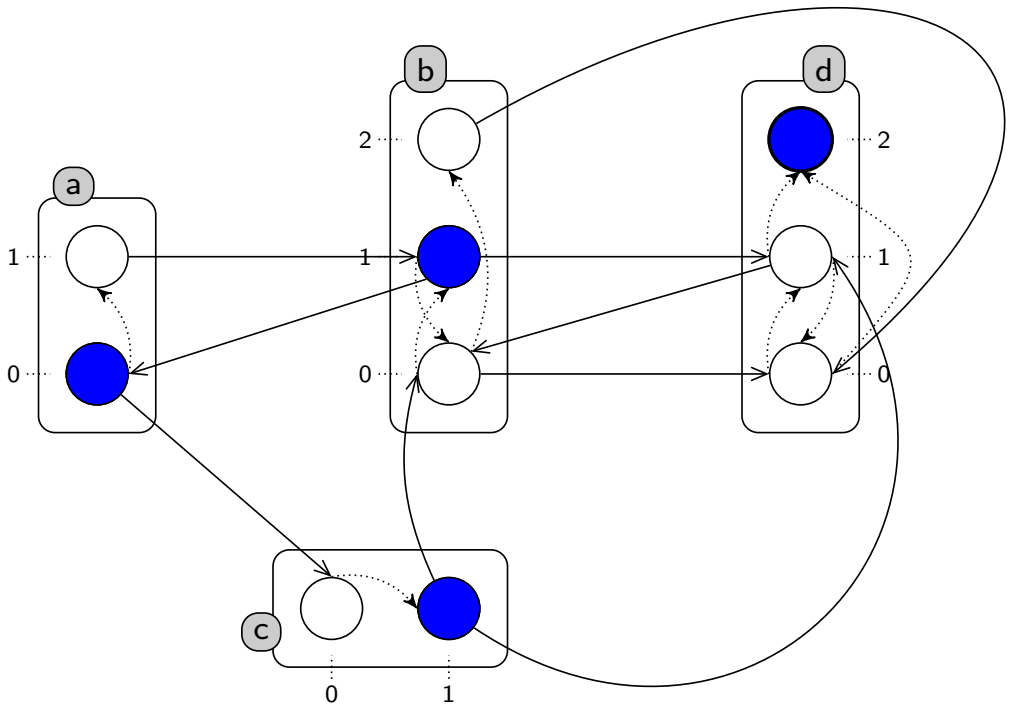
$\rightarrow$ Concretization of the objective = scenario

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

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

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

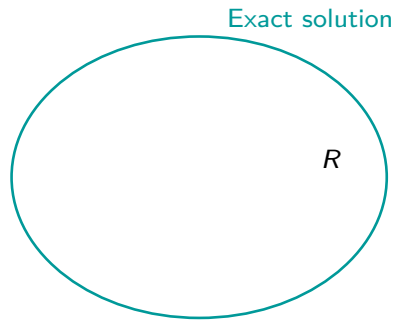
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



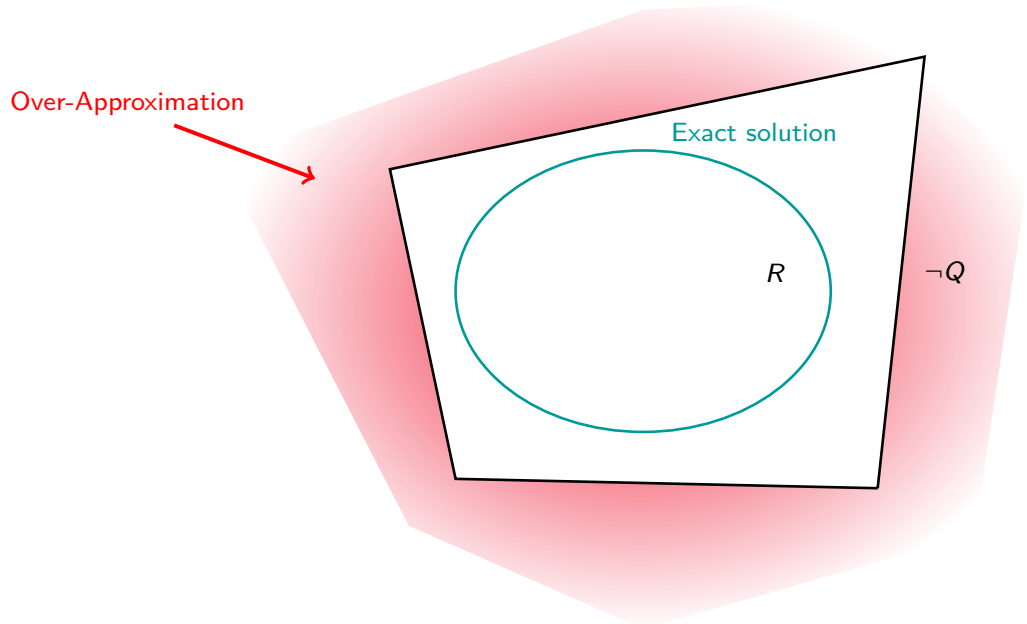
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



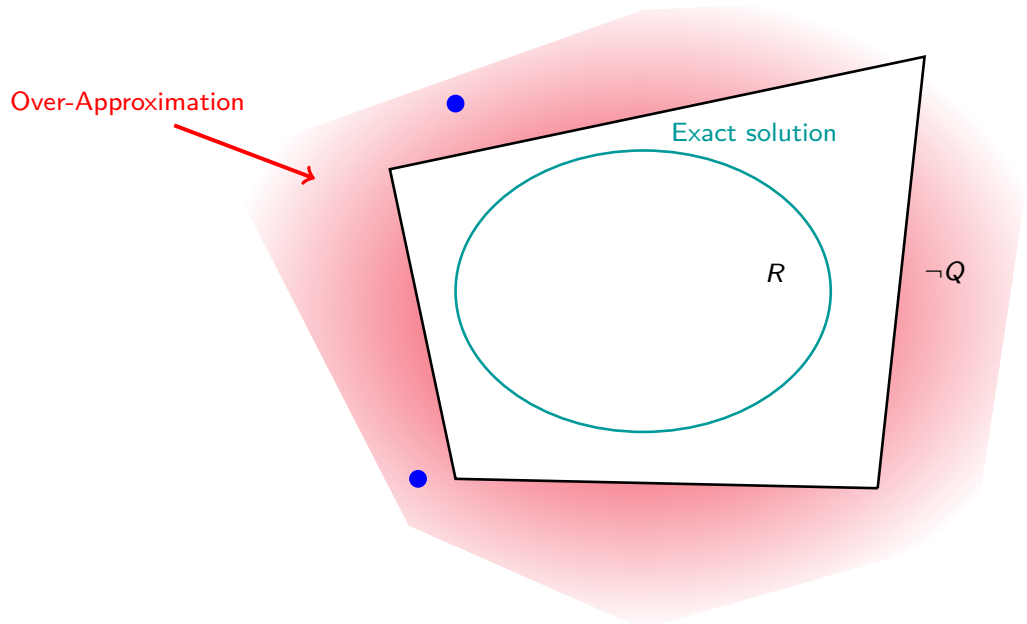
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



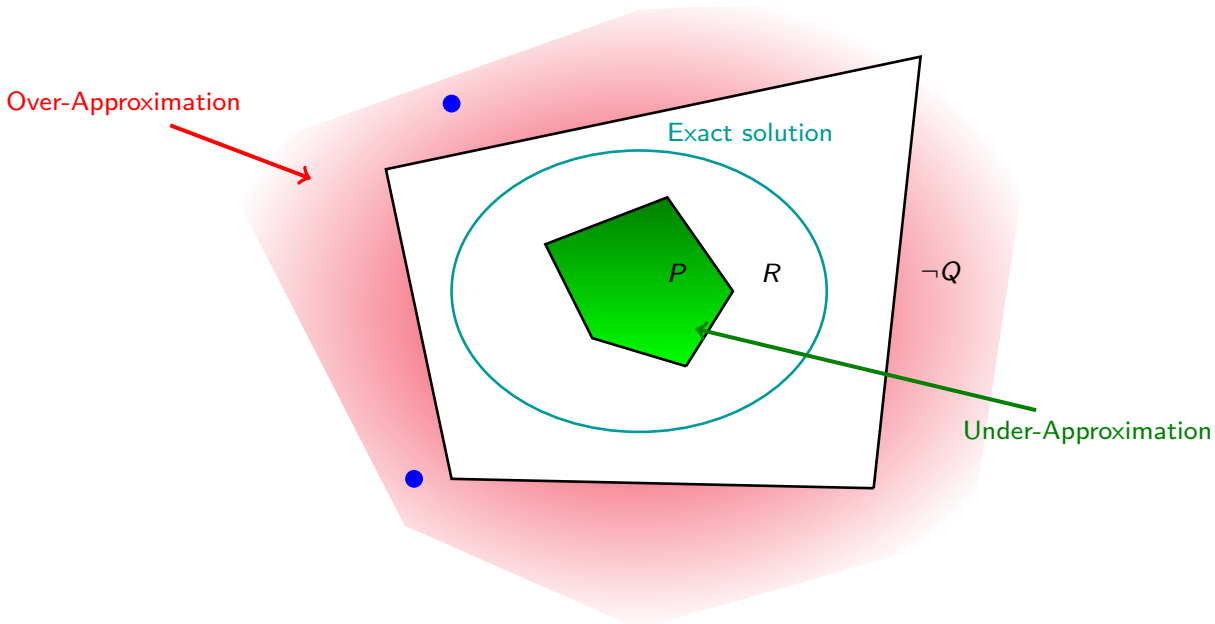
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



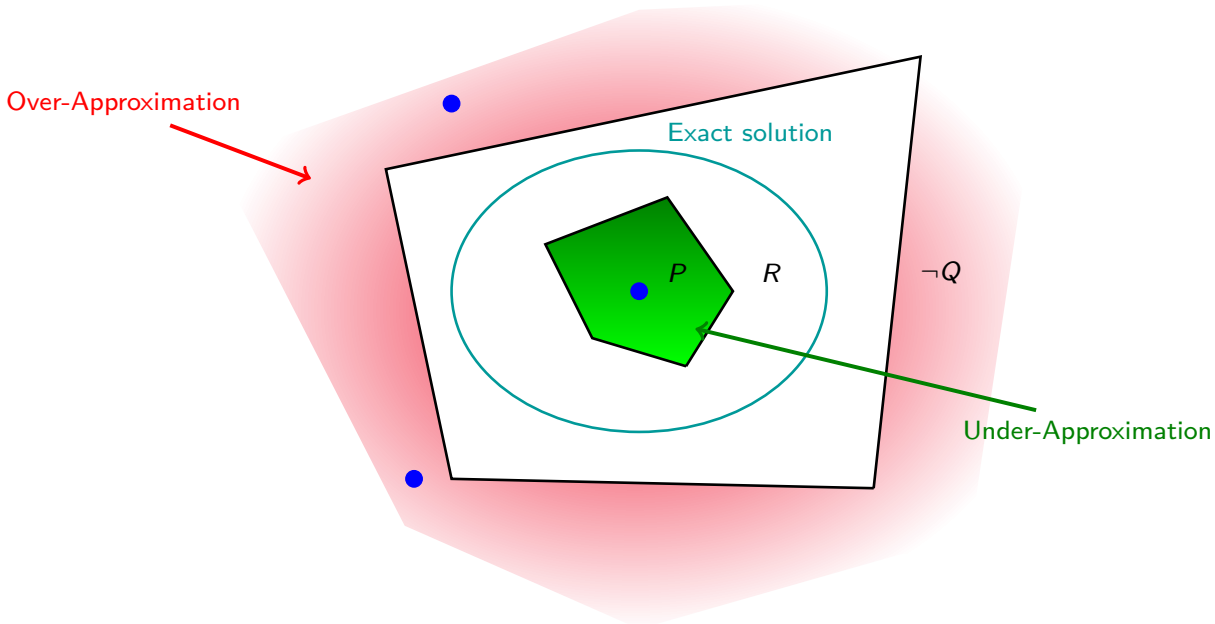
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



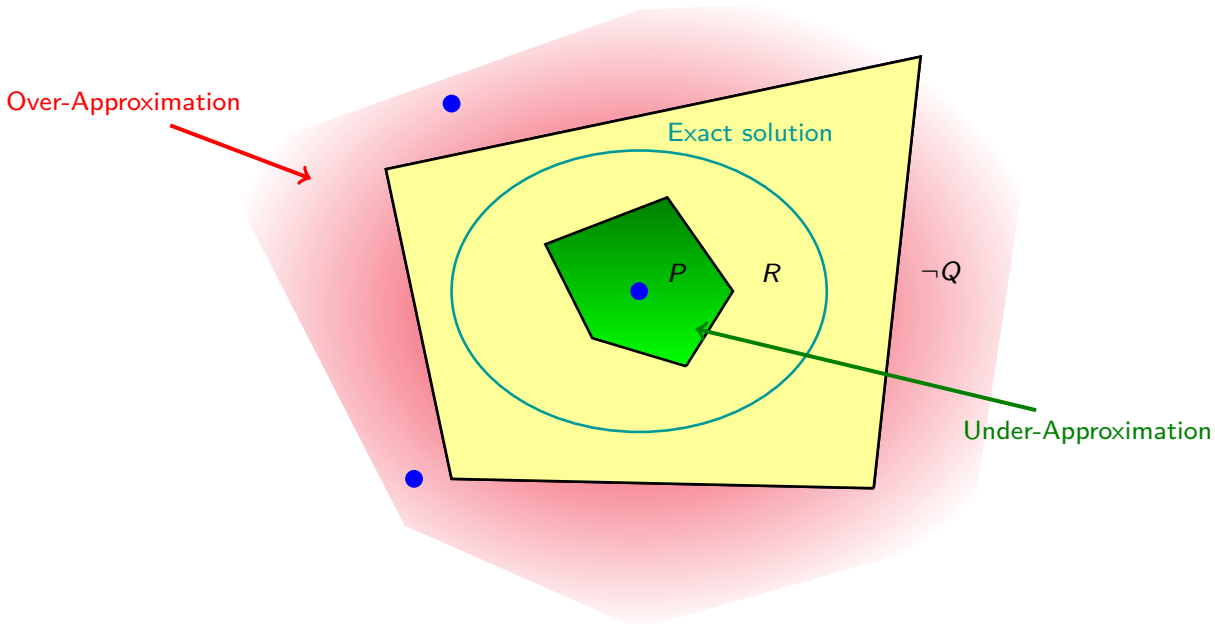
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



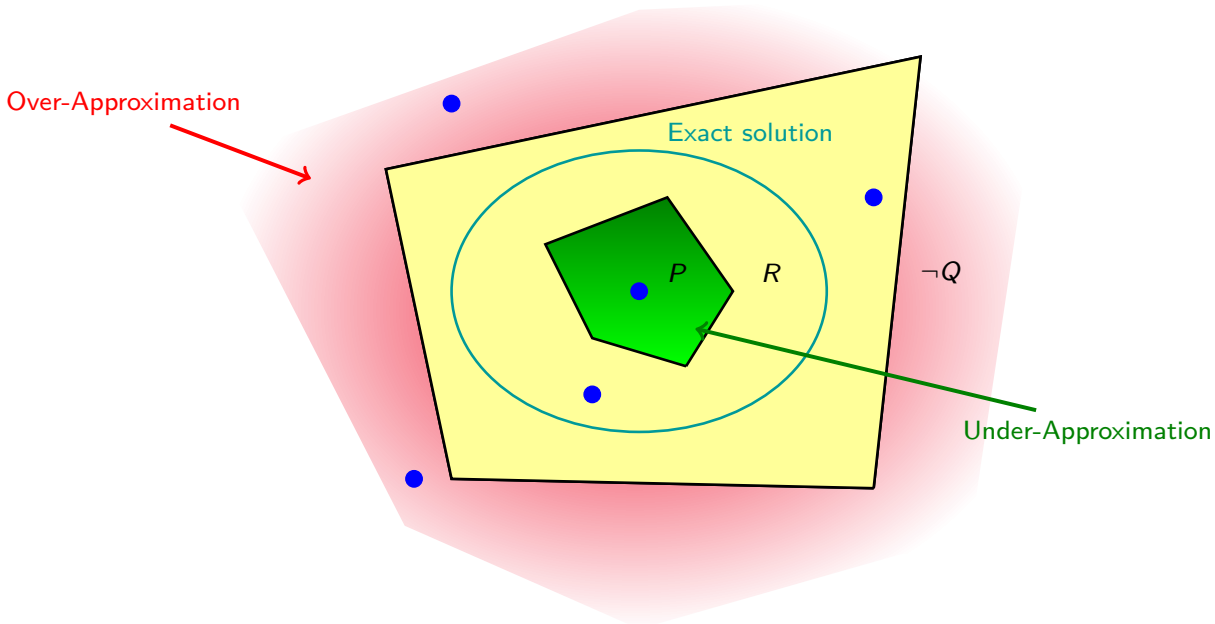
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



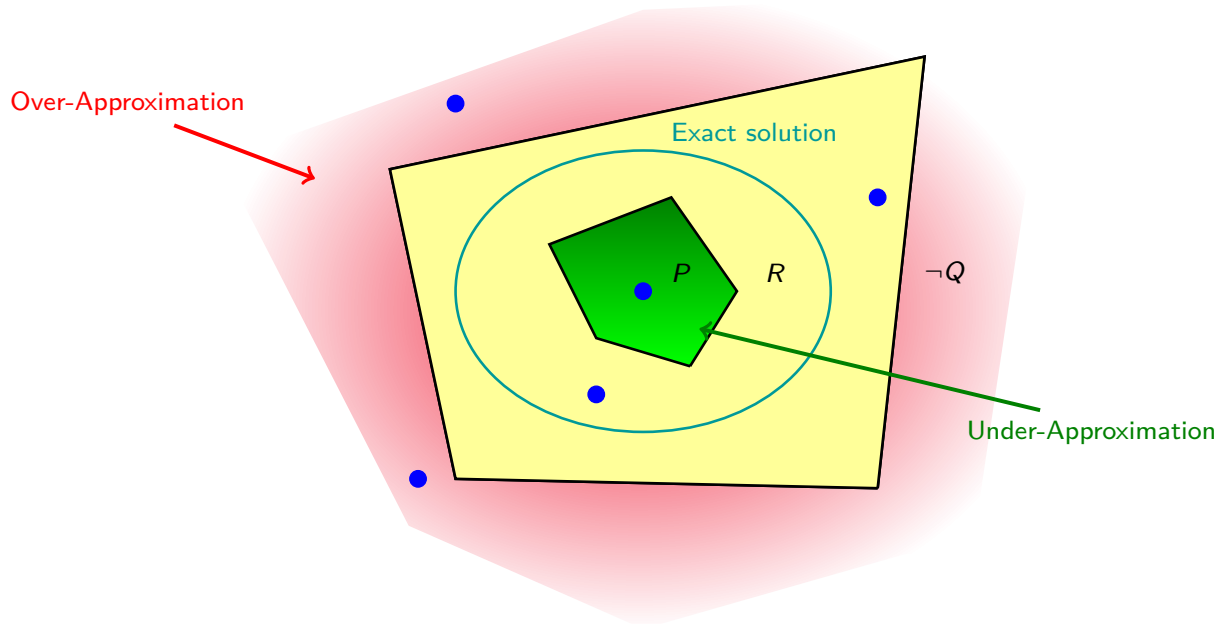
Over- and Under-approximations

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

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

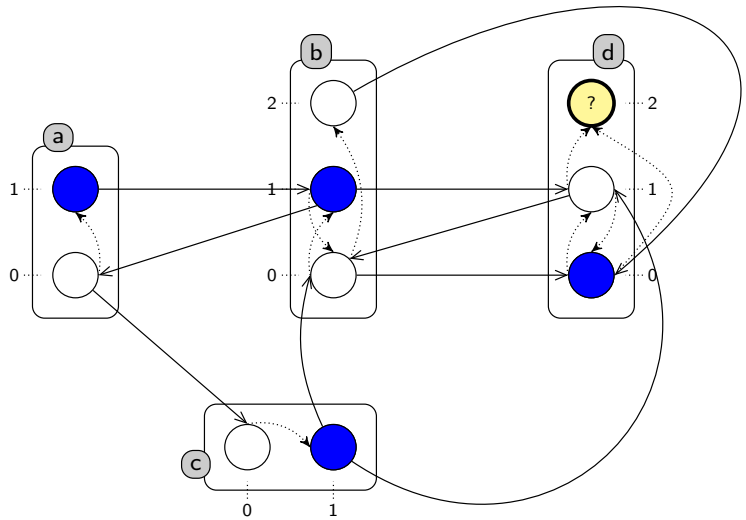
→ Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



Polynomial w.r.t. the number of sorts and
exponential w.r.t. the number of processes in each sort

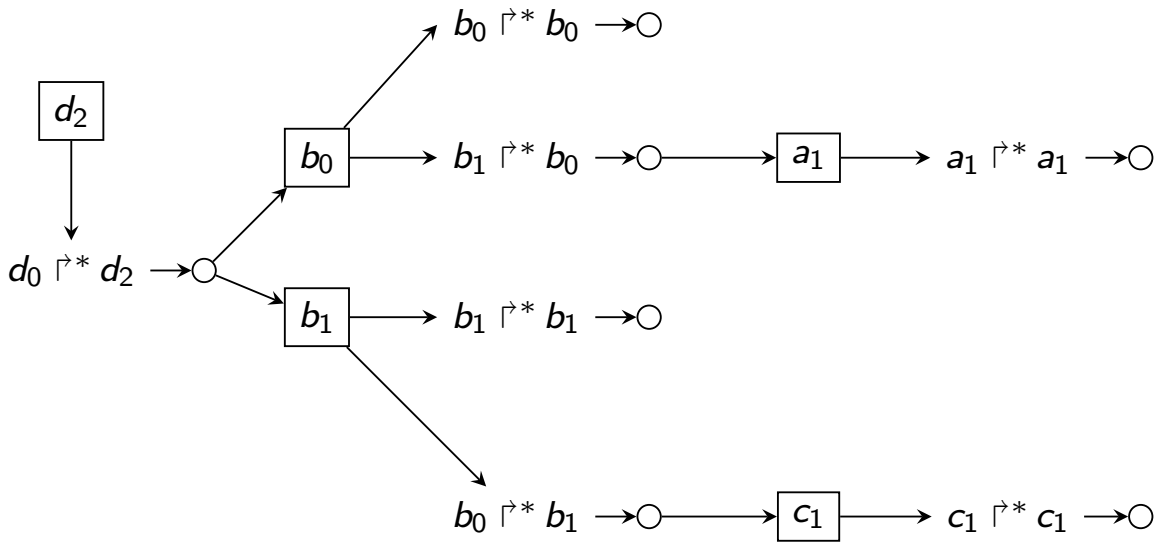
→ Efficient for big models with few levels of expression

Under-approximation



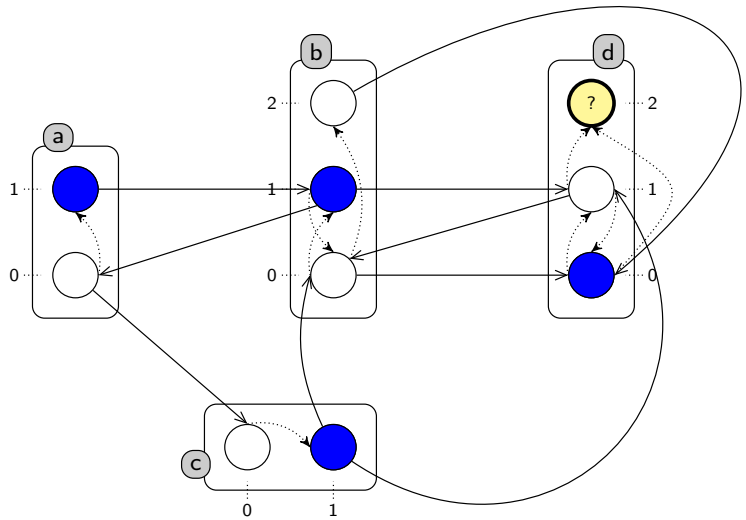
Sufficient condition:

- no cycle
- each objective has a solution



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

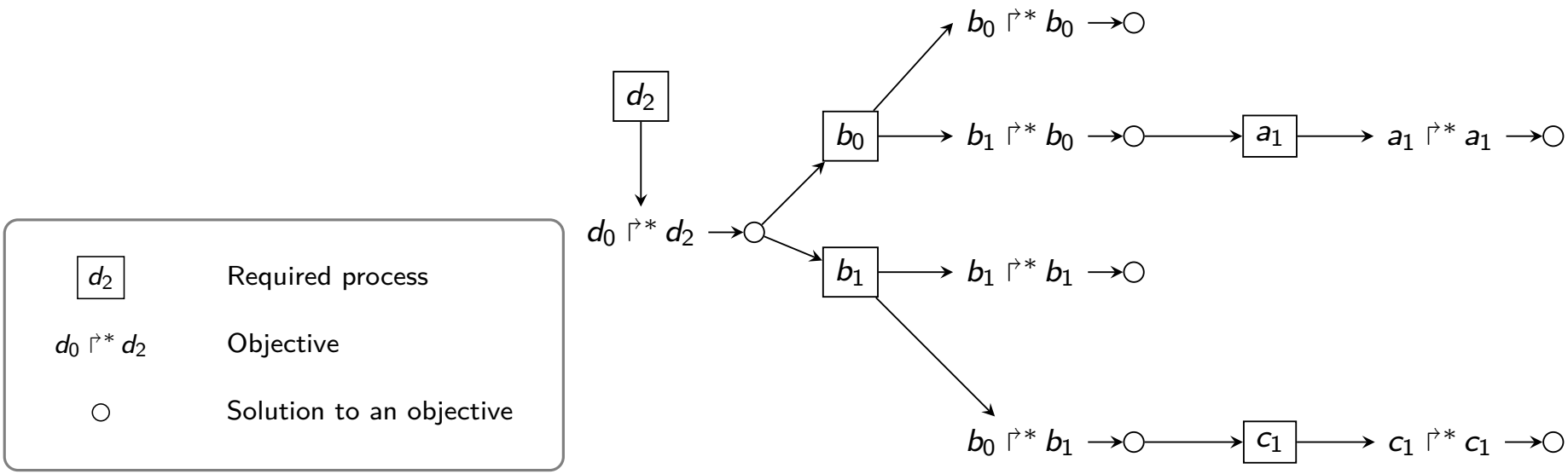
Under-approximation



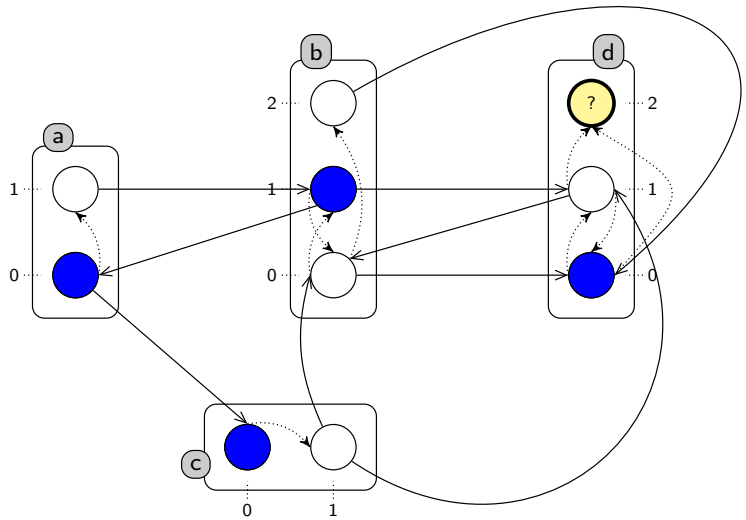
Sufficient condition:

- no cycle
- each objective has a solution

R is **true**

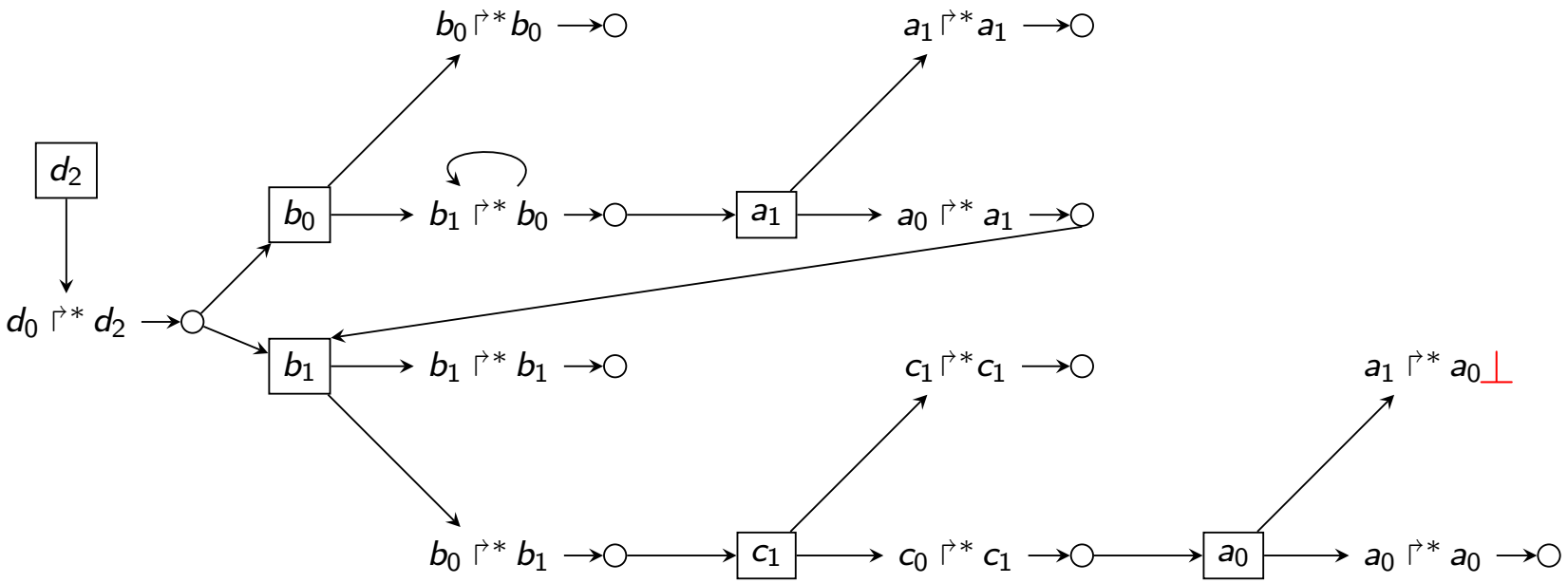


Under-approximation

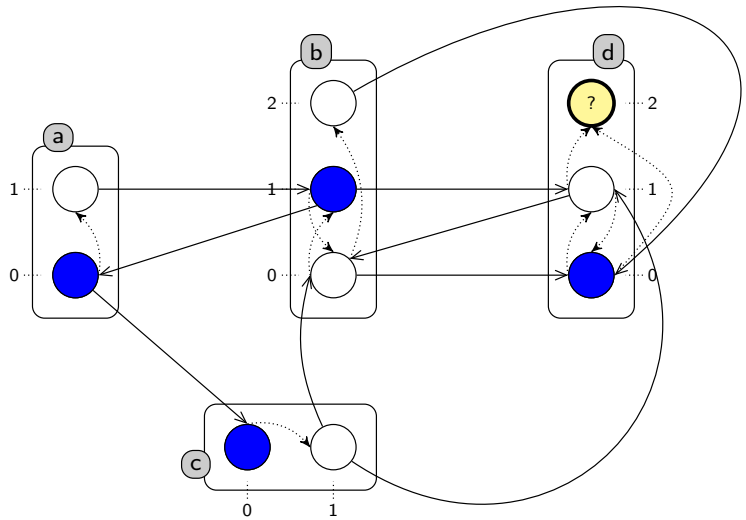


Sufficient condition:

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



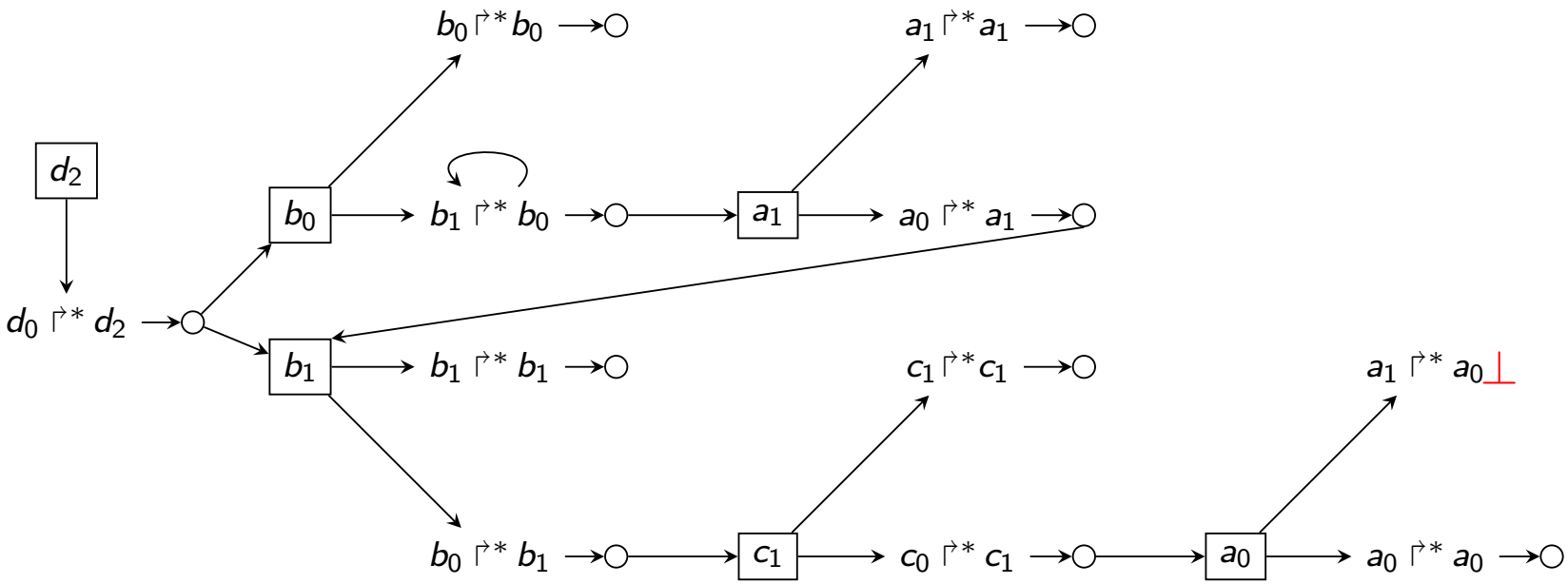
Under-approximation



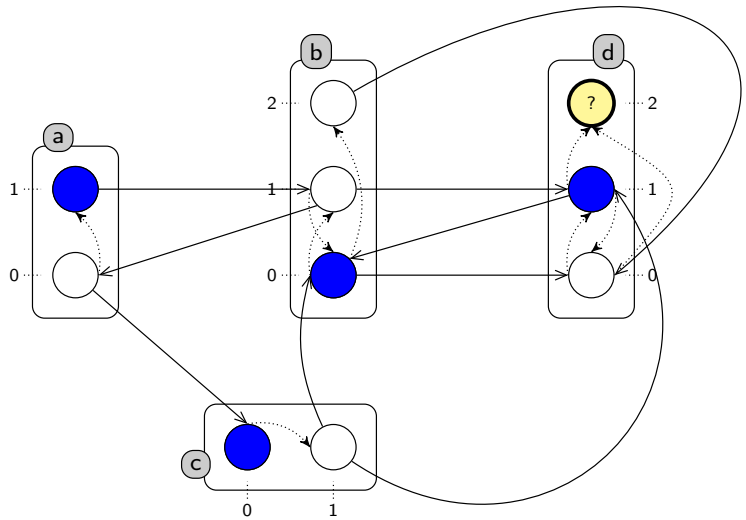
Sufficient condition:

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

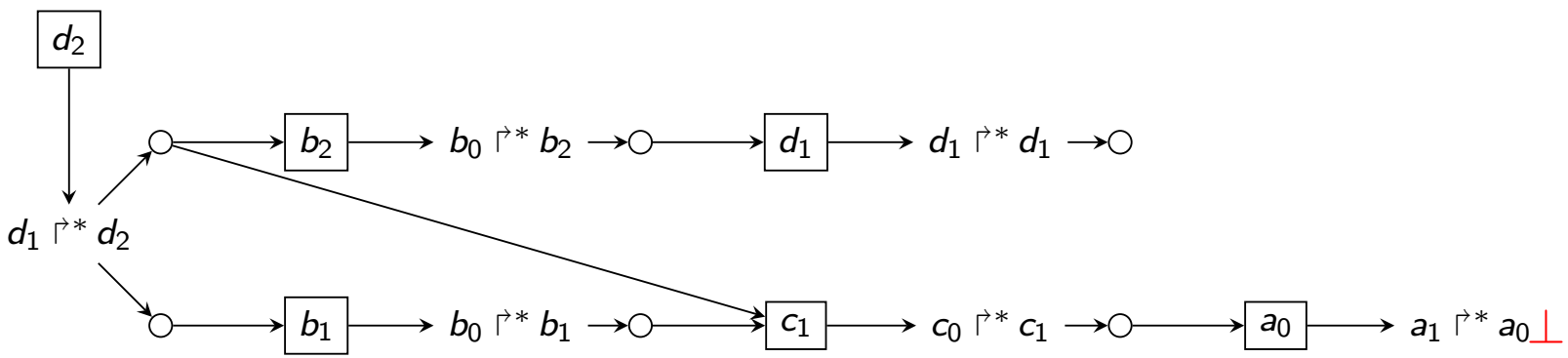
Inconclusive



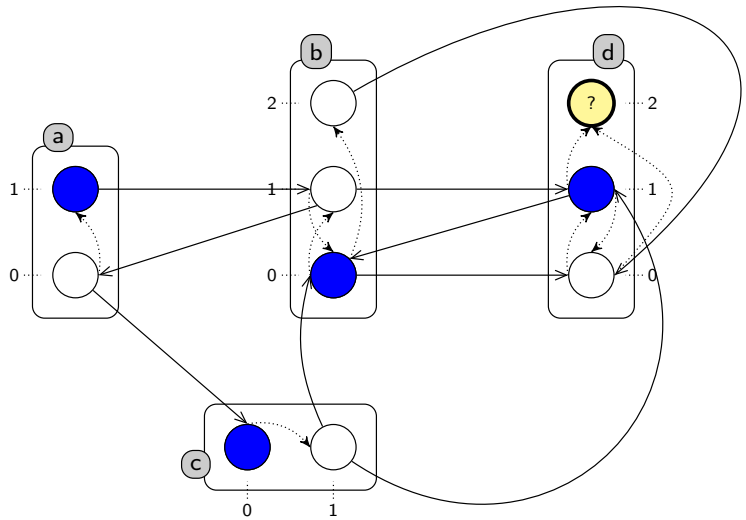
Over-approximation



Necessary condition:



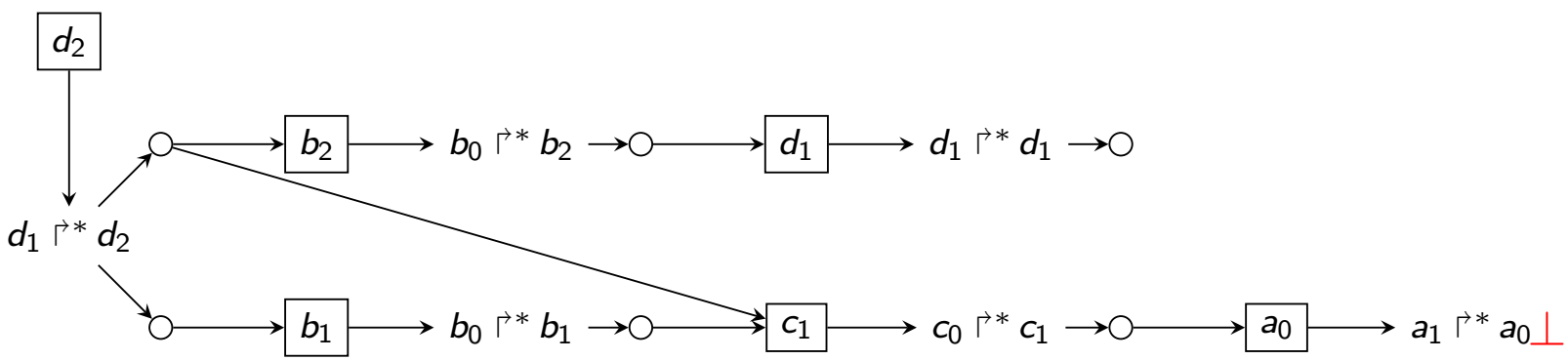
Over-approximation



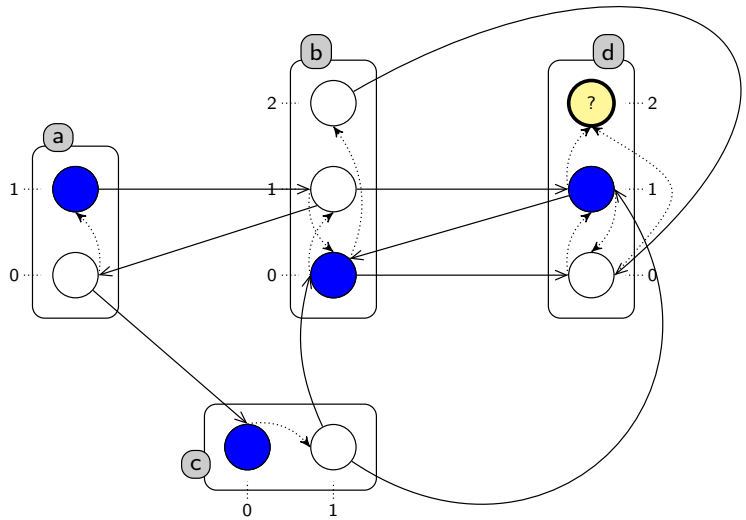
Necessary condition:

There exists a traversal with no cycle

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



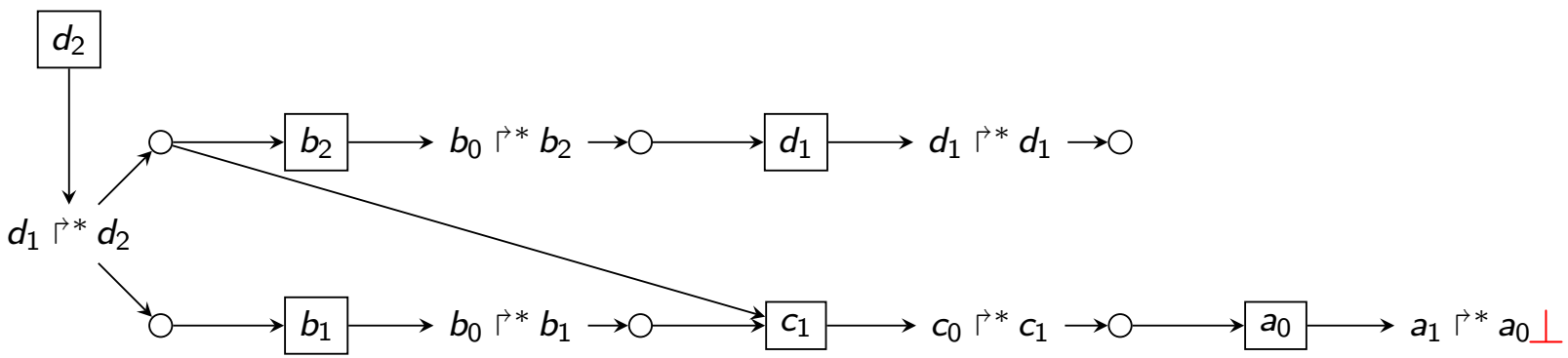
Over-approximation



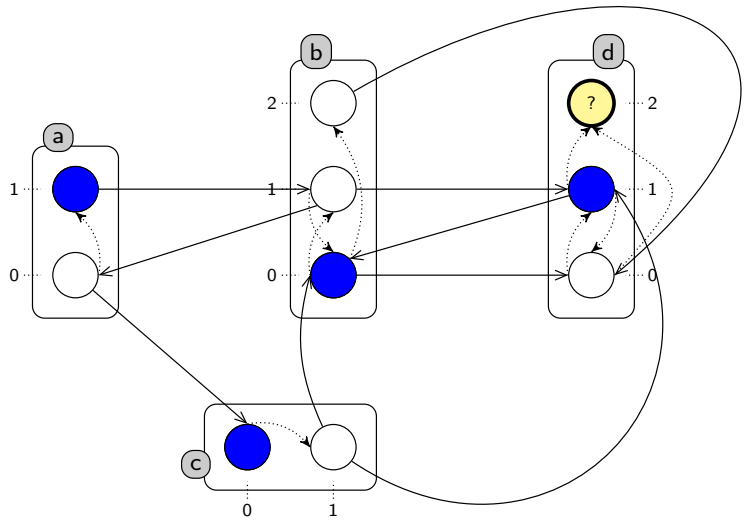
Necessary condition:

~~There exists a traversal~~ with no cycle

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



Over-approximation

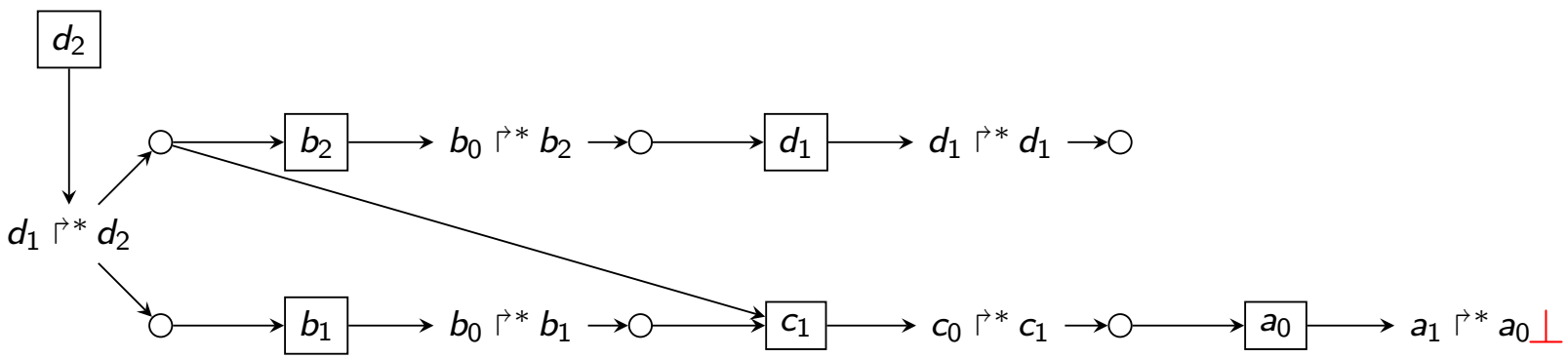


Necessary condition:

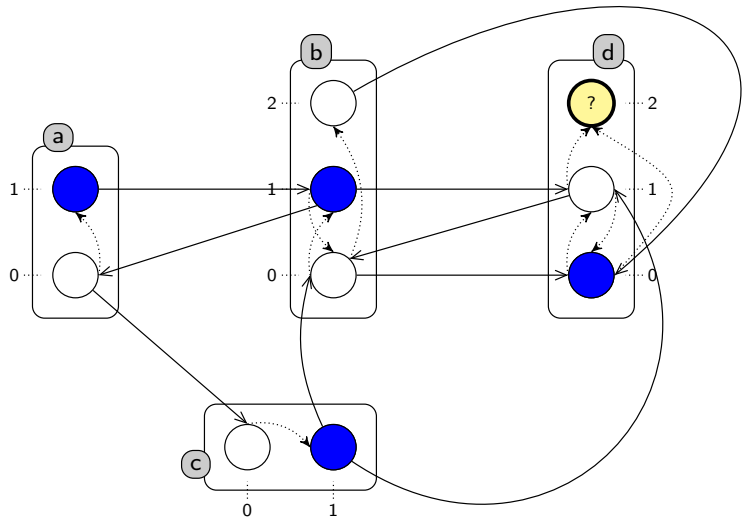
~~There exists a traversal~~ with no cycle

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

R is **false**



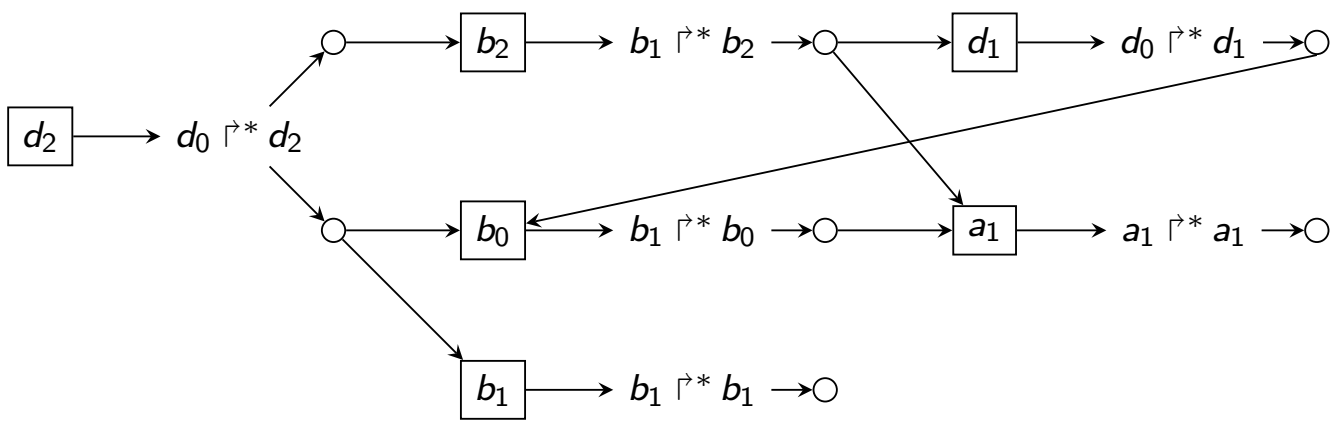
Over-approximation



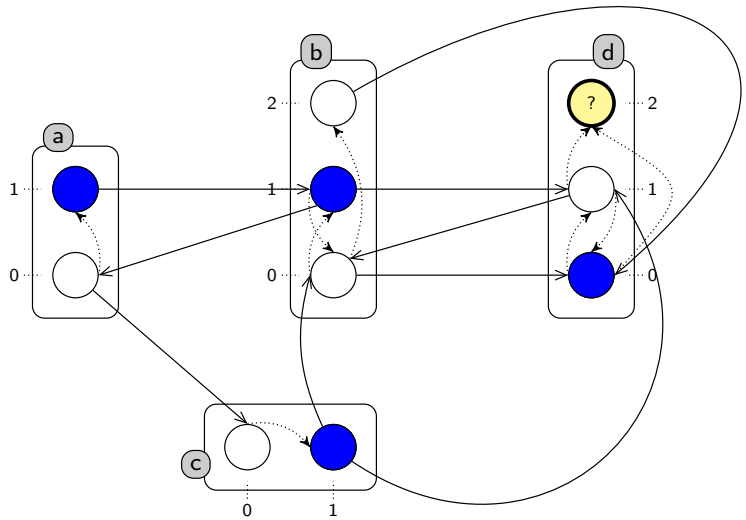
Necessary condition:

There exists a traversal with no cycle

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



Over-approximation

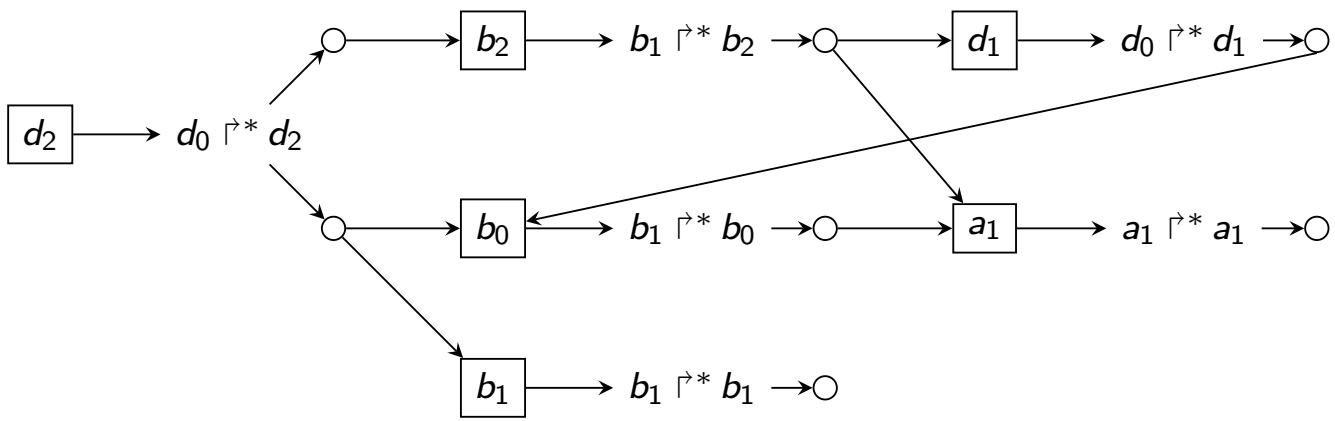


Necessary condition:

There exists a traversal with no cycle

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

Inconclusive



The Process Hitting modeling

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

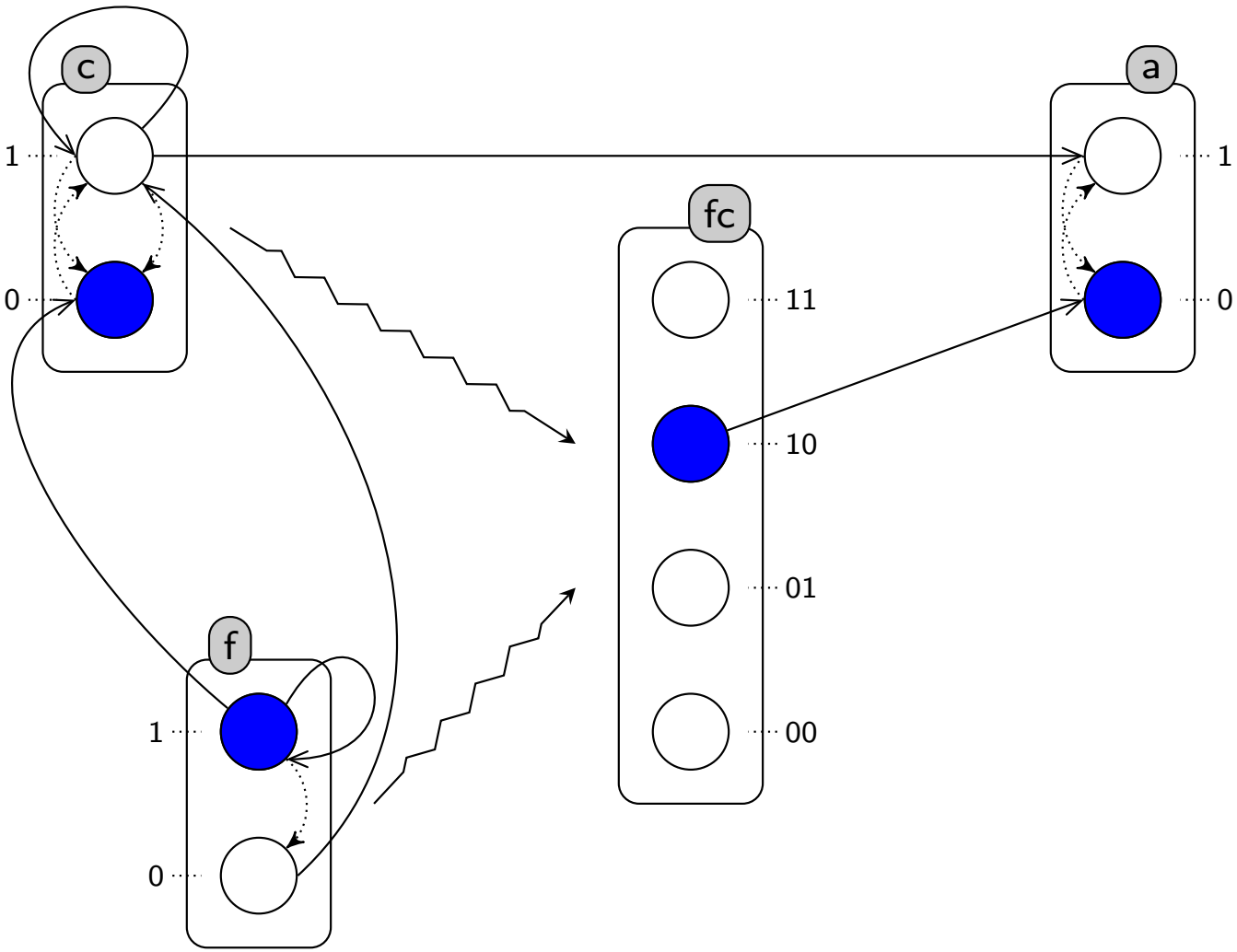
Model	Sorts	Procs	Actions	States	Biocham[1]	libddd[2]	PINT
egfr20	35	196	670	2^{64}	[3s–∞]	[1s–150s]	**0.007s**
tcrsig40	54	156	301	2^{73}	[1s–∞]	[0.6s–∞]	**0.004s**
tcrsig94	133	448	1124	2^{194}	∞	∞	**0.030s**
egfr104	193	748	2356	2^{320}	∞	∞	**0.050s**

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

egfr20: [Epidermal Growth Factor Receptor, by Özgür Sahin *et al.*]
egfr104: [Epidermal Growth Factor Receptor, by Regina Samaga *et al.*]
tcrsig40: [T-Cell Receptor Signaling, by Steffen Klamt *et al.*]
tcrsig94: [T-Cell Receptor Signaling, by Julio Saez-Rodriguez *et al.*]

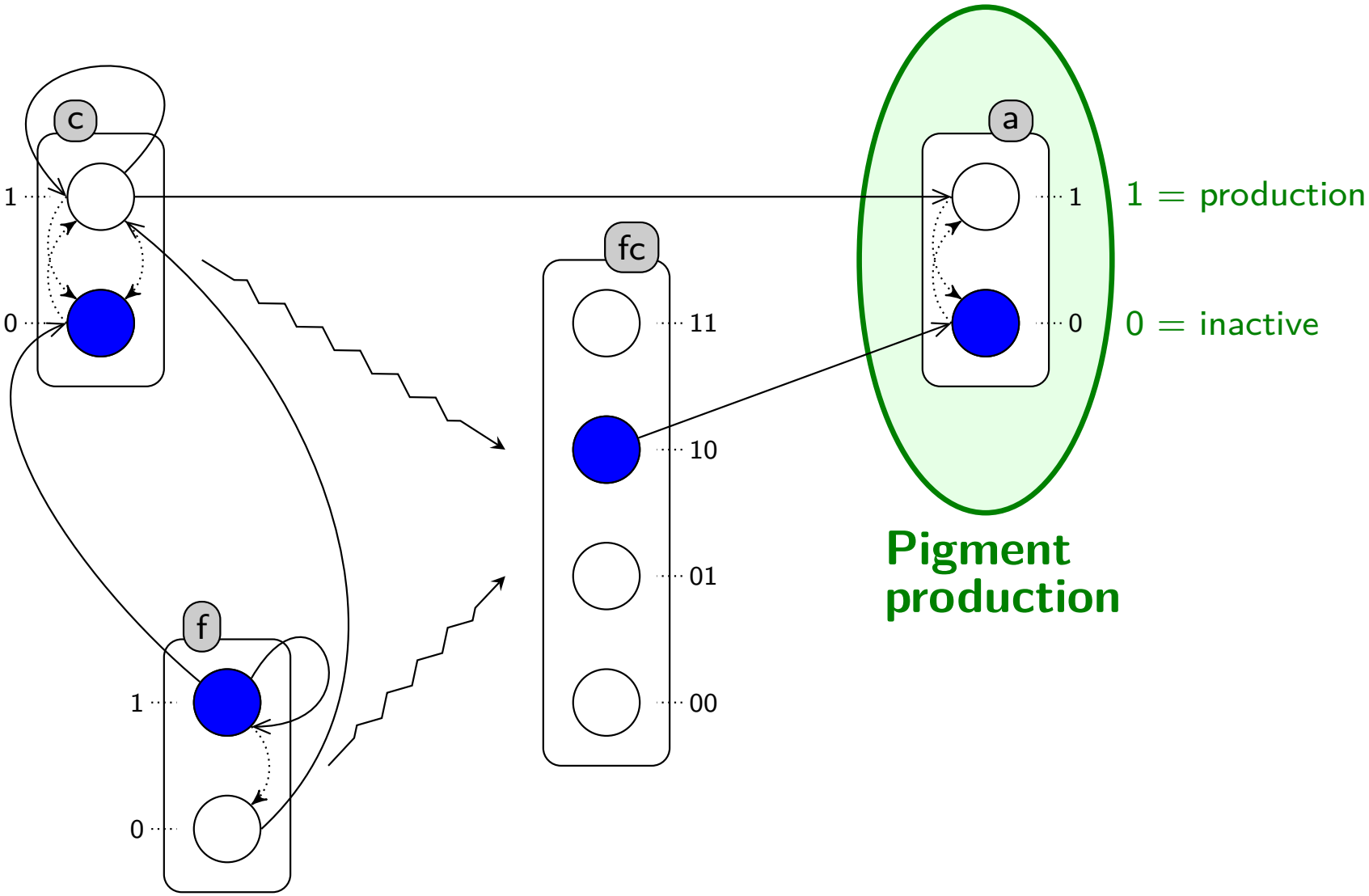
The Metazoan Segmentation Model

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



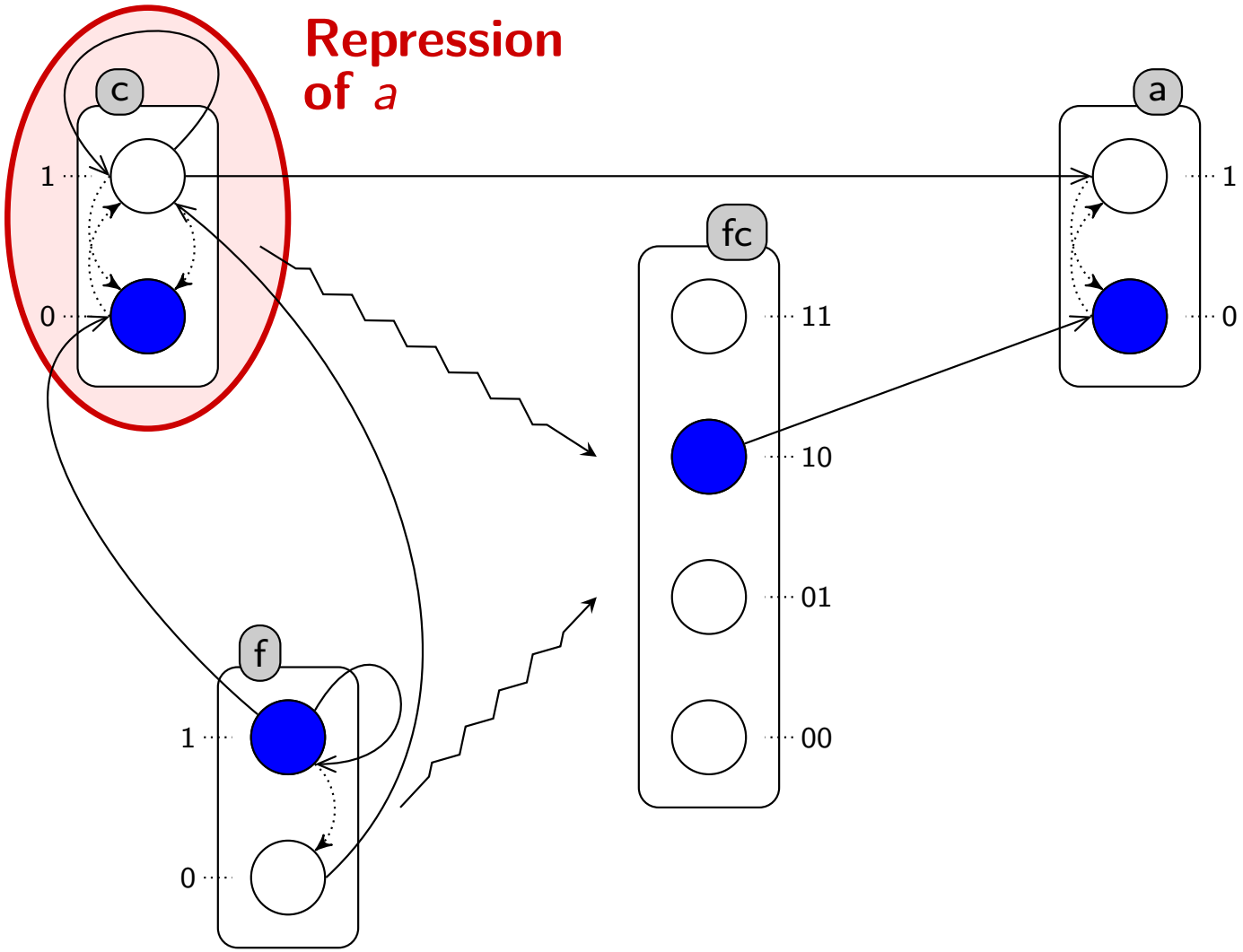
The Metazoan Segmentation Model

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



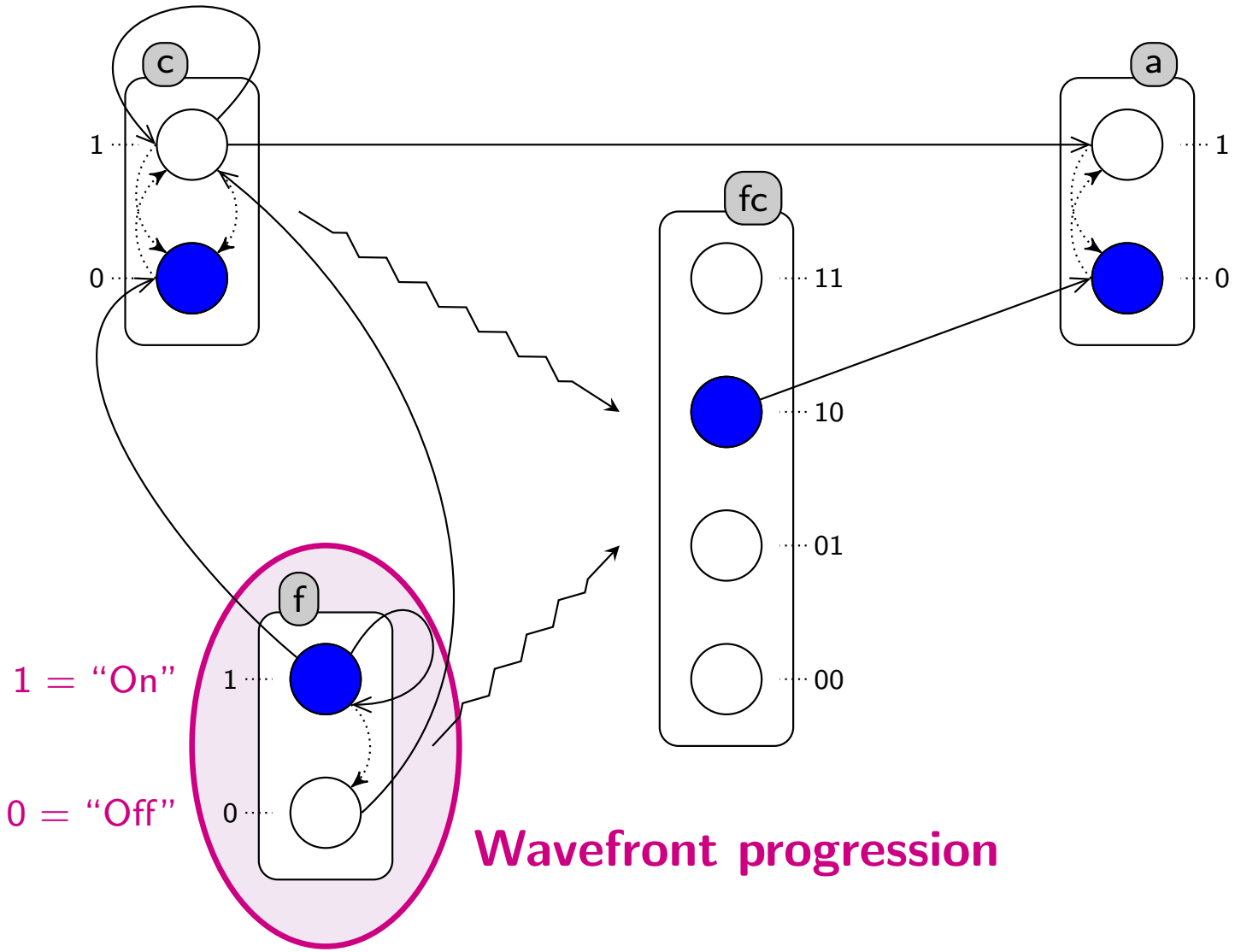
The Metazoan Segmentation Model

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



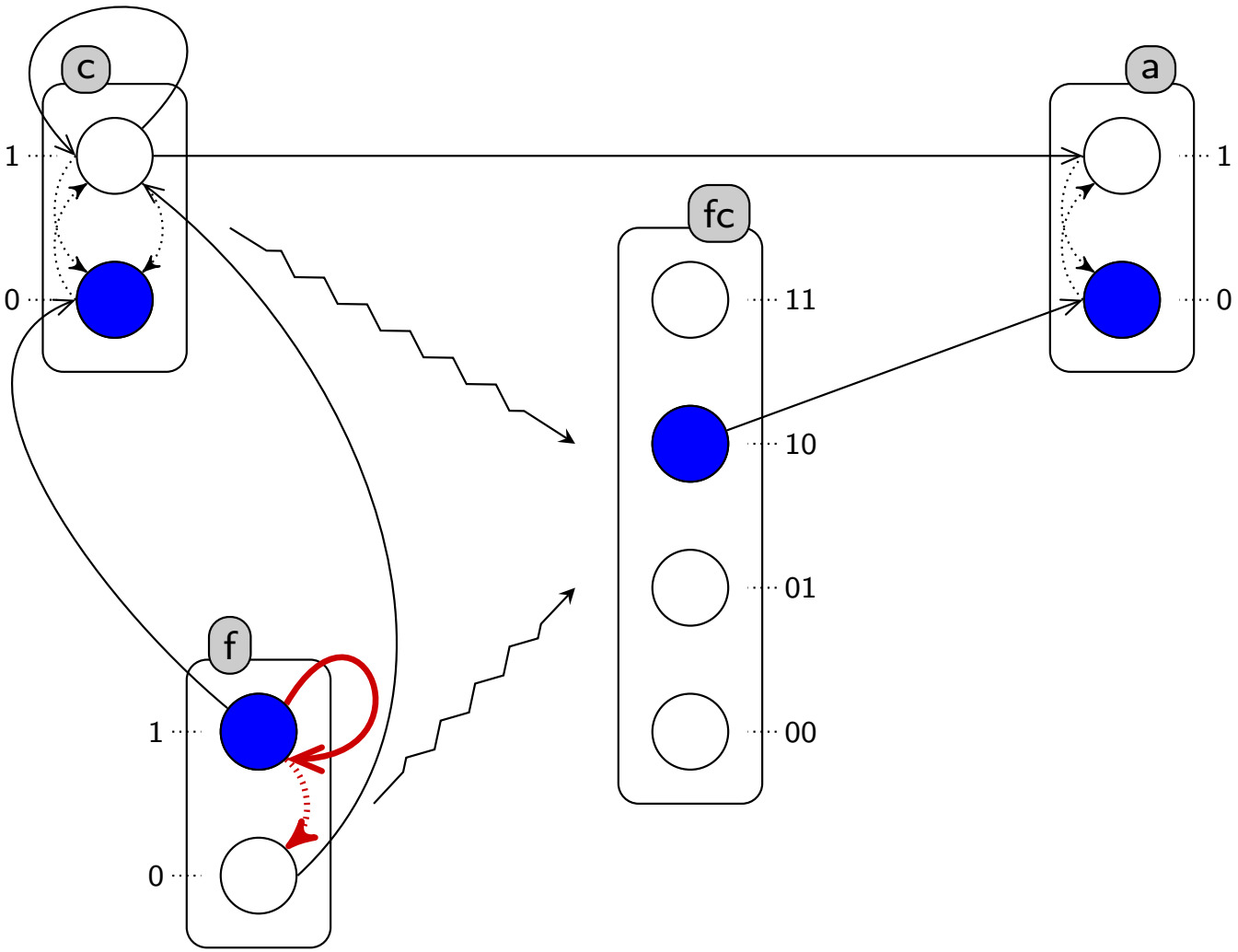
The Metazoan Segmentation Model

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



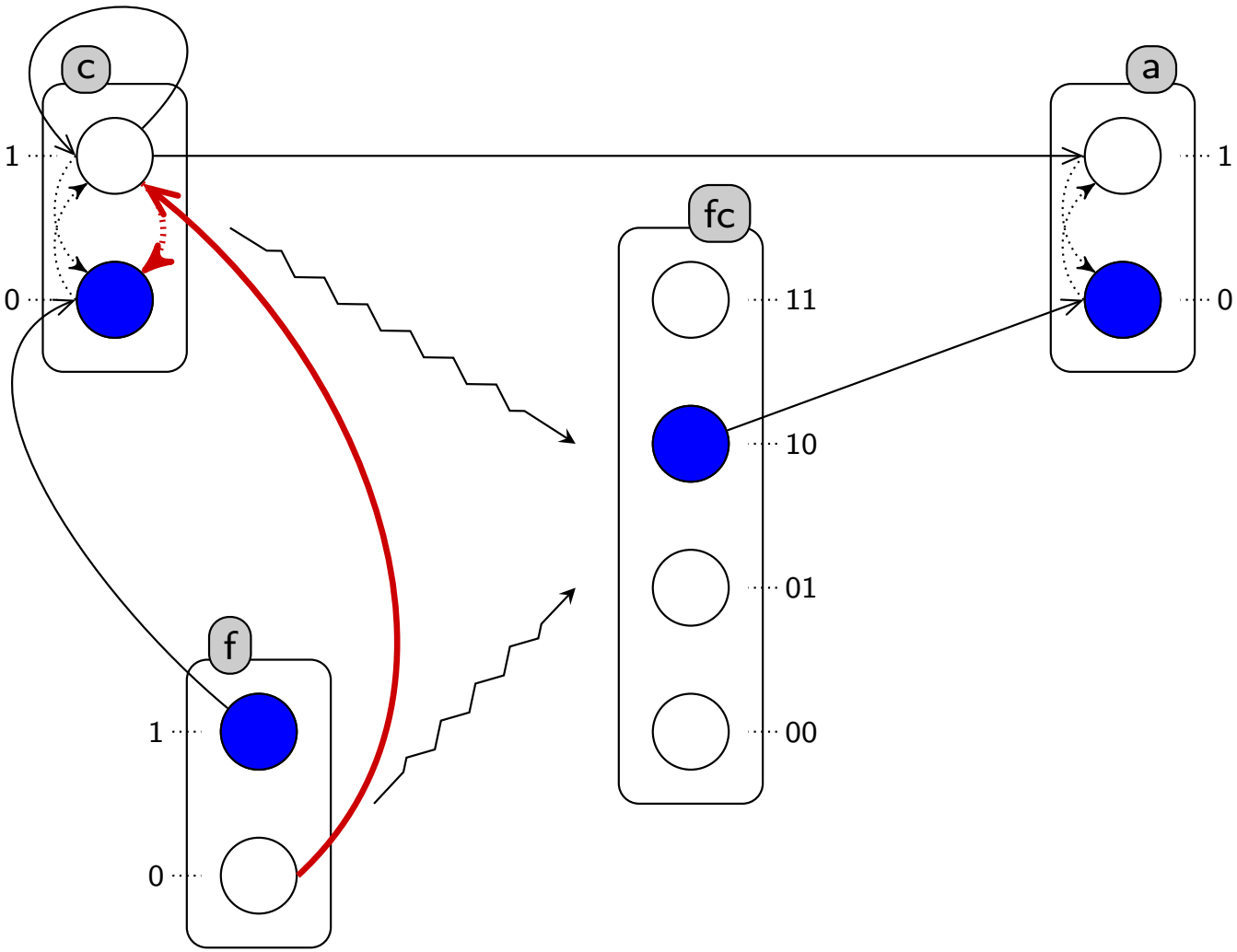
The Metazoan Segmentation Model

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



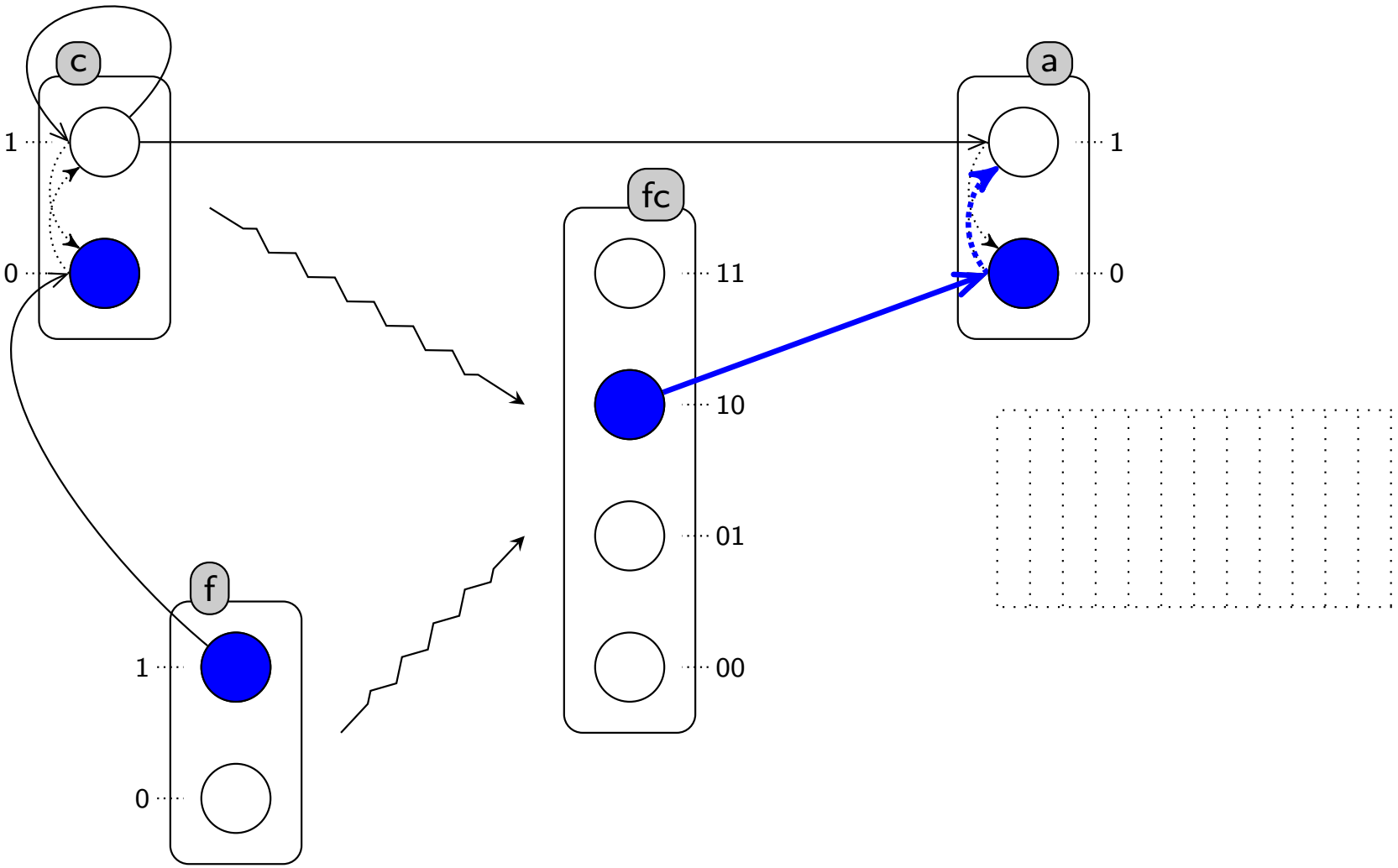
The Metazoan Segmentation Model

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



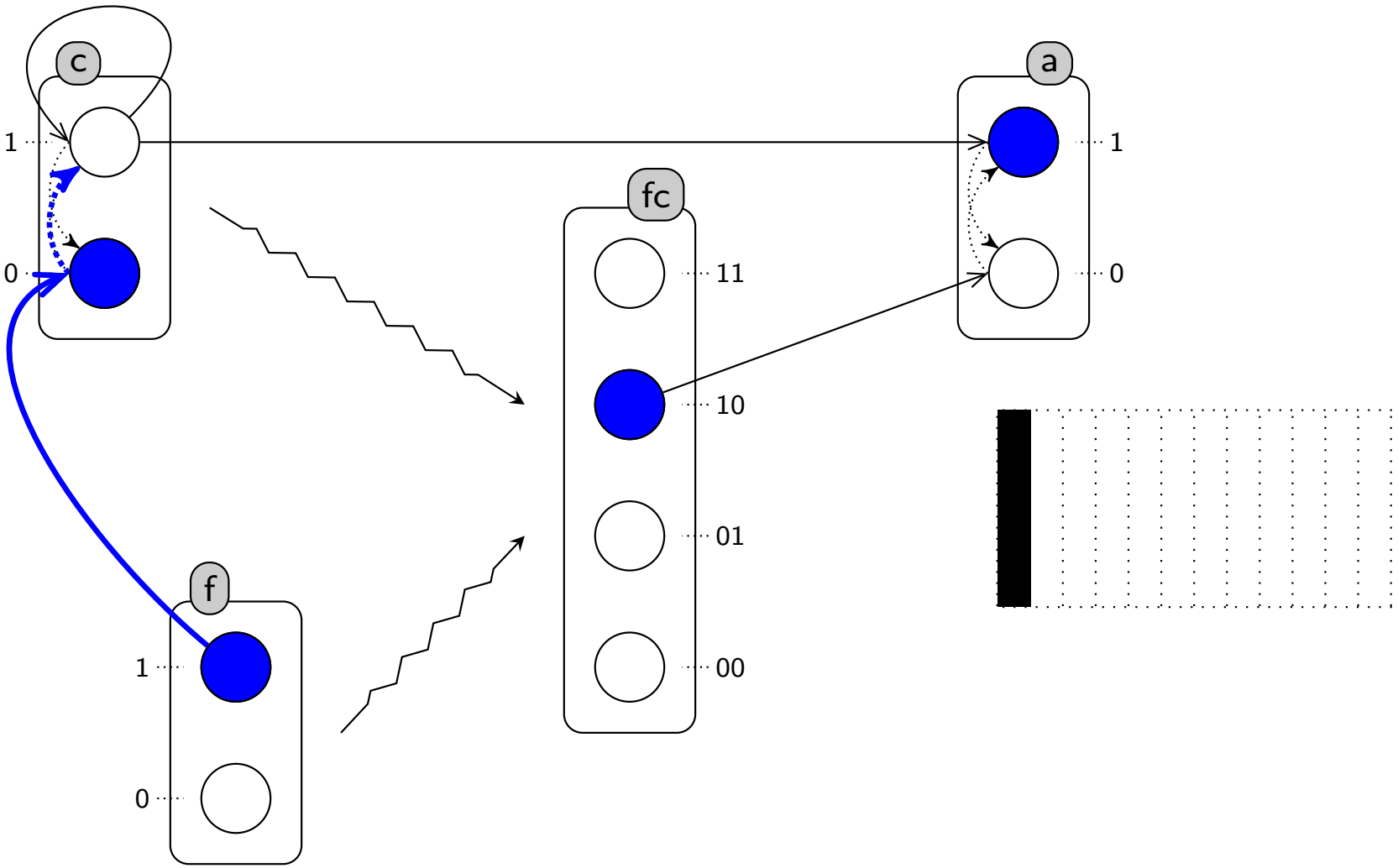
The Metazoan Segmentation Model

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



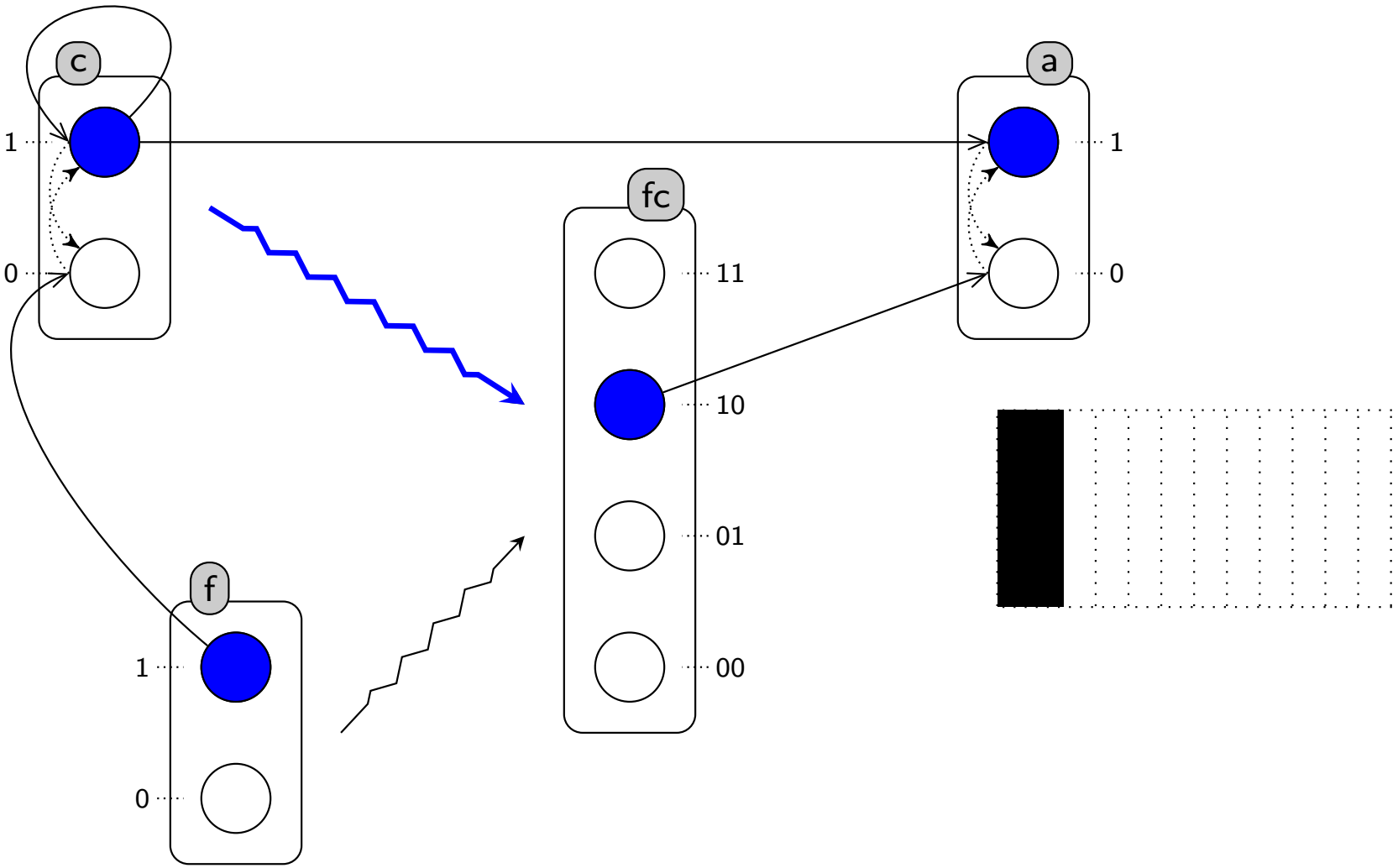
The Metazoan Segmentation Model

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



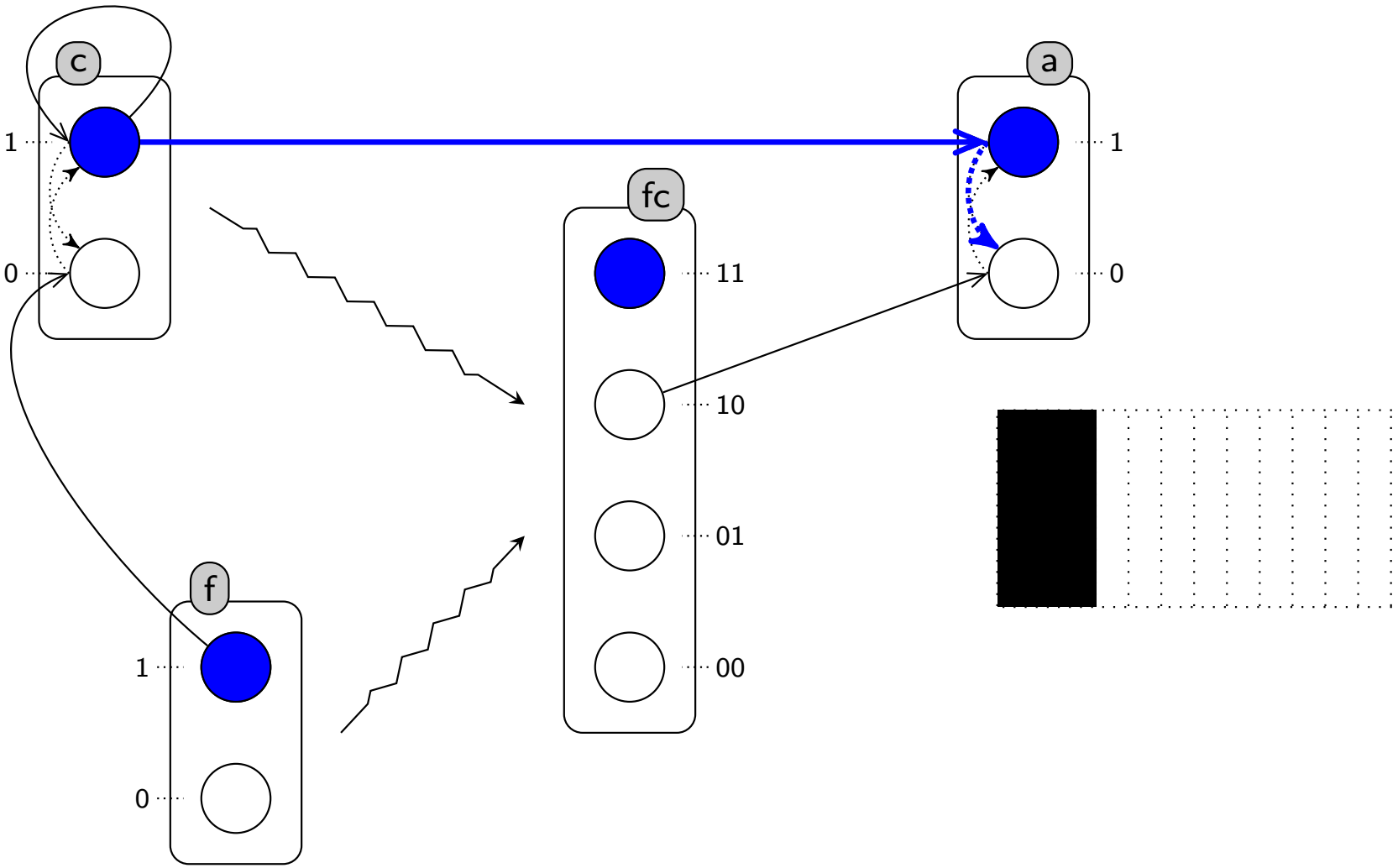
The Metazoan Segmentation Model

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



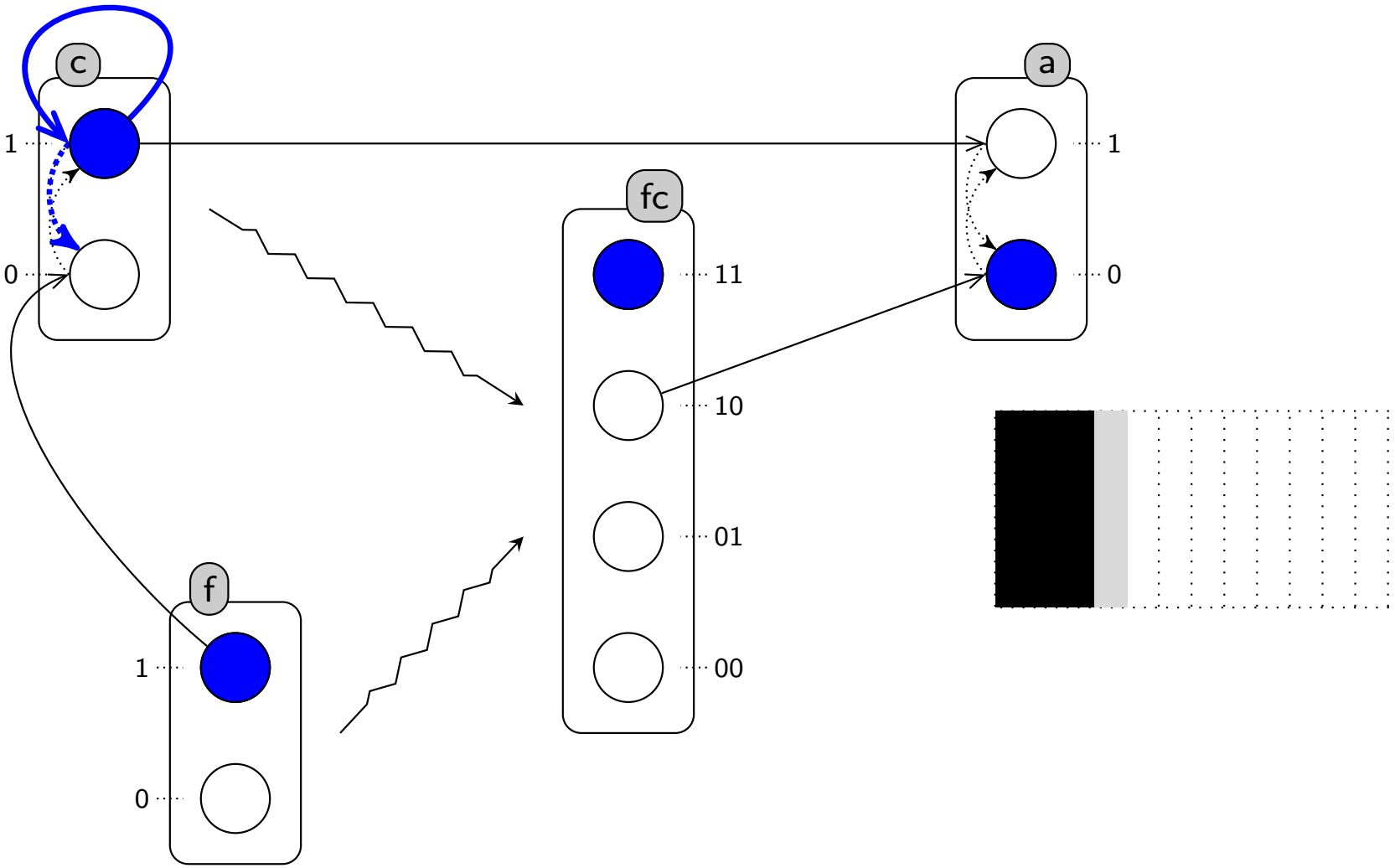
The Metazoan Segmentation Model

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



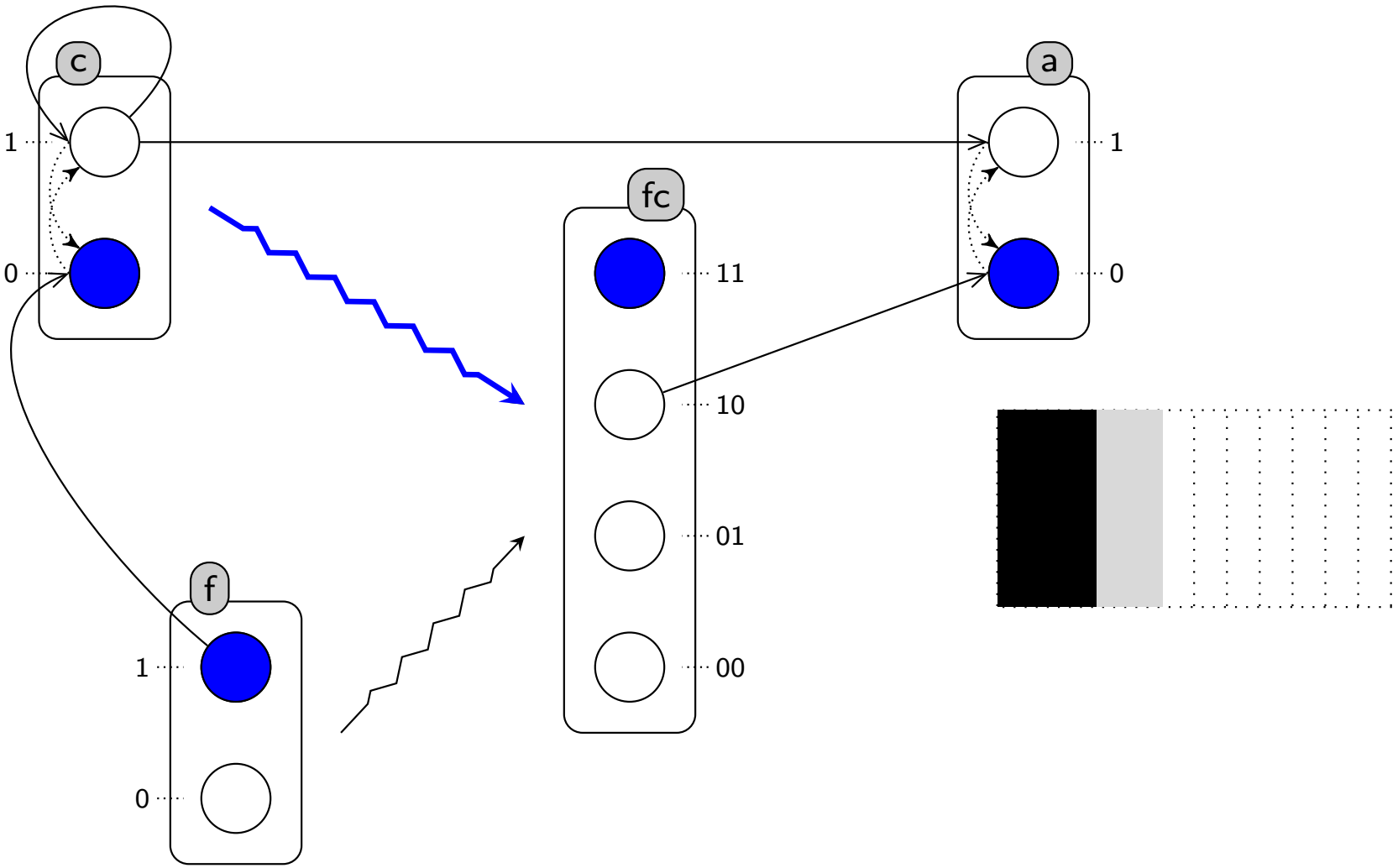
The Metazoan Segmentation Model

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



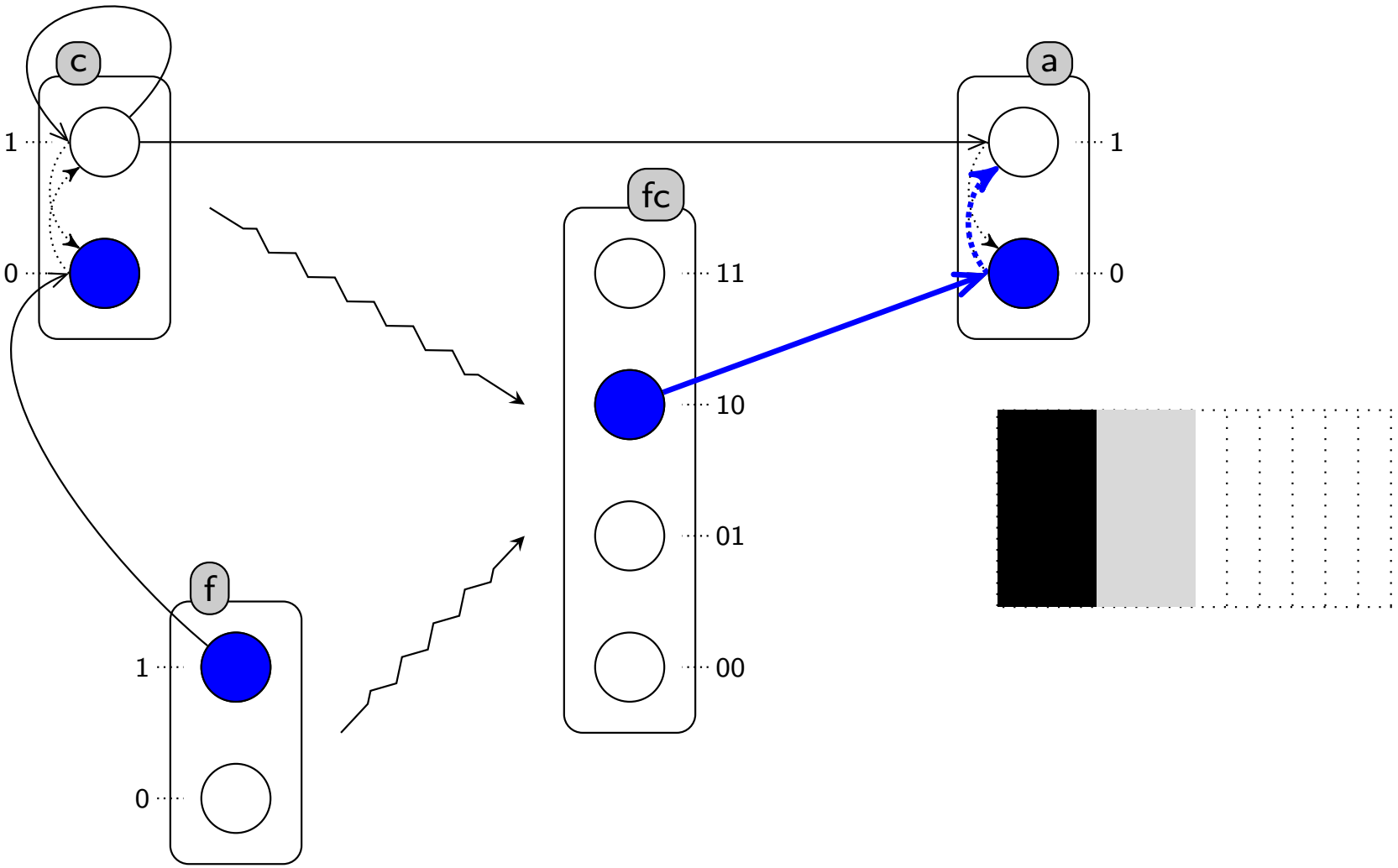
The Metazoan Segmentation Model

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



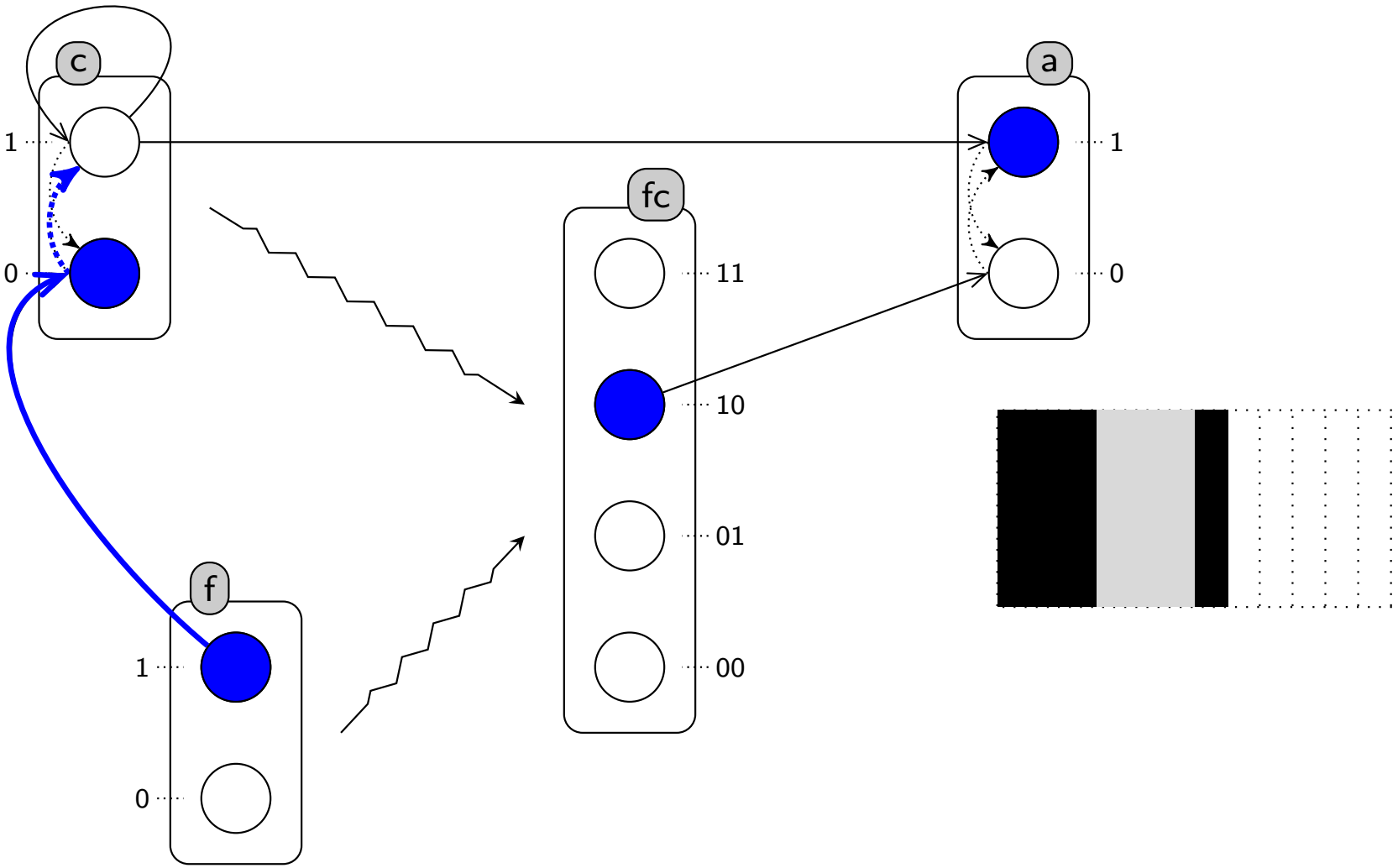
The Metazoan Segmentation Model

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



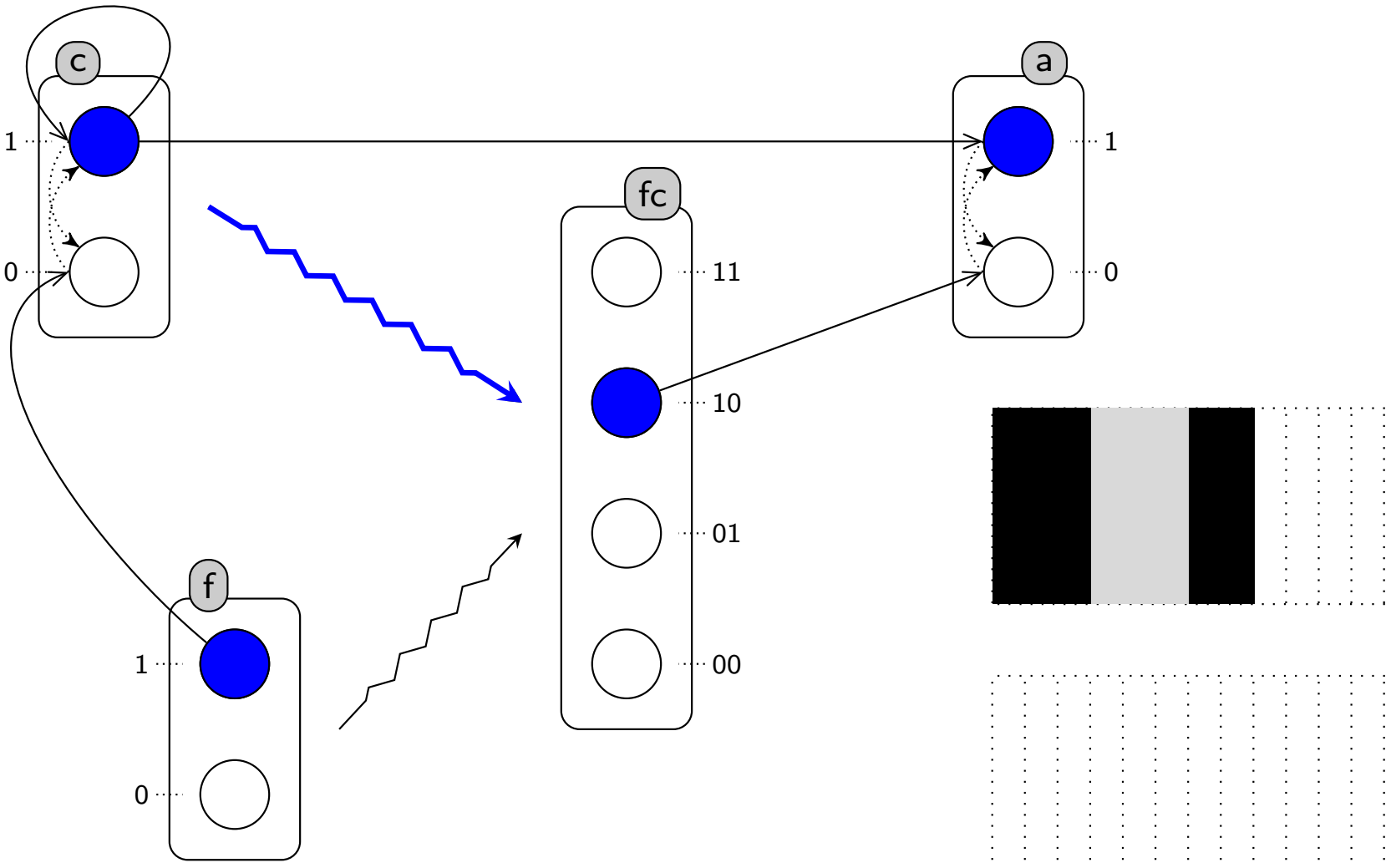
The Metazoan Segmentation Model

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



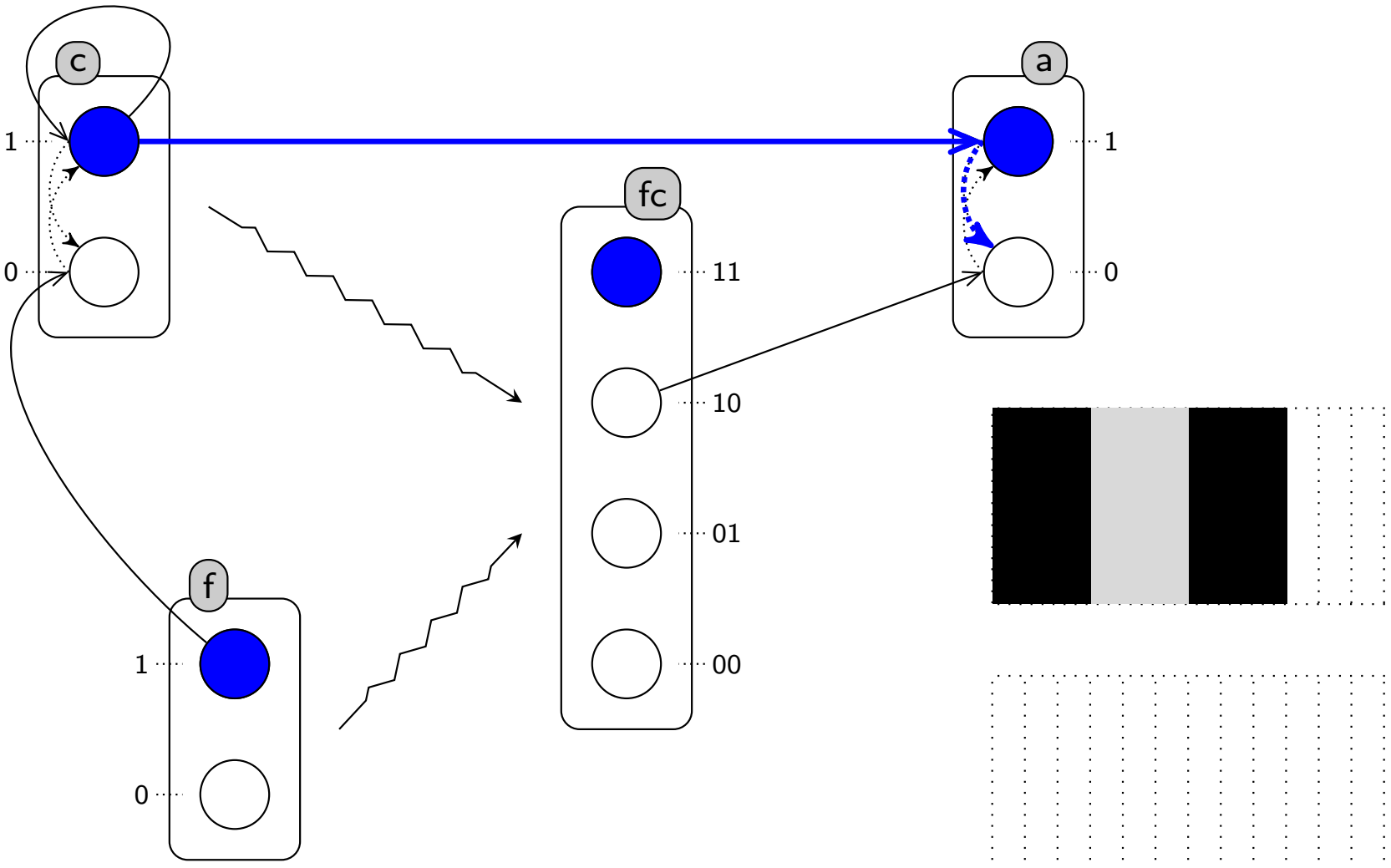
The Metazoan Segmentation Model

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



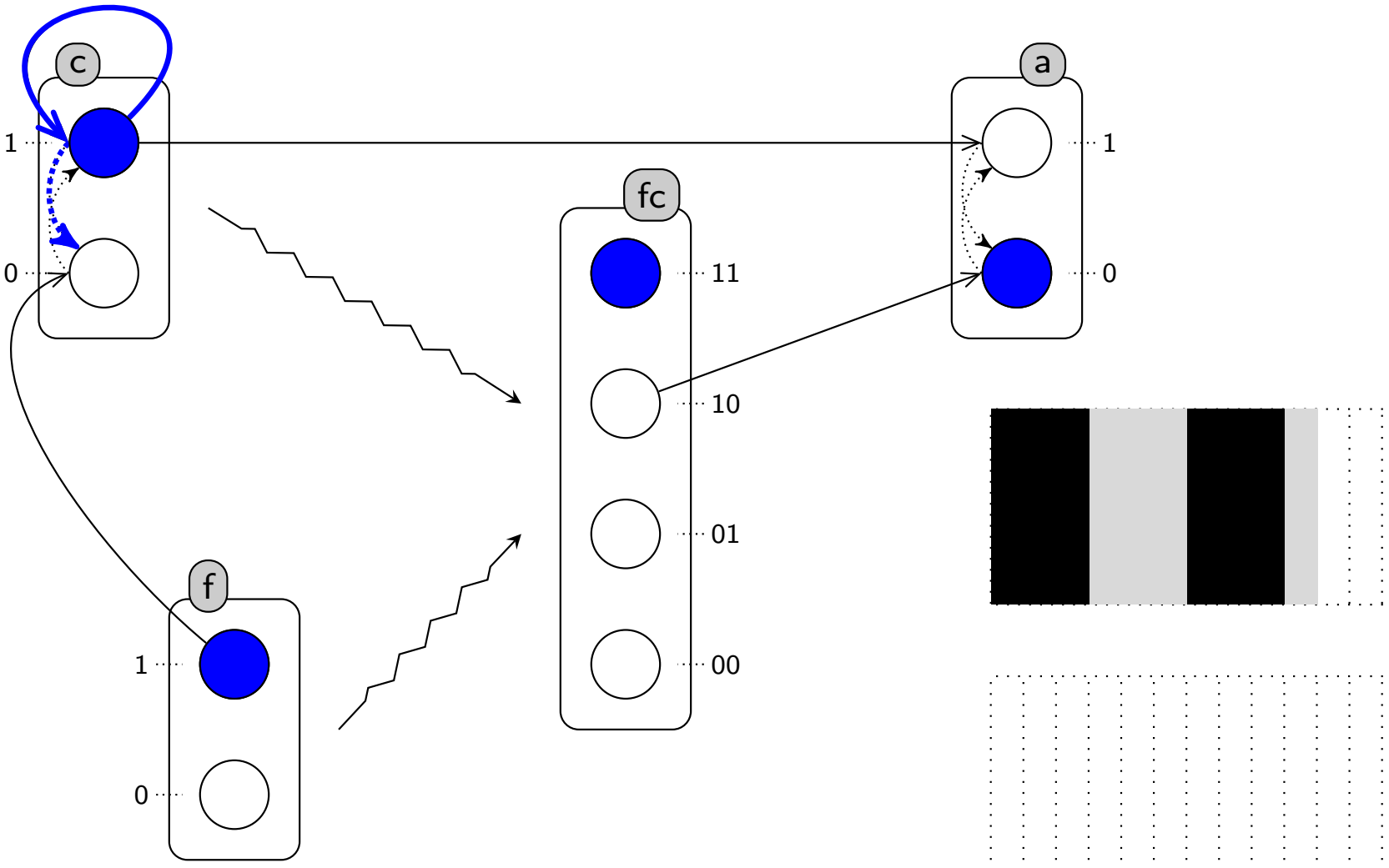
The Metazoan Segmentation Model

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



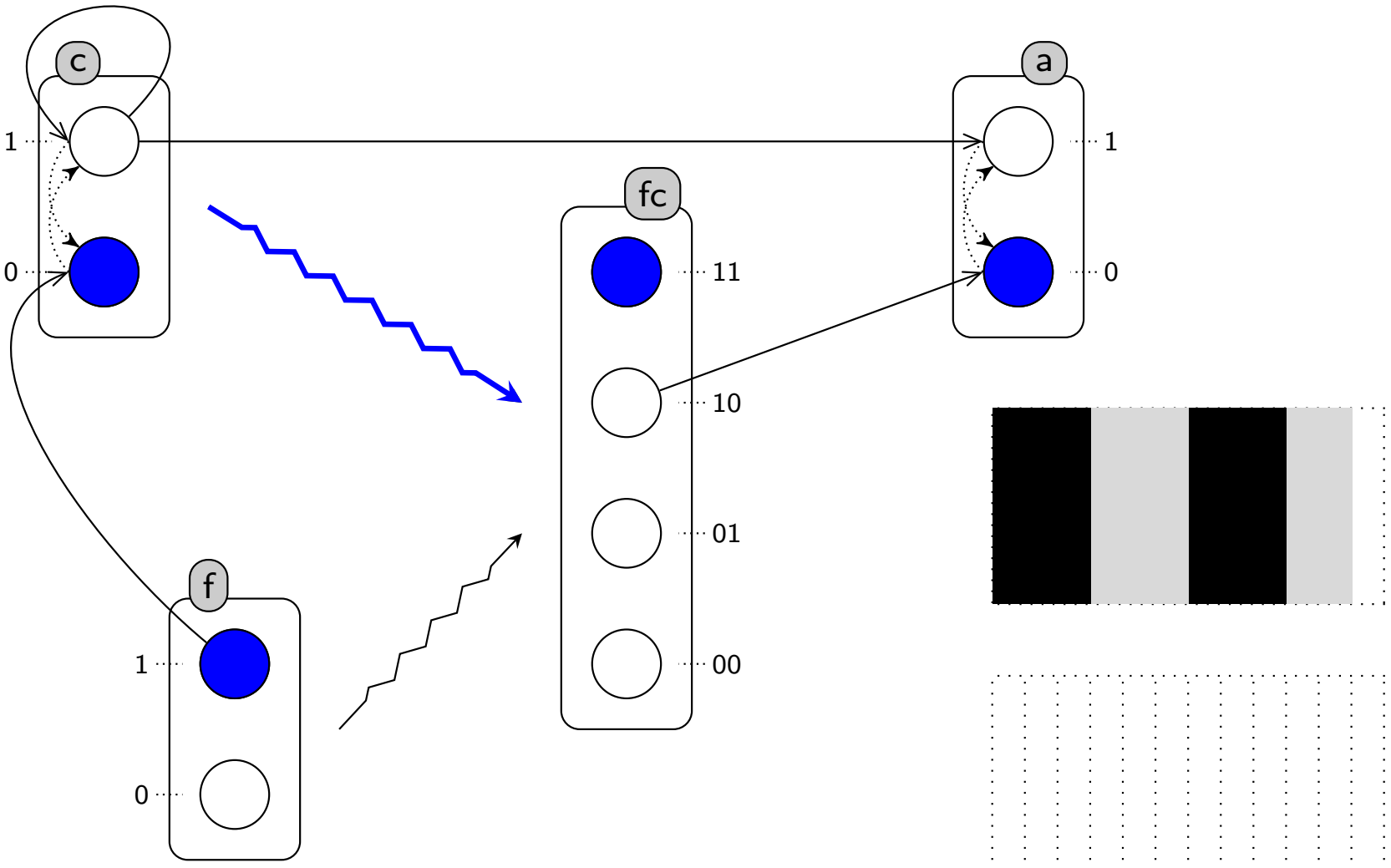
The Metazoan Segmentation Model

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



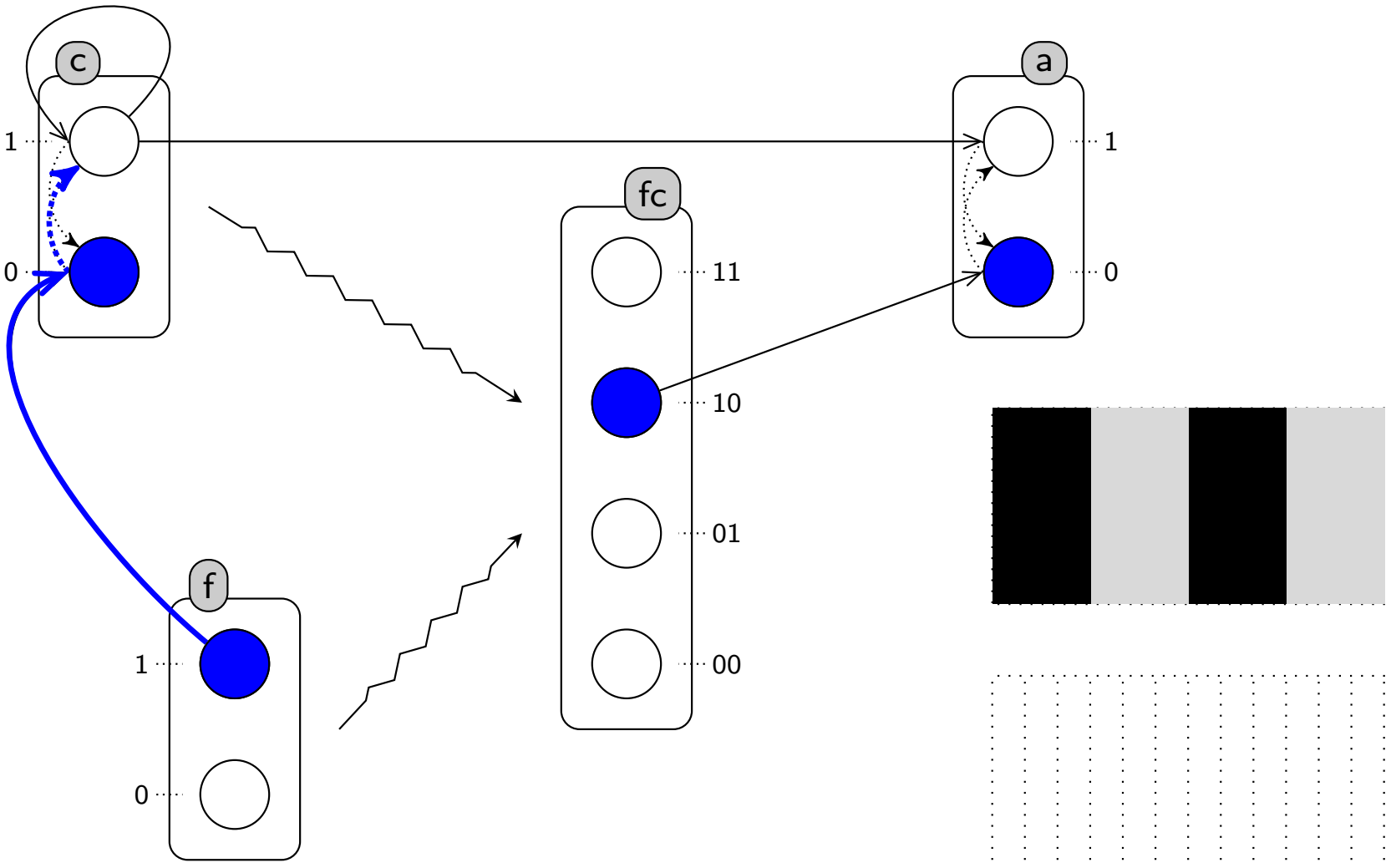
The Metazoan Segmentation Model

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



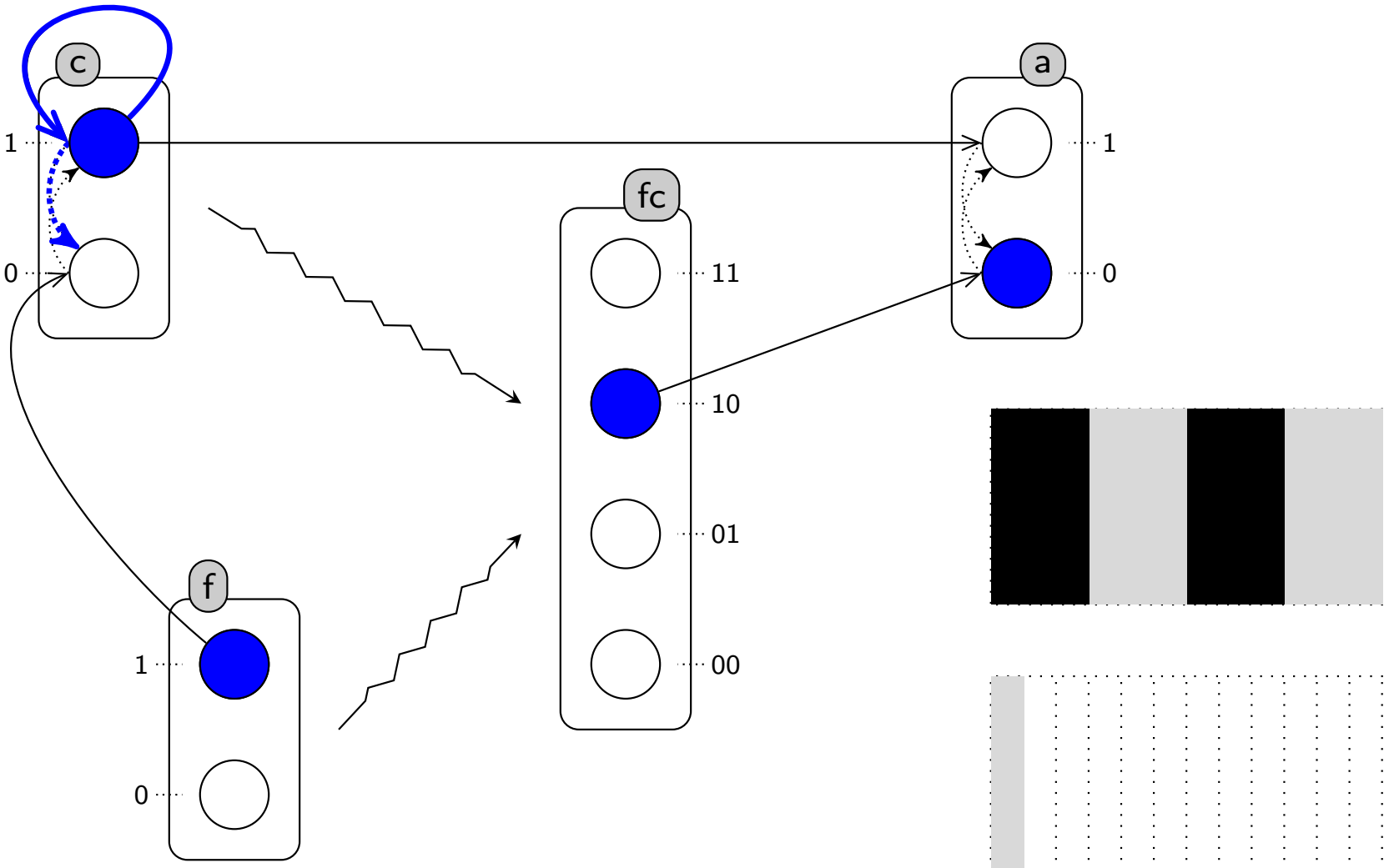
The Metazoan Segmentation Model

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



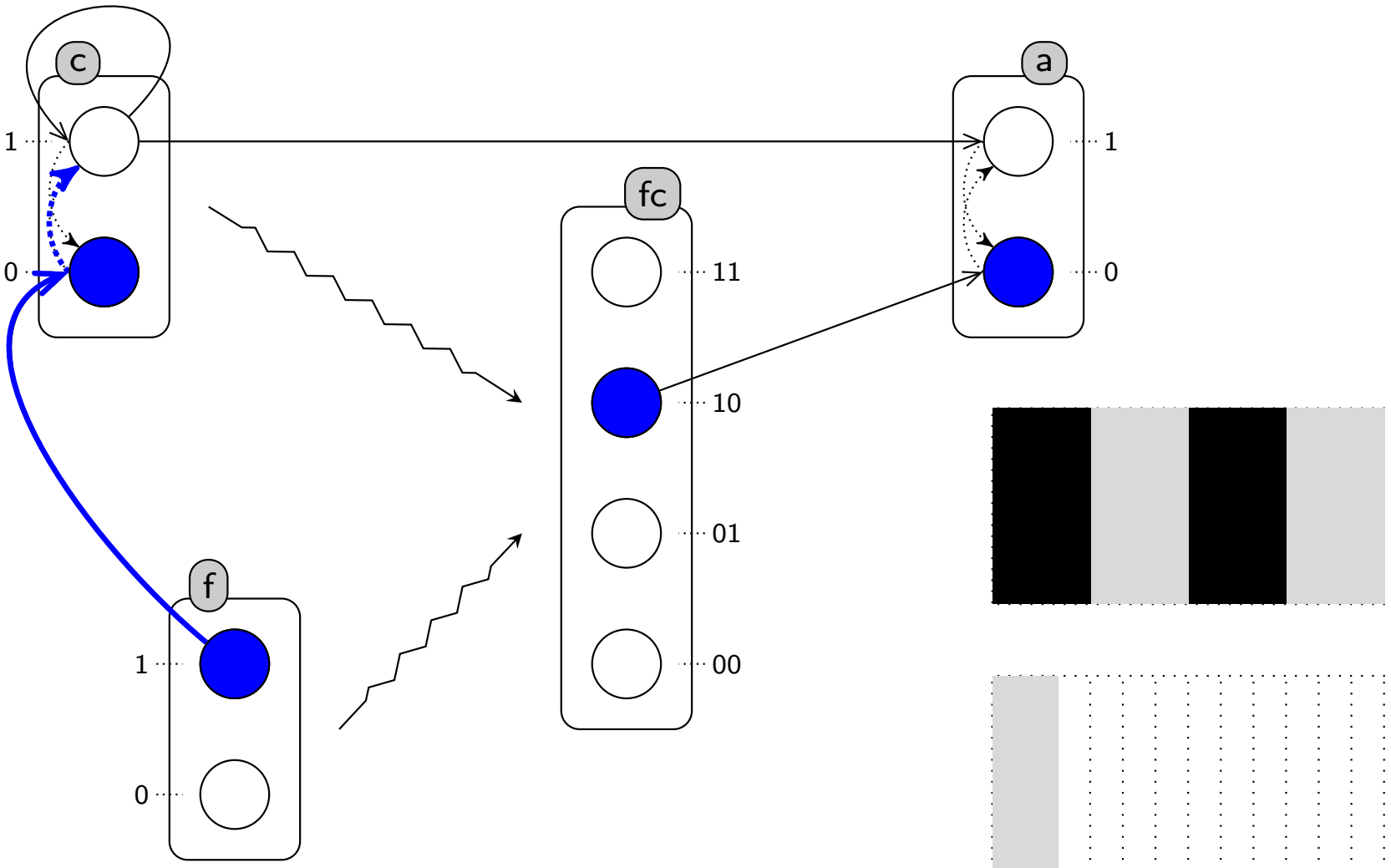
The Metazoan Segmentation Model

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



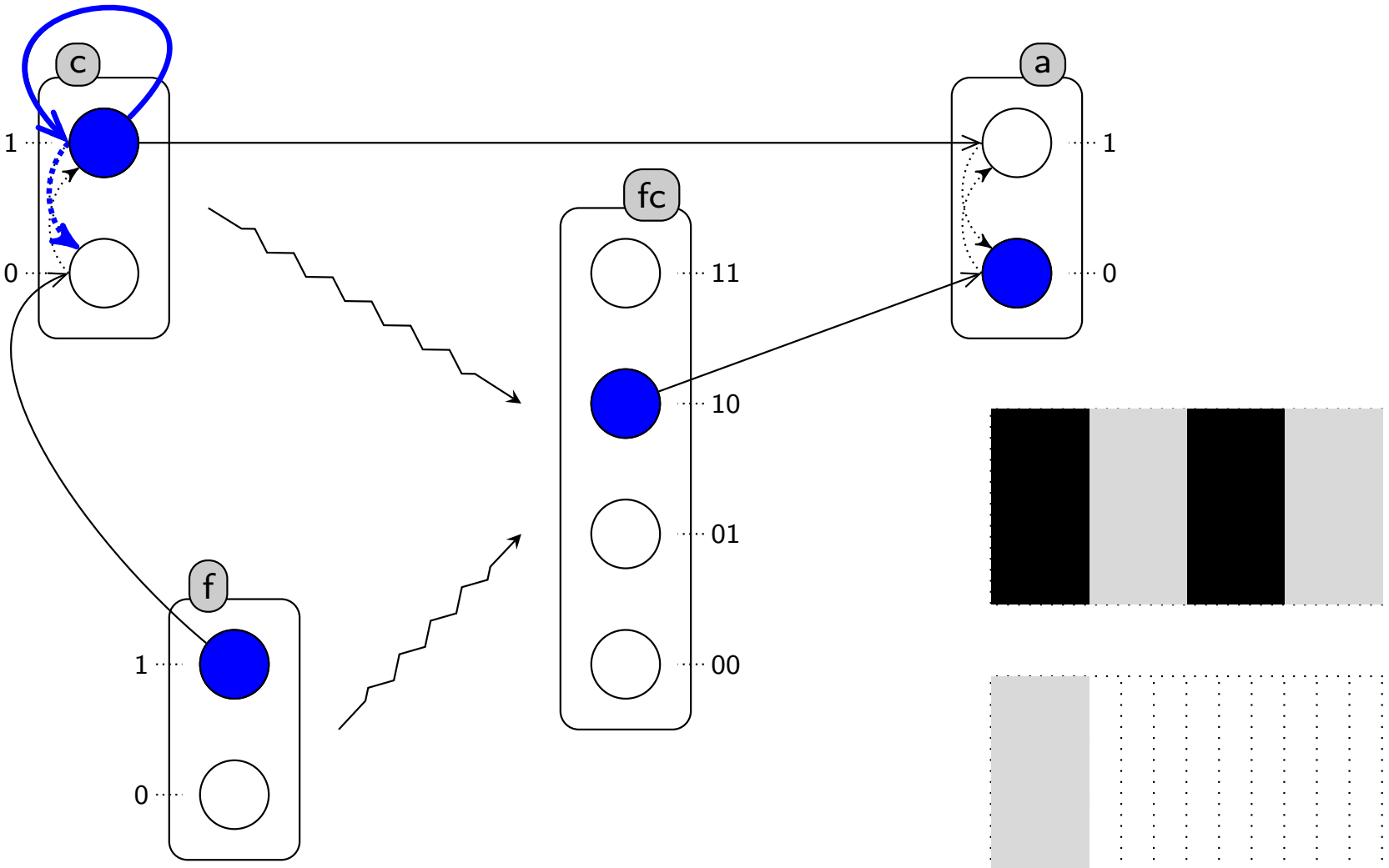
The Metazoan Segmentation Model

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



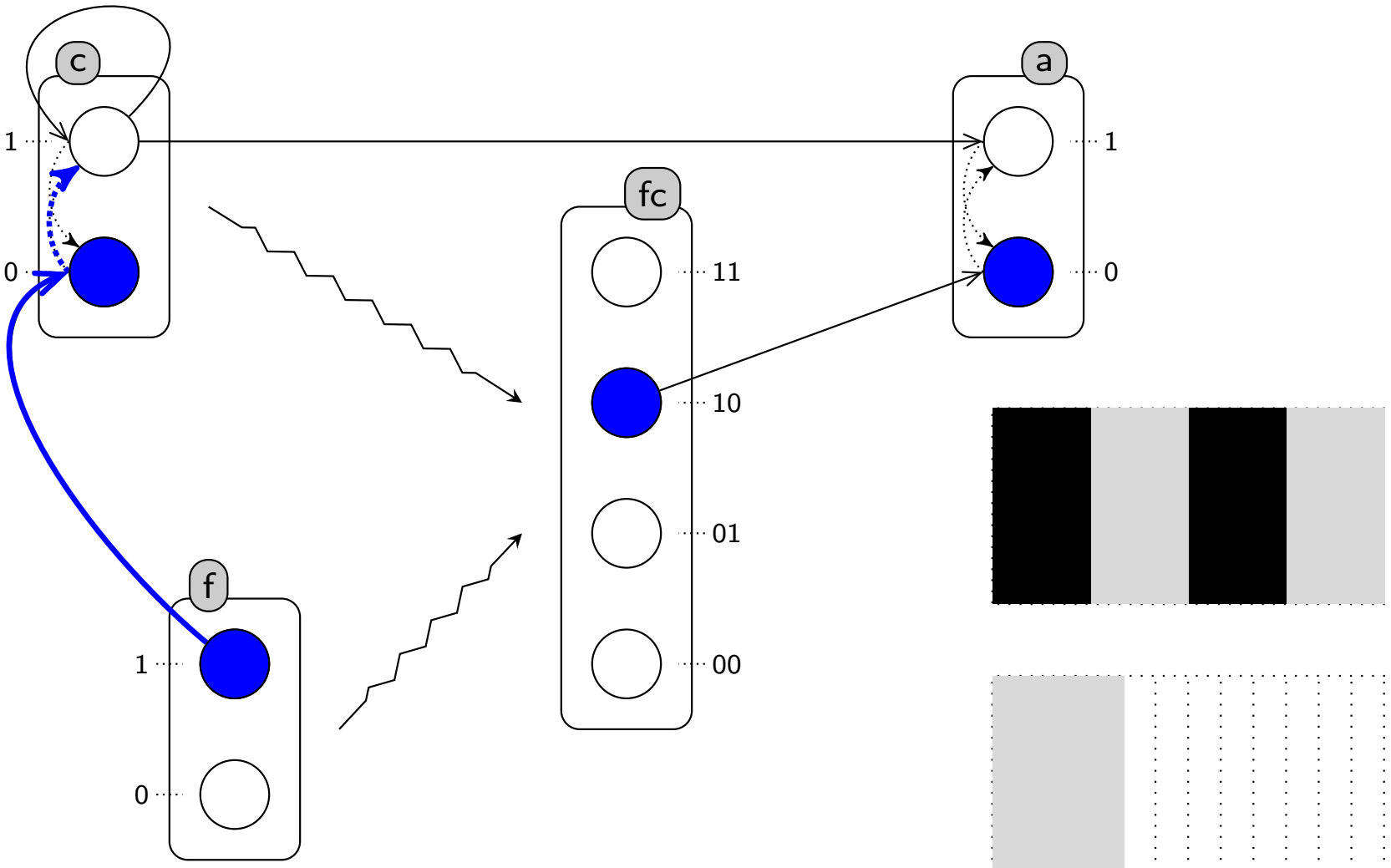
The Metazoan Segmentation Model

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



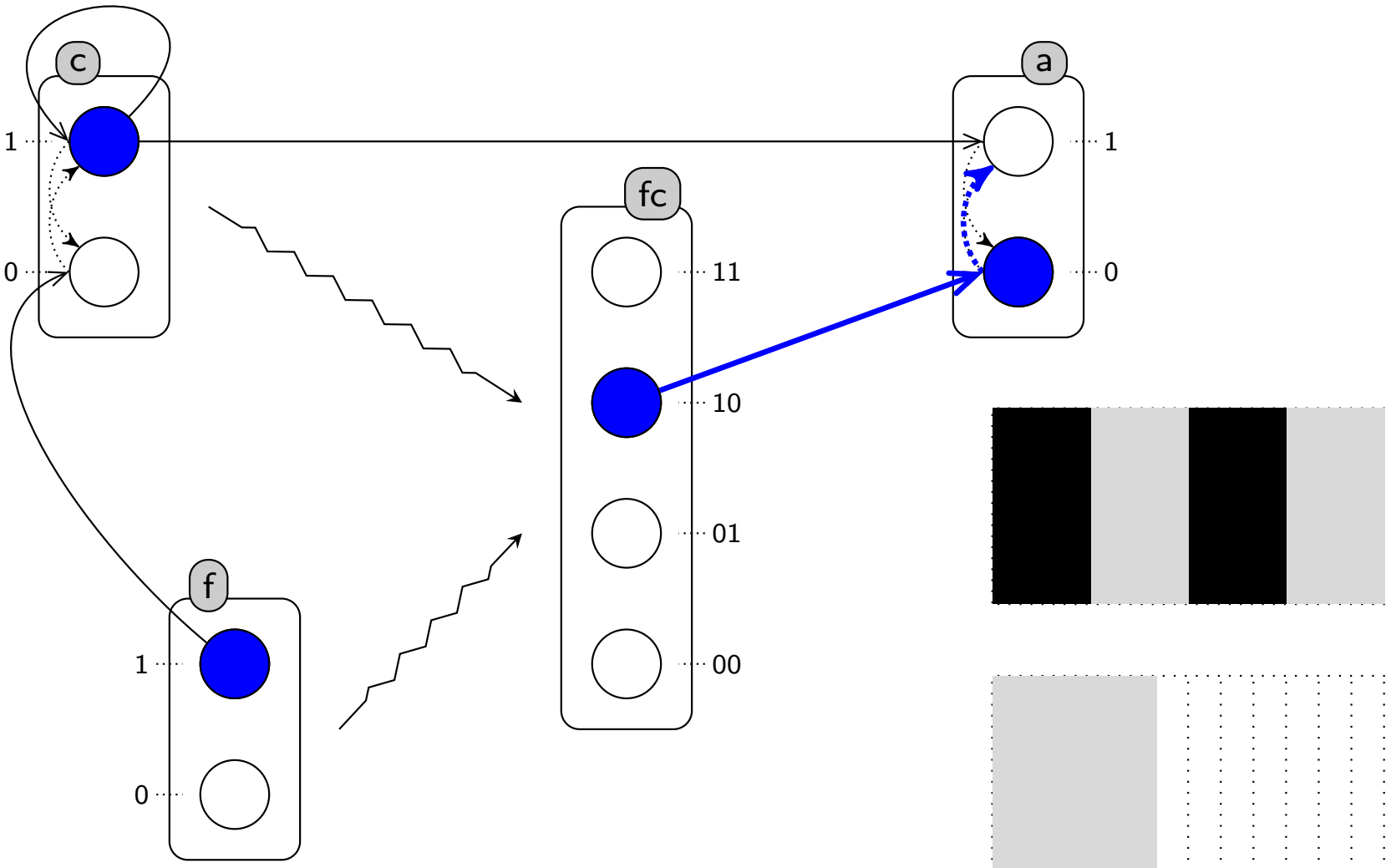
The Metazoan Segmentation Model

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



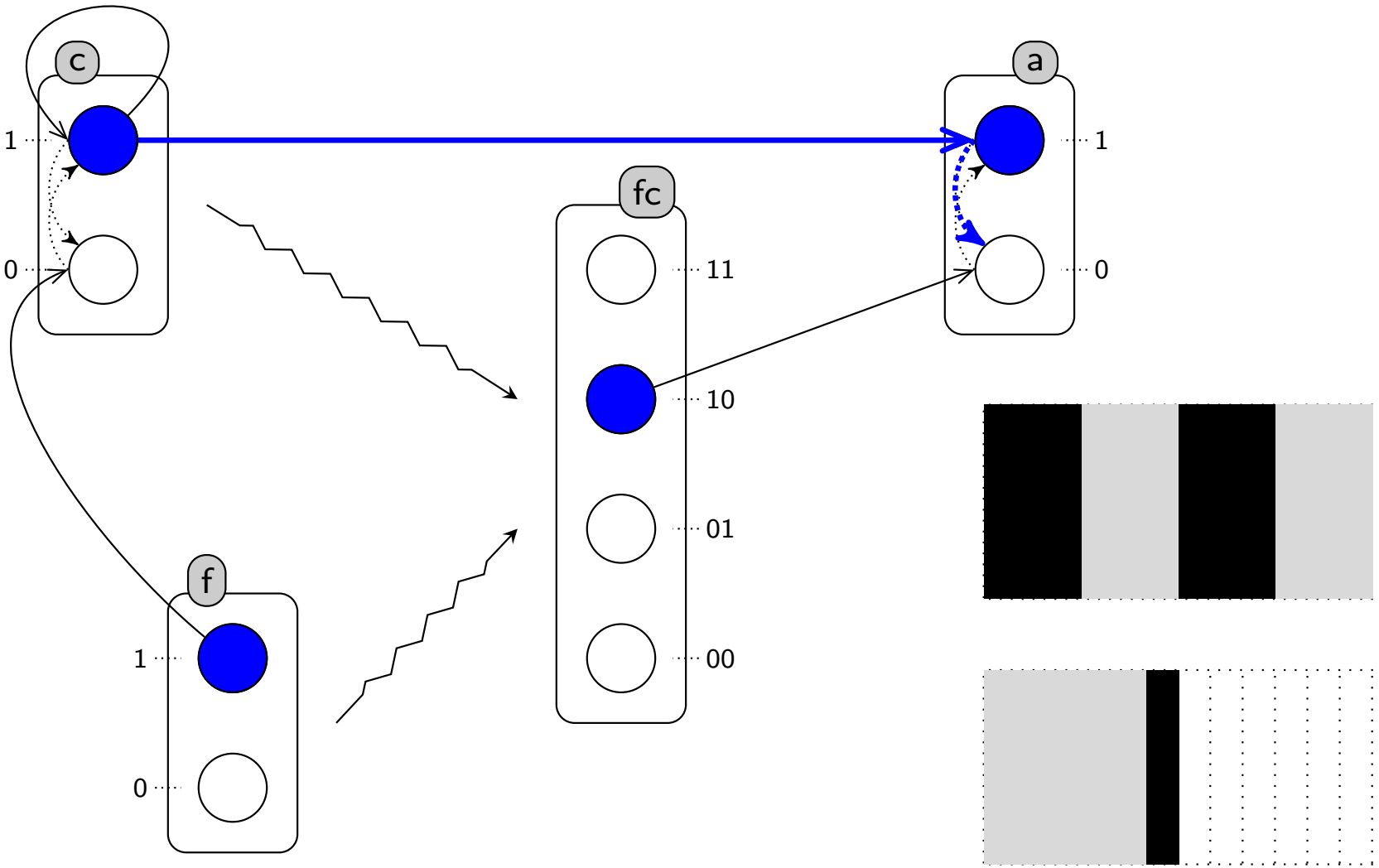
The Metazoan Segmentation Model

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



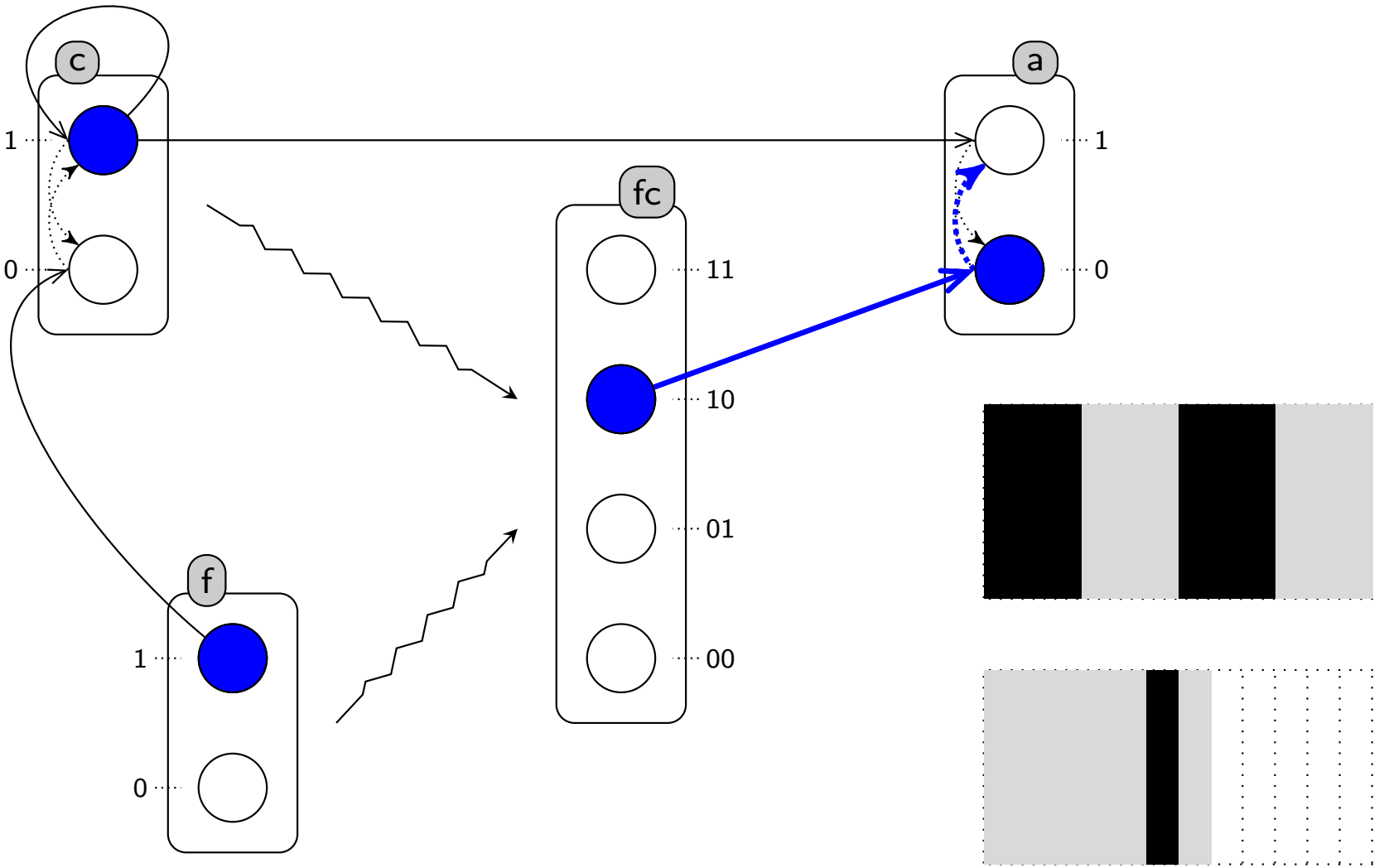
The Metazoan Segmentation Model

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



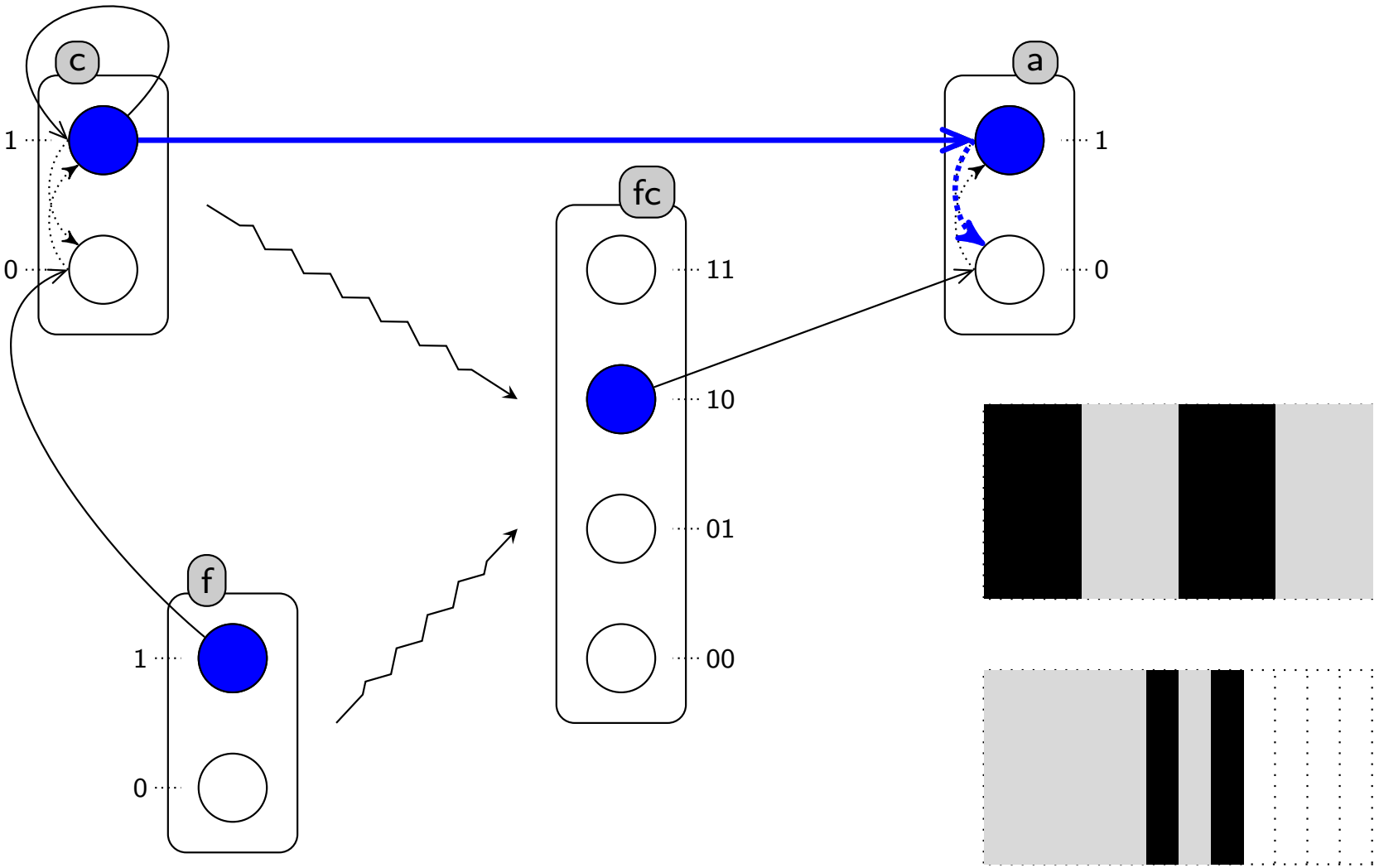
The Metazoan Segmentation Model

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



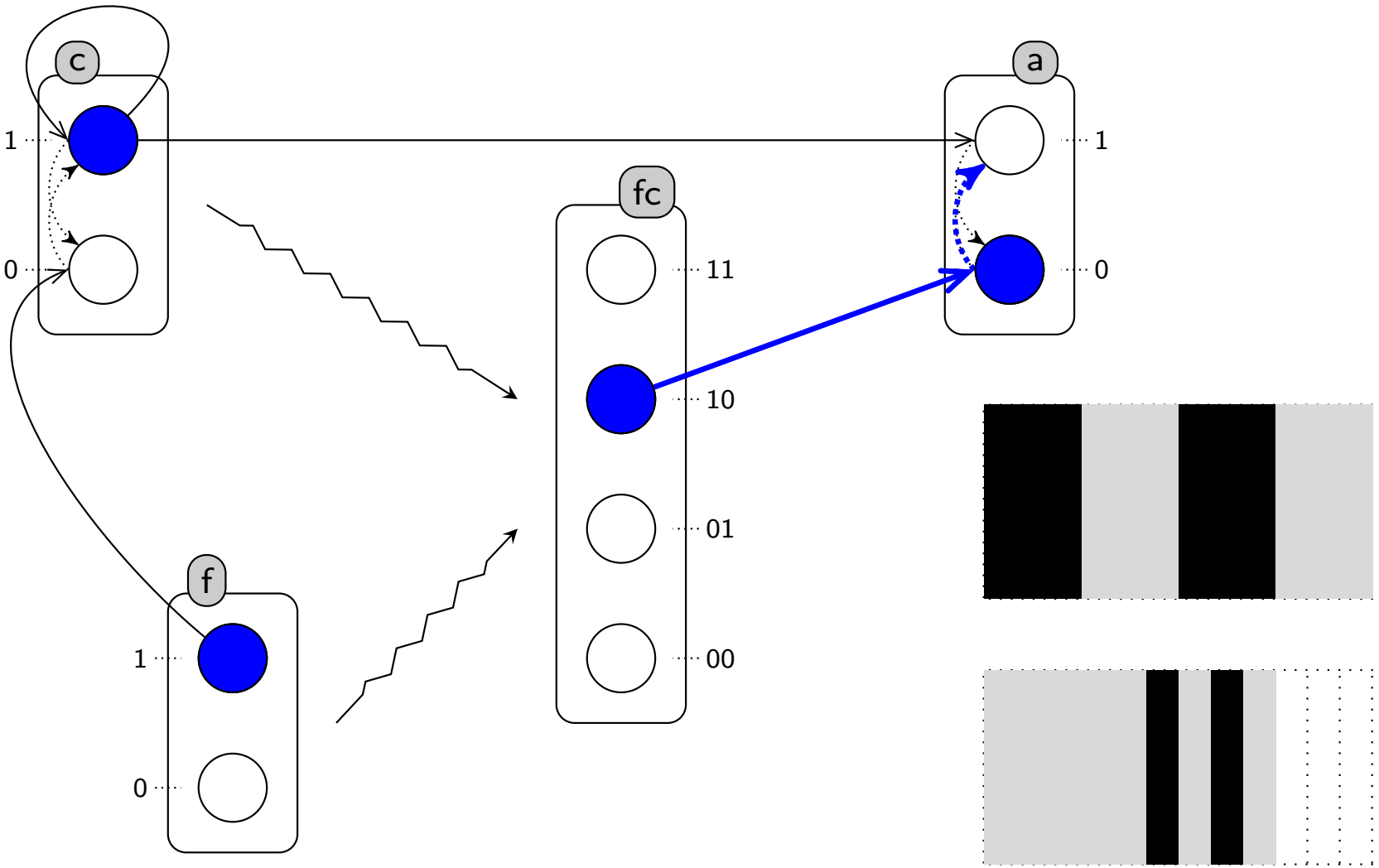
The Metazoan Segmentation Model

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



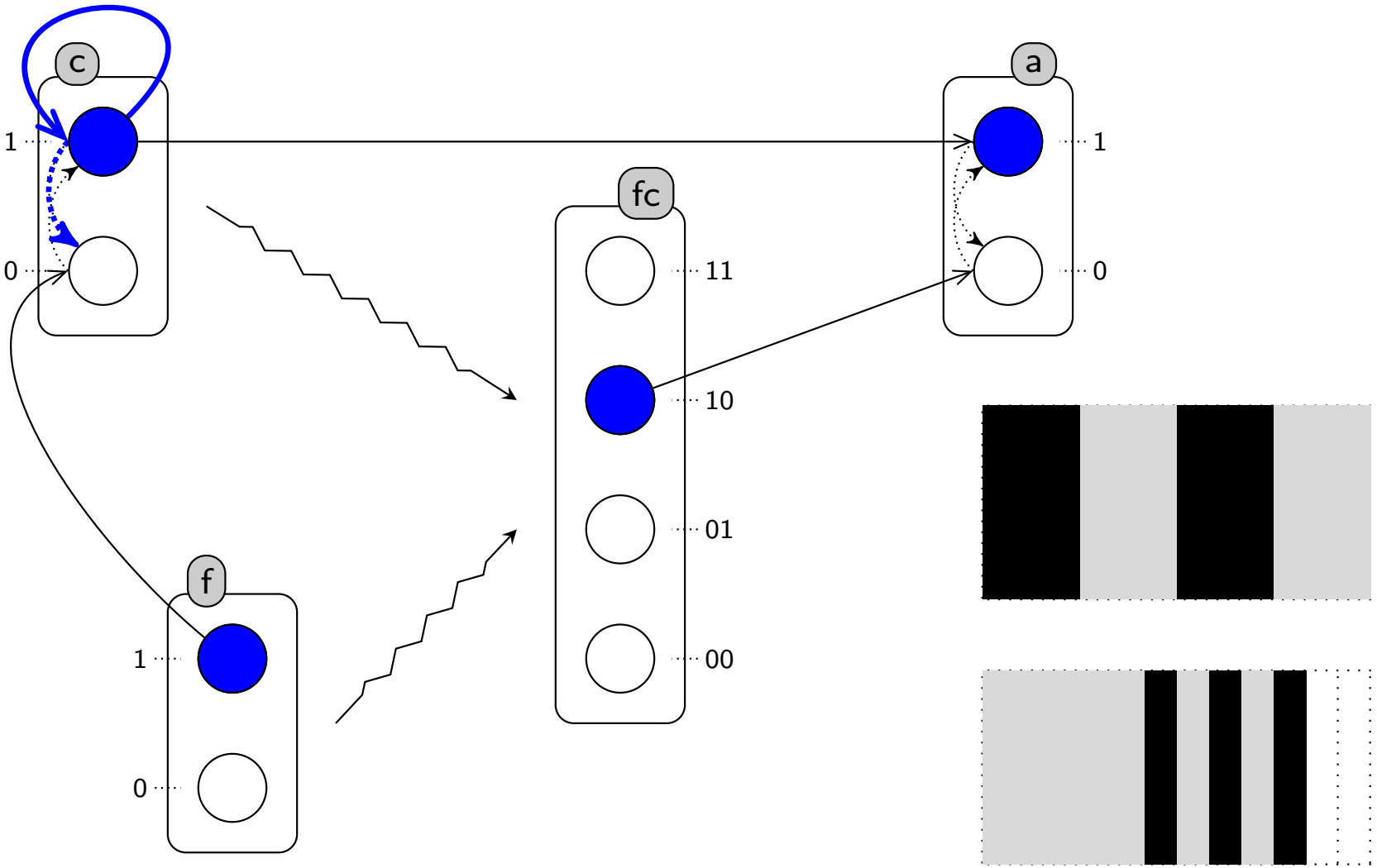
The Metazoan Segmentation Model

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



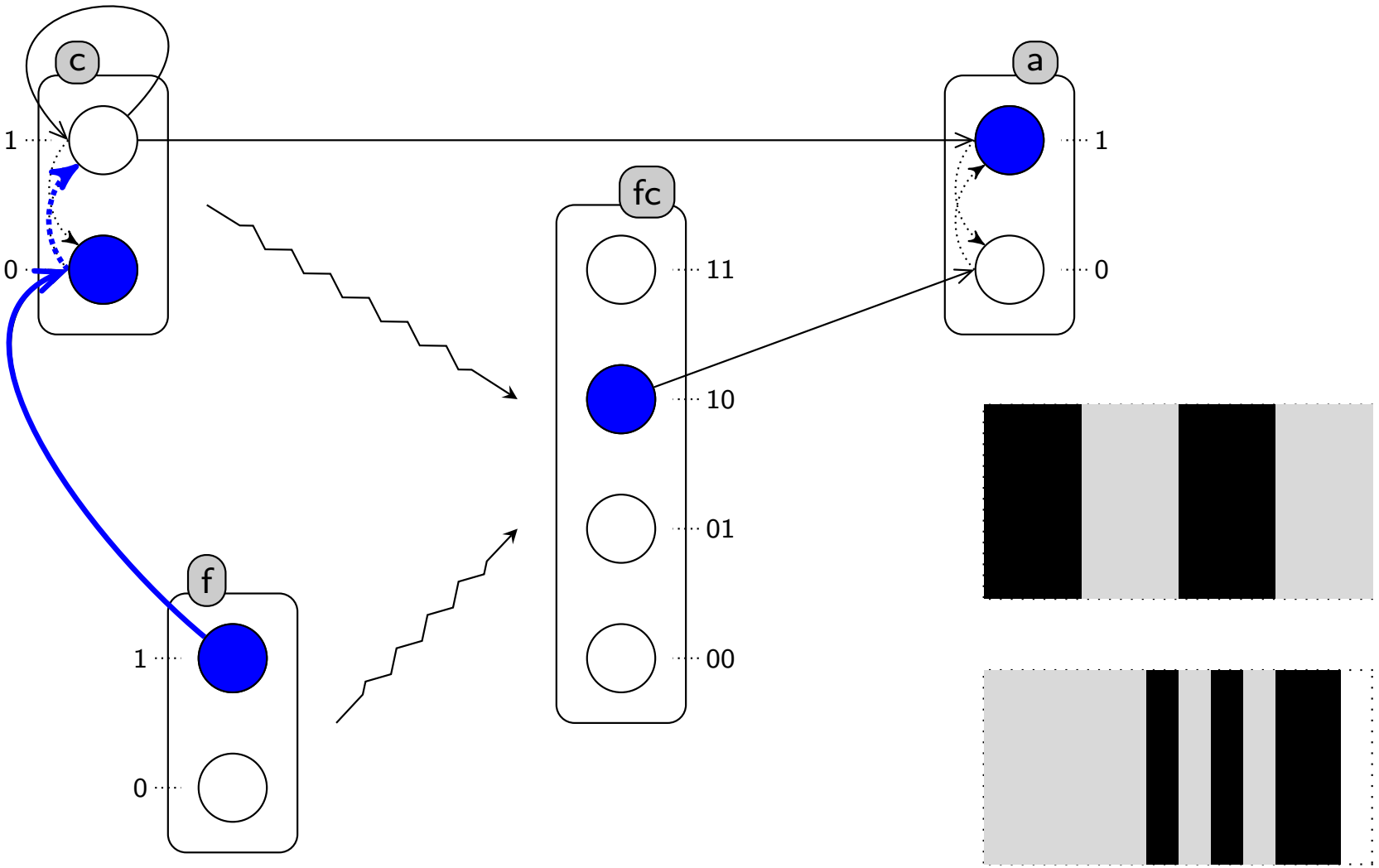
The Metazoan Segmentation Model

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



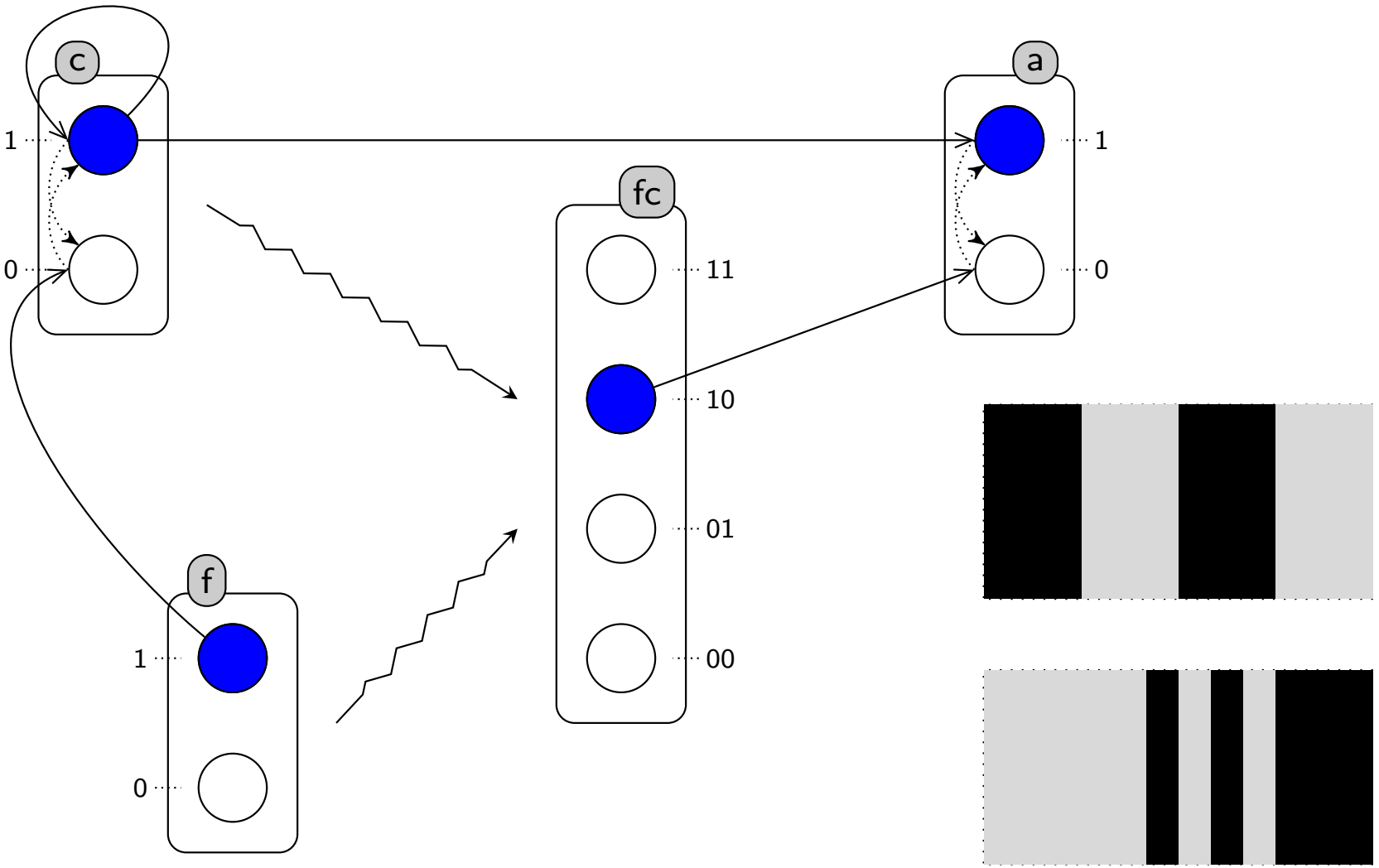
The Metazoan Segmentation Model

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



The Metazoan Segmentation Model

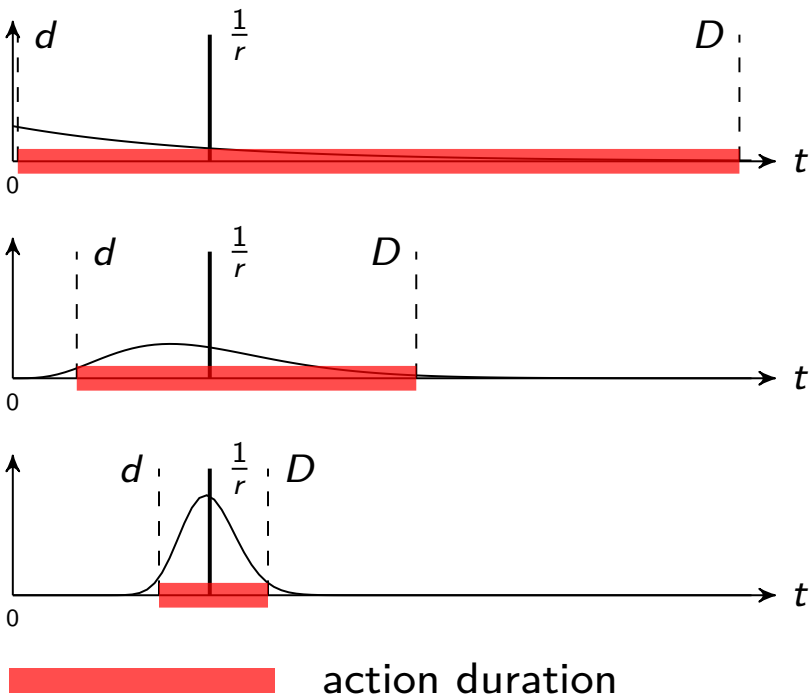
[François *et al.* in Molecular Systems Biology, 2007]



Stochastic Features

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

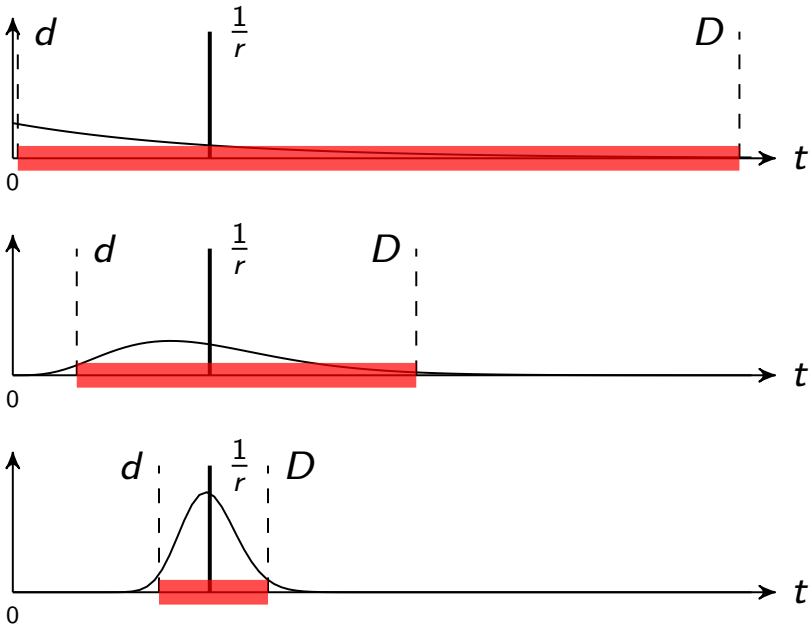
- Introduces time features
- Parameters: either (r, sa), or the **firing interval** $[d; D]$.



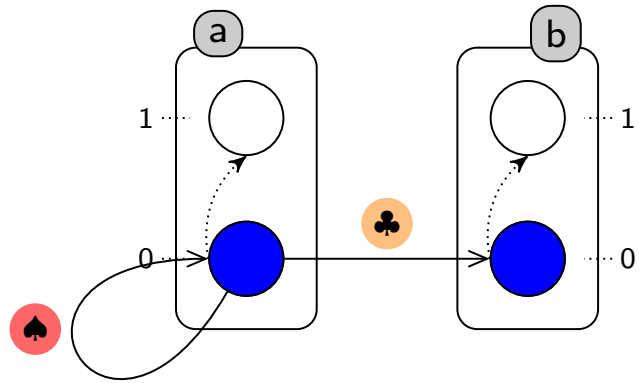
Stochastic Features

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

- Introduces time features
- Parameters: either (r, sa), or the **firing interval** $[d; D]$.

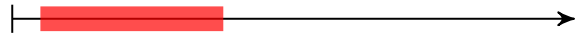


action duration



$a_0 \rightarrow b_0 \Rsh b_1$ (♣)

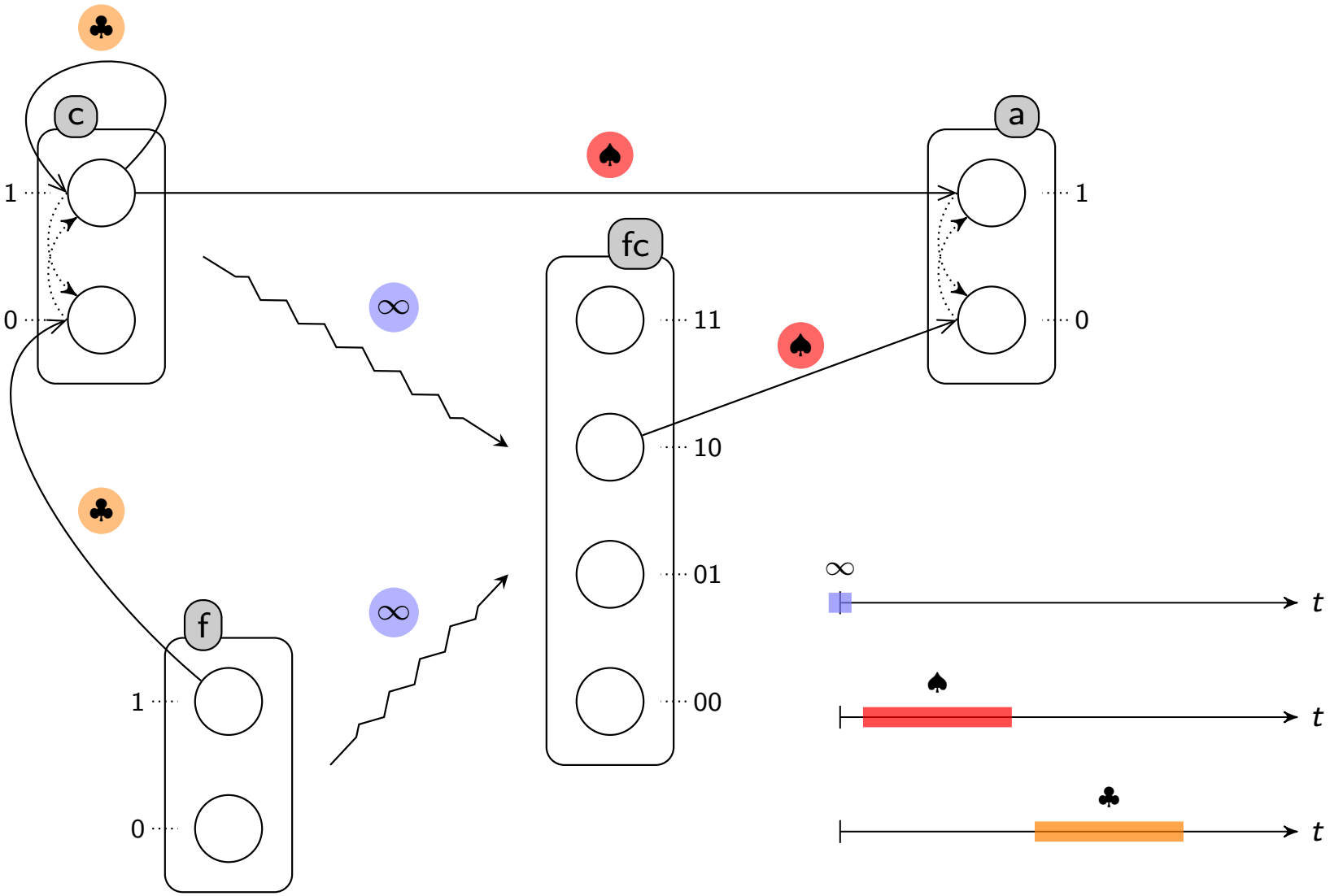
$a_0 \rightarrow a_0 \Rsh a_1$ (♠)



$\rightarrow$ b_1 reached with a **very low probability**.

Metazoan Segmentation with Stochastic Parameters

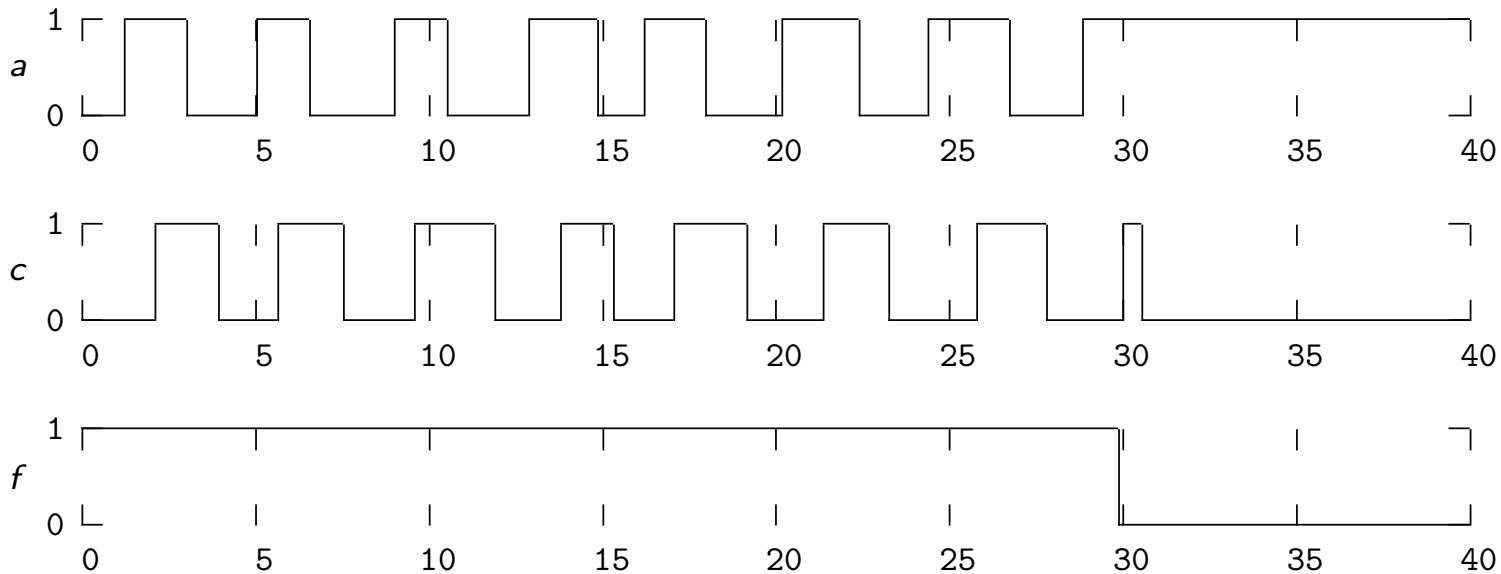
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Pros and Cons of Stochastic Parameters

Pros:

- Introduction of temporal features
- Simulation in continuous time

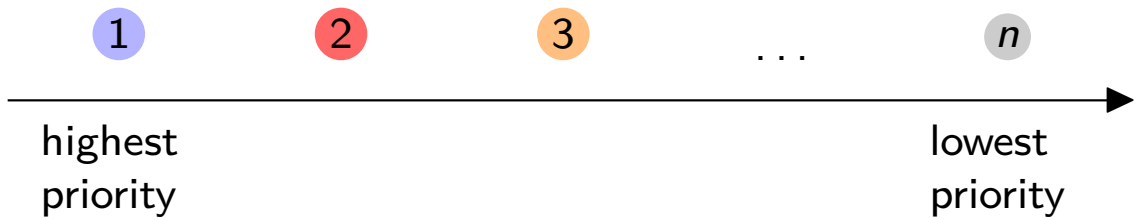
Cons:

- Very hard to analyze
  - Either multiple runs (statistics)
  - Or model checkers (PRISM) but combinatorial explosion

Introduction of Classes of Priorities

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

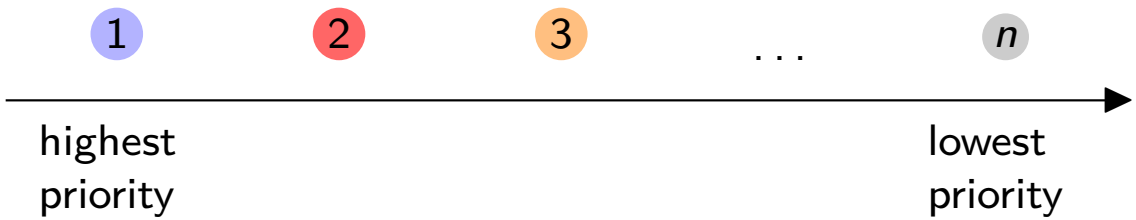
- Each action is associated to a class of priority.
- An action cannot be played if another action of higher priority is playable.

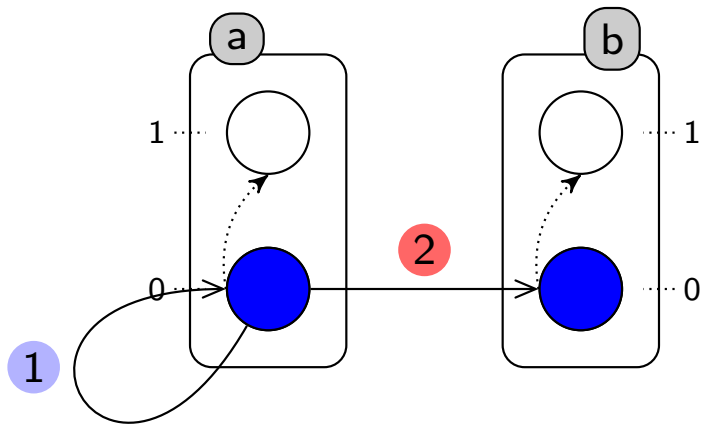


Introduction of Classes of Priorities

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

- Each action is associated to a class of priority.
- An action cannot be played if another action of higher priority is playable.



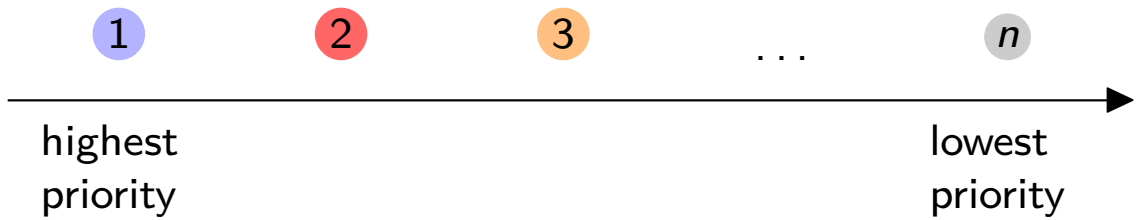


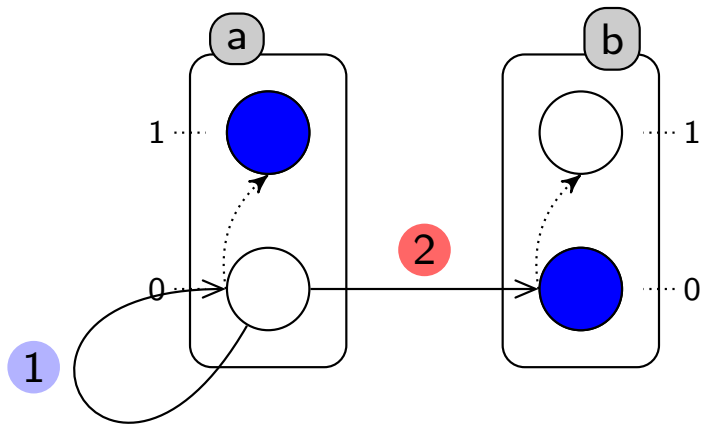
$\rightarrow$ b_1 **cannot be reached**

Introduction of Classes of Priorities

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

- Each action is associated to a class of priority.
- An action cannot be played if another action of higher priority is playable.

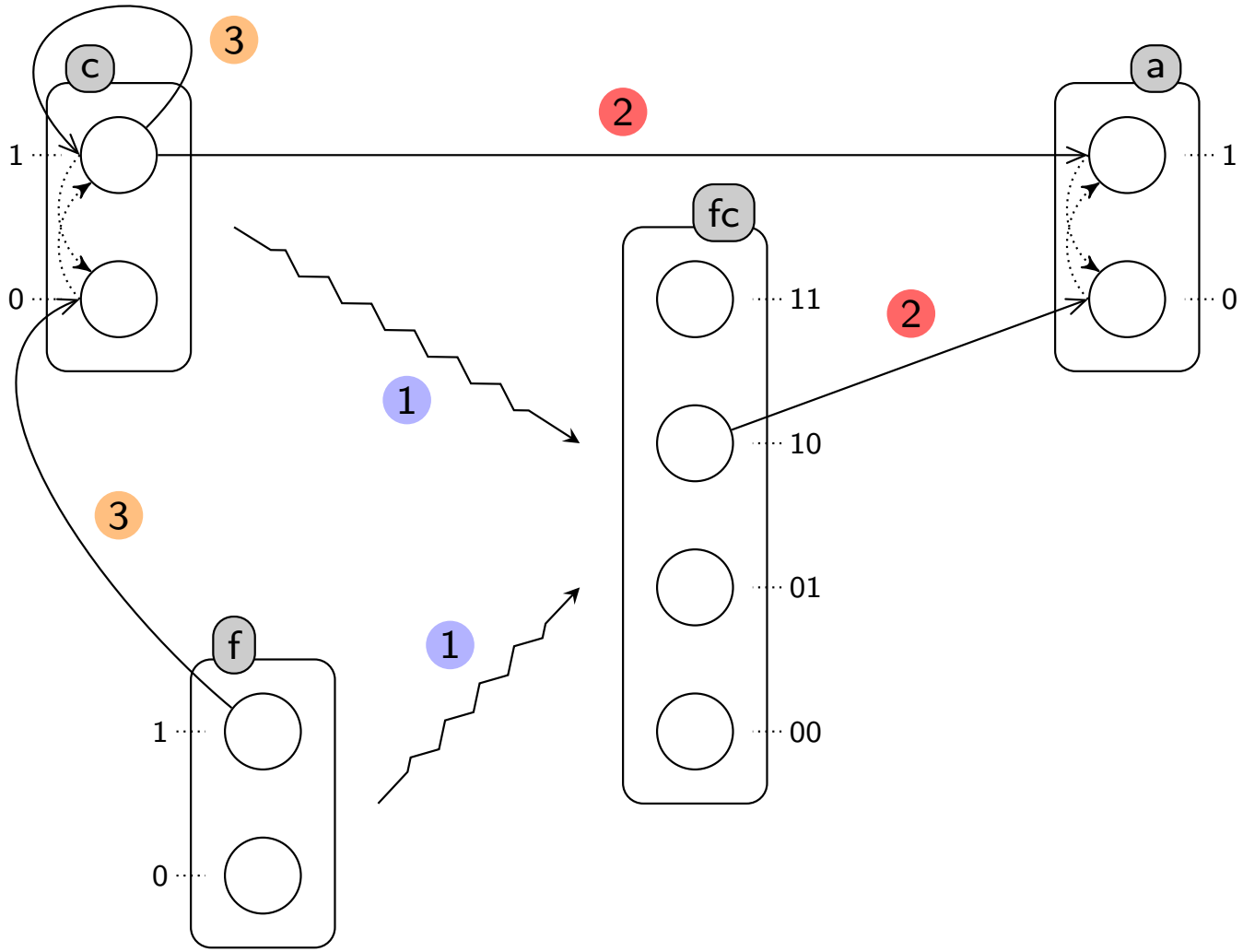




$\rightarrow$ b_1 **cannot be reached**

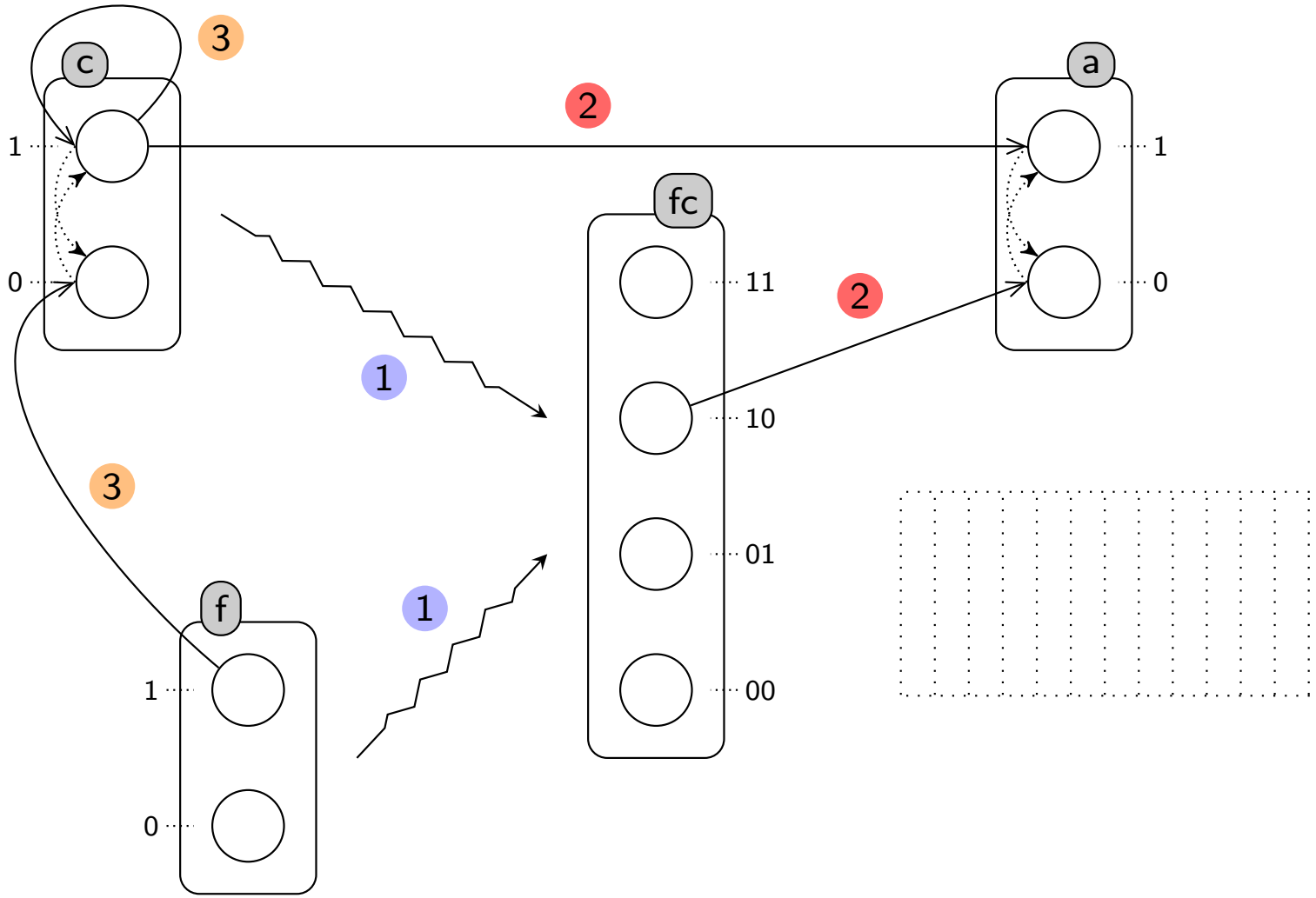
Metazoan Segmentation with Priorities

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



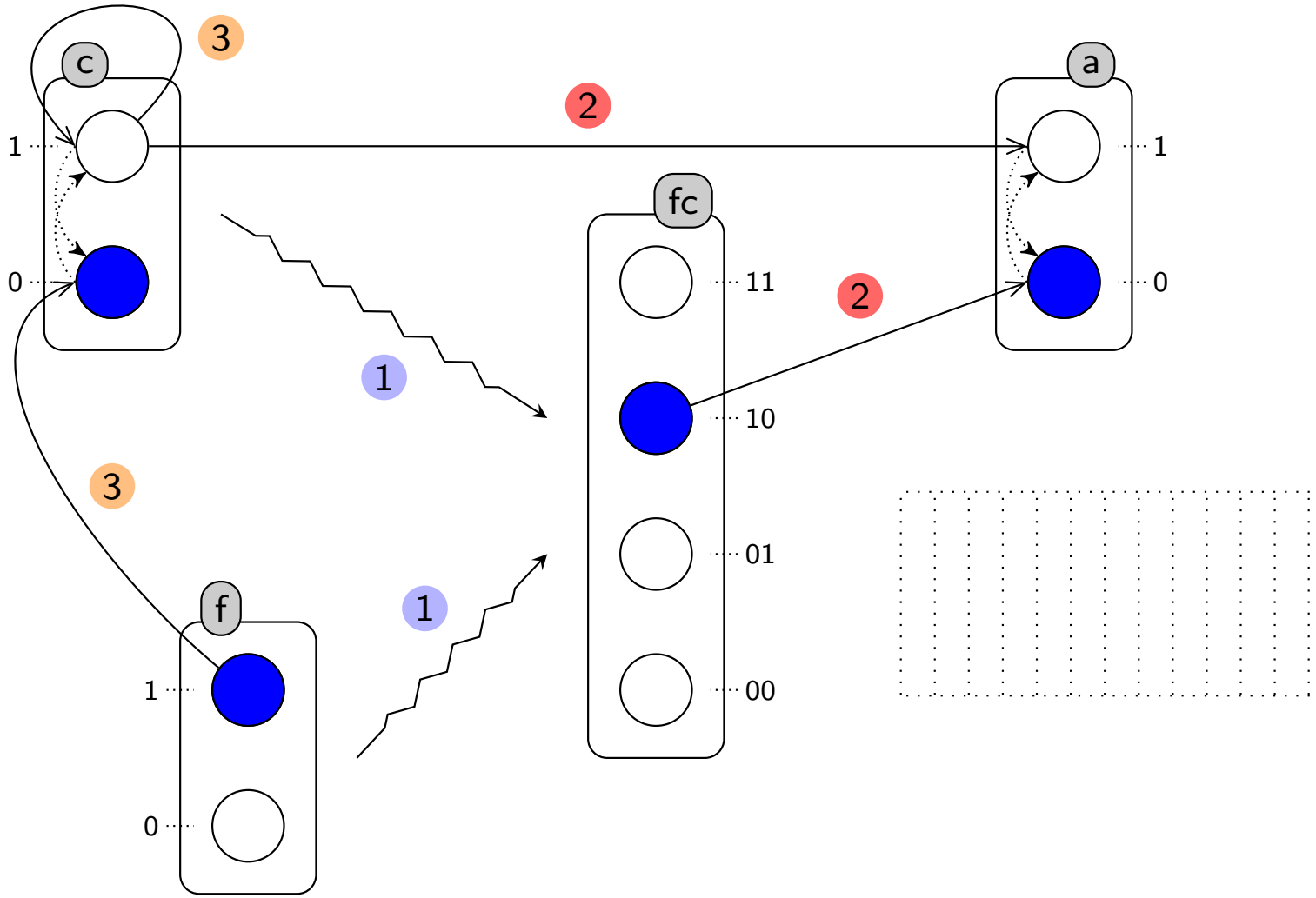
Metazoan Segmentation with Priorities

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



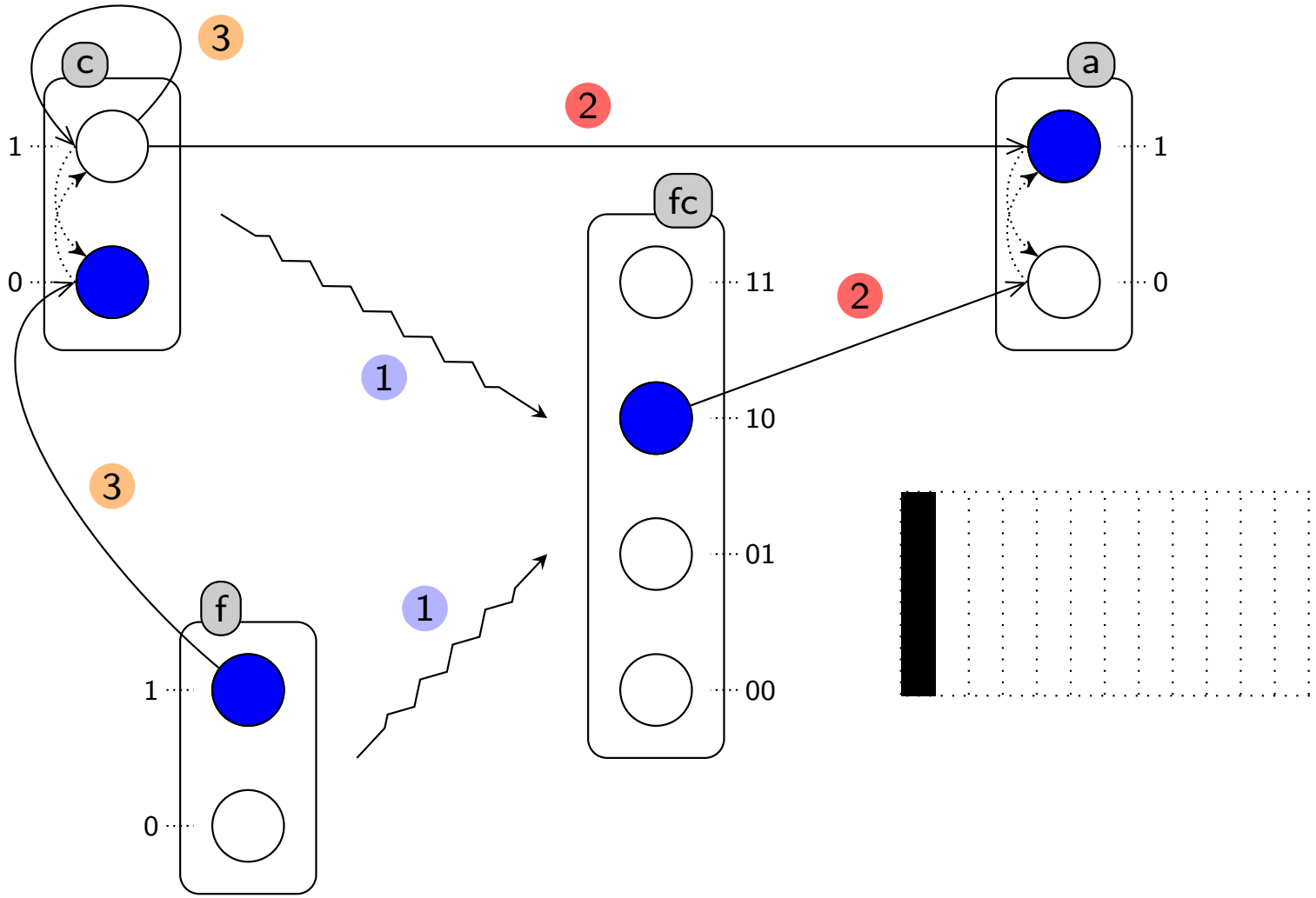
Metazoan Segmentation with Priorities

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



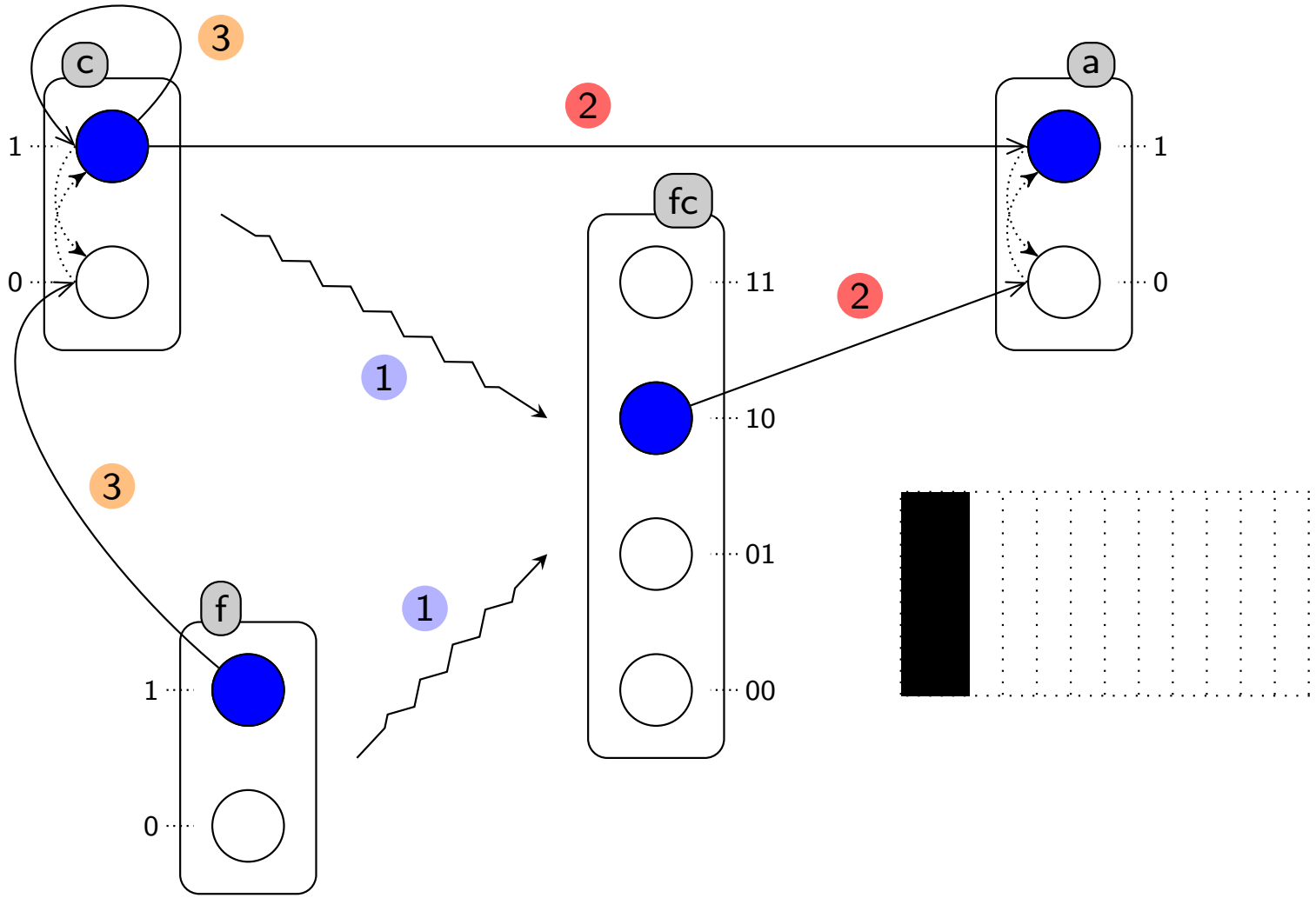
Metazoan Segmentation with Priorities

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



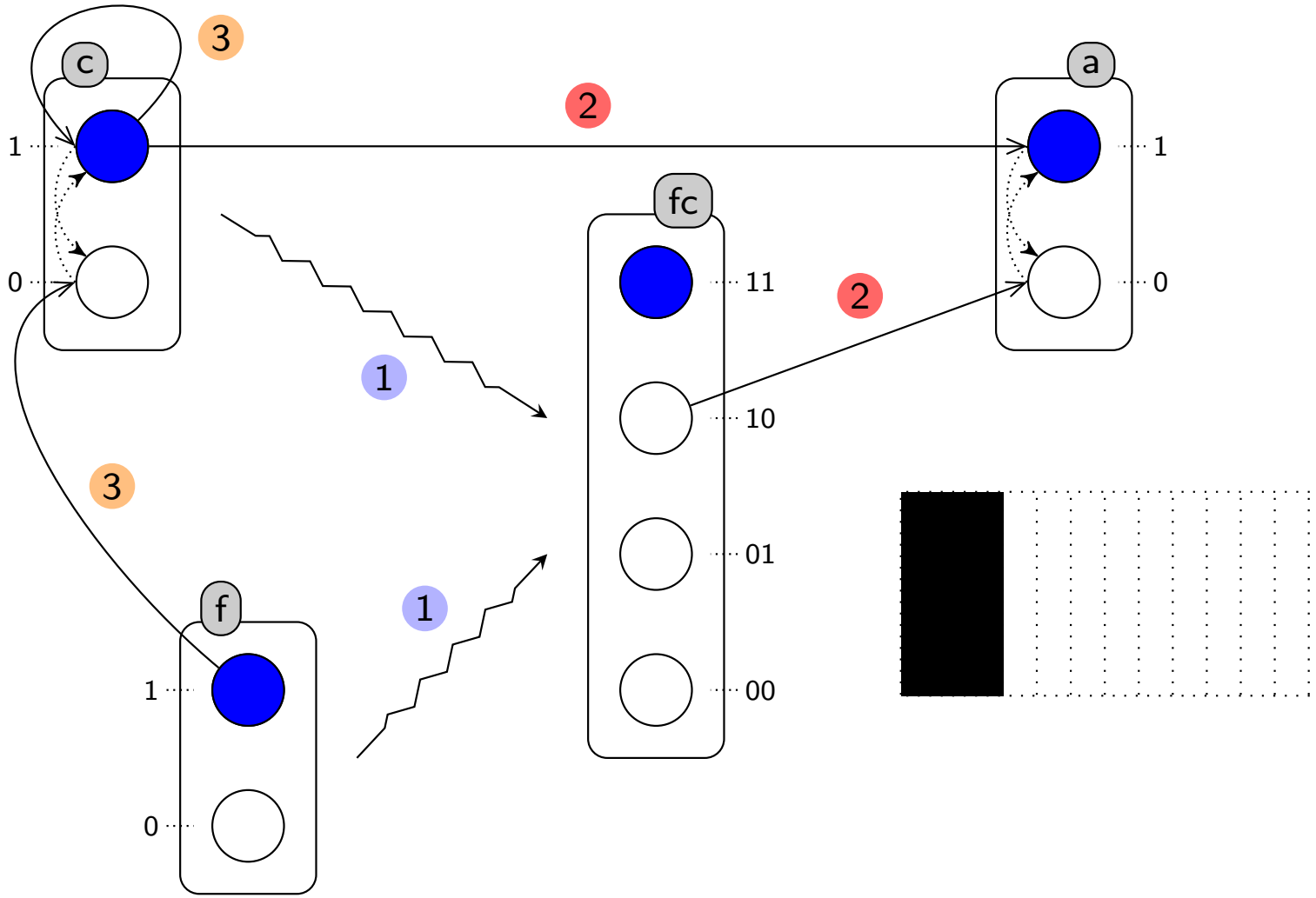
Metazoan Segmentation with Priorities

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



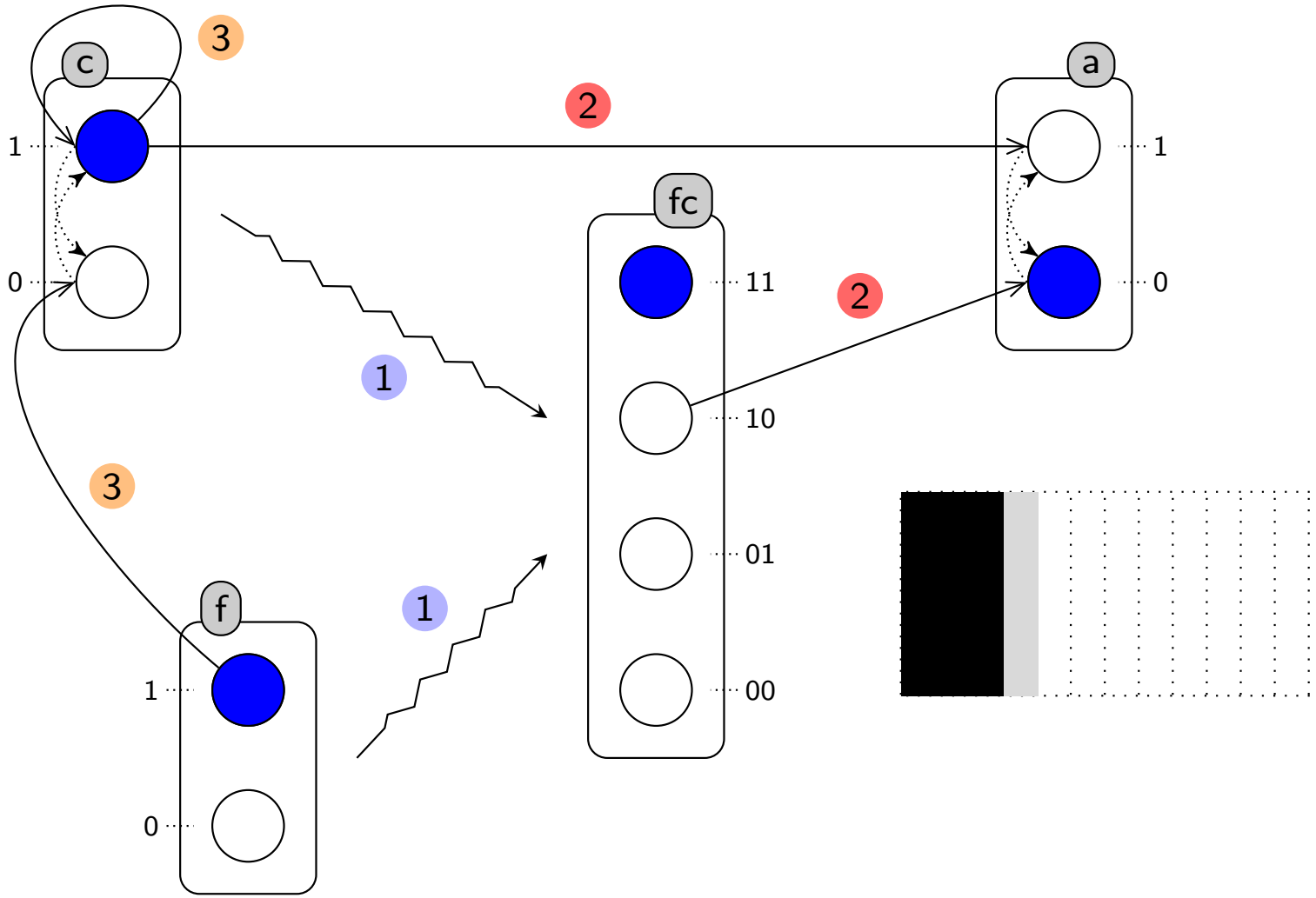
Metazoan Segmentation with Priorities

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



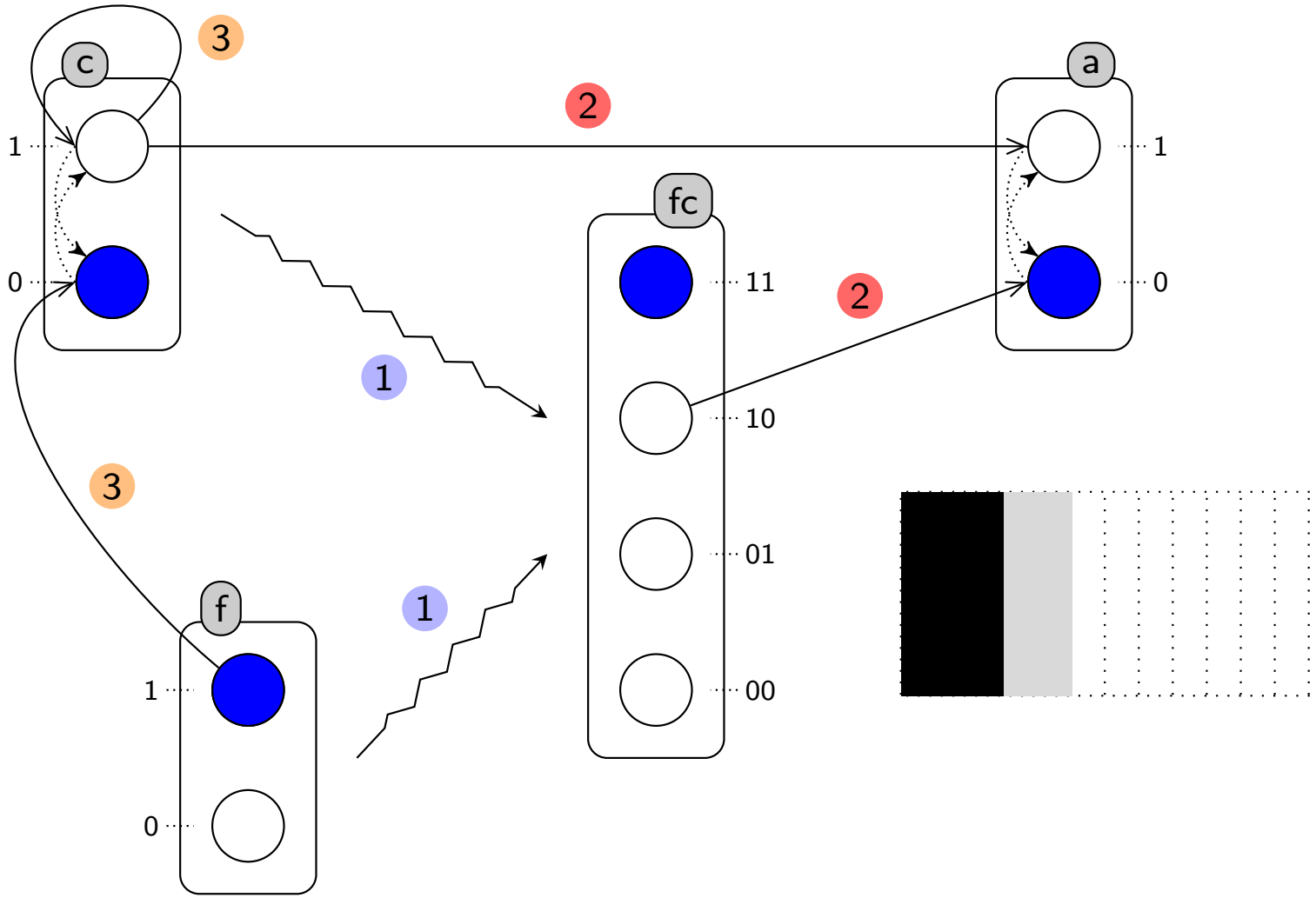
Metazoan Segmentation with Priorities

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



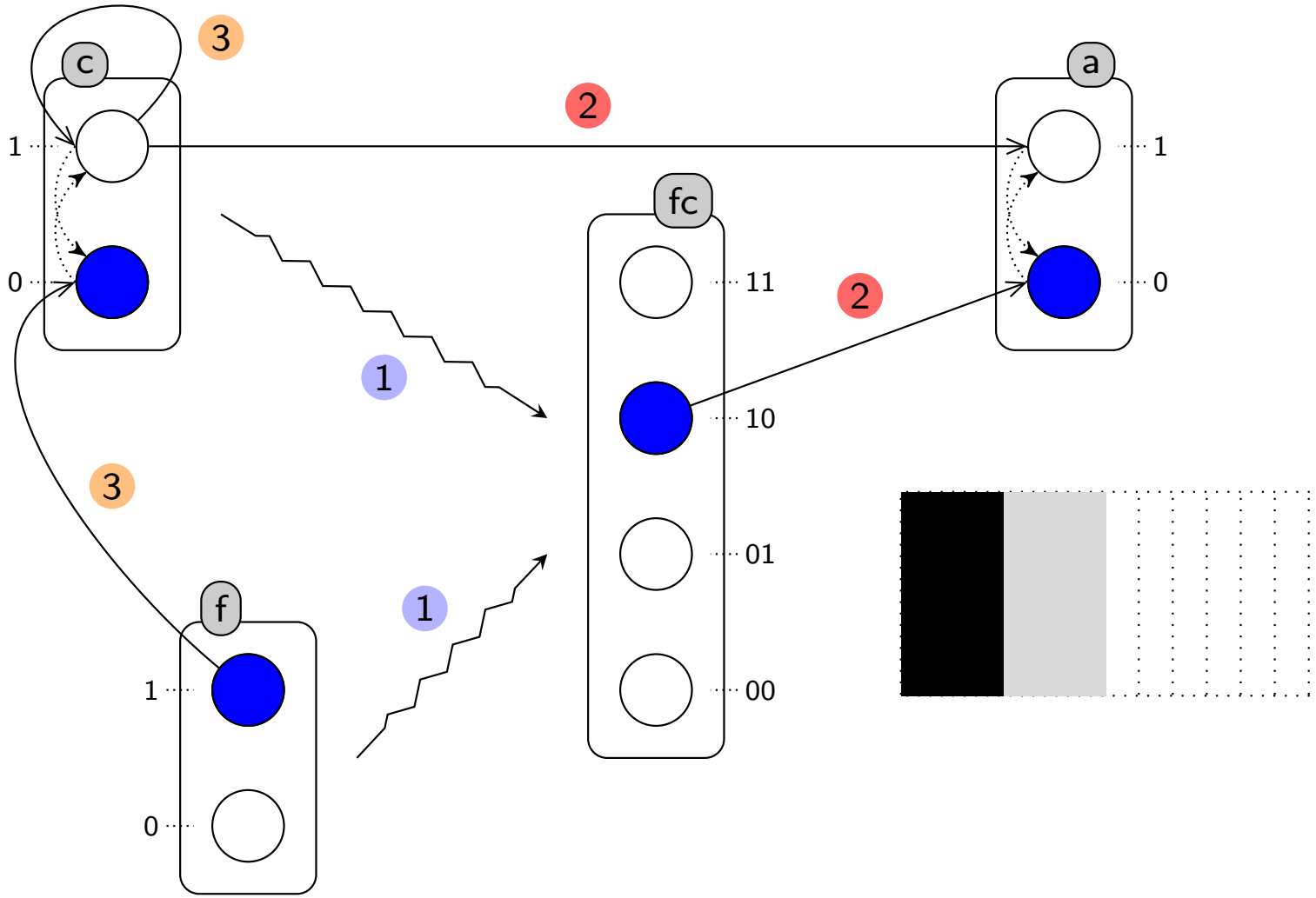
Metazoan Segmentation with Priorities

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



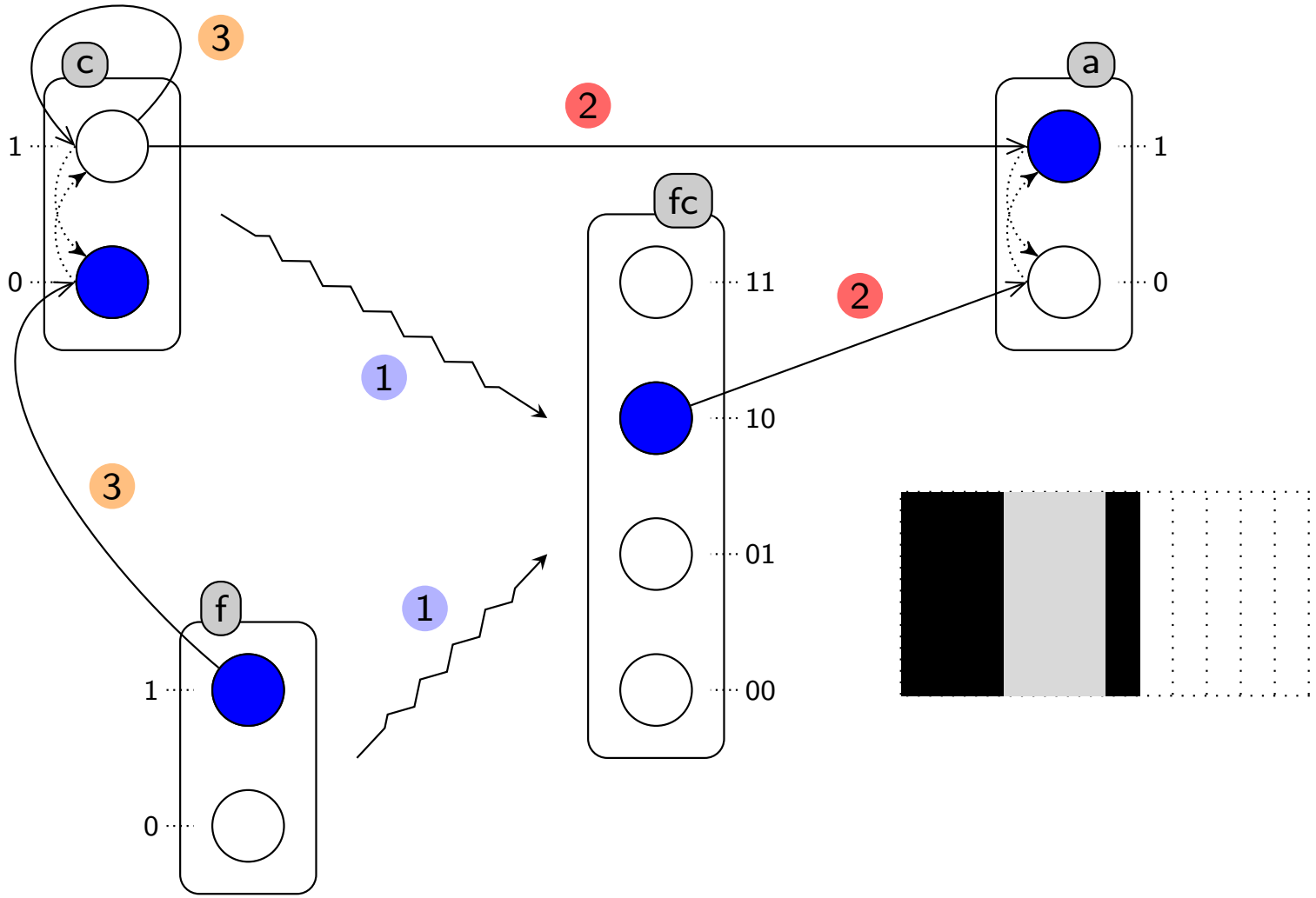
Metazoan Segmentation with Priorities

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



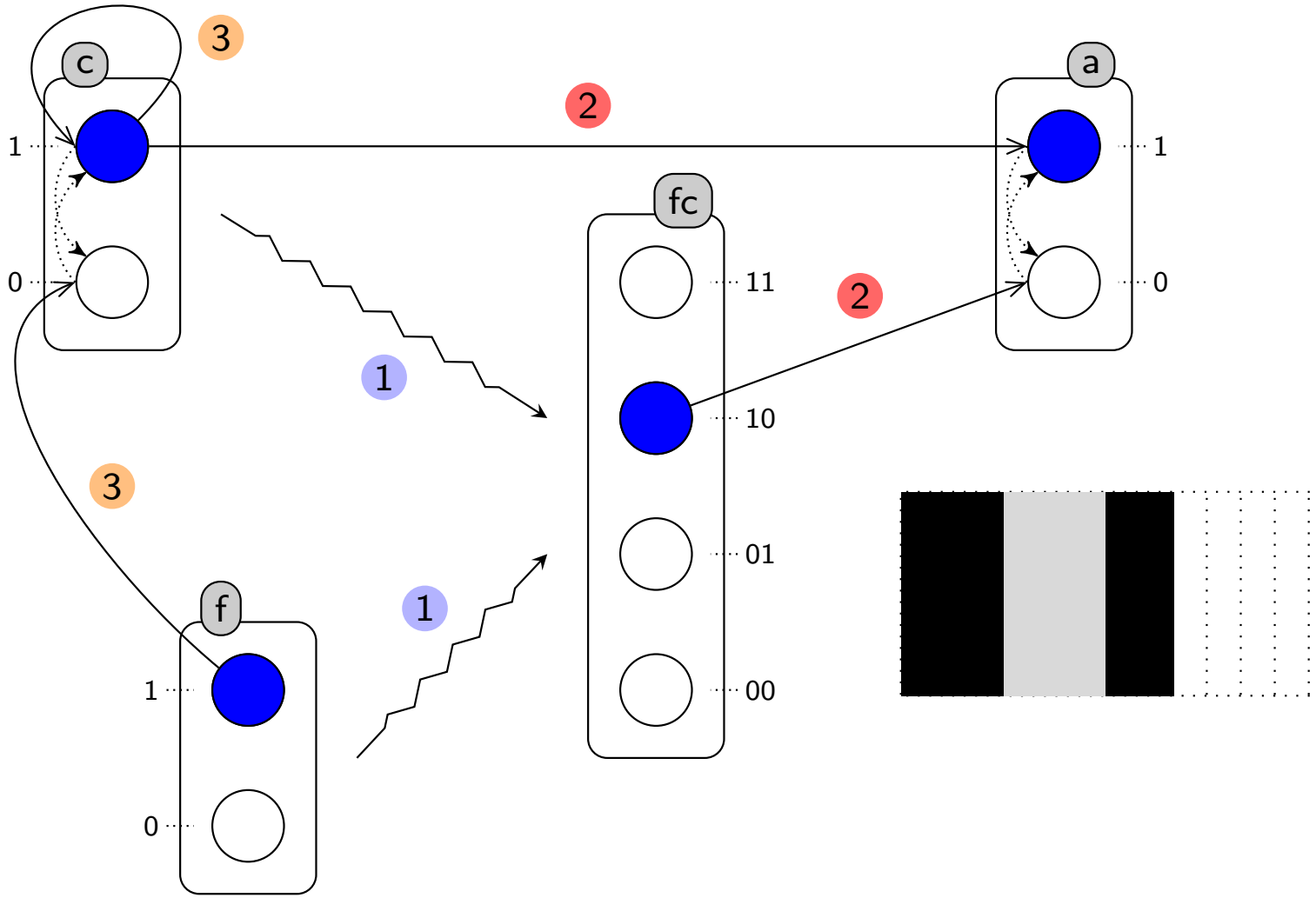
Metazoan Segmentation with Priorities

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



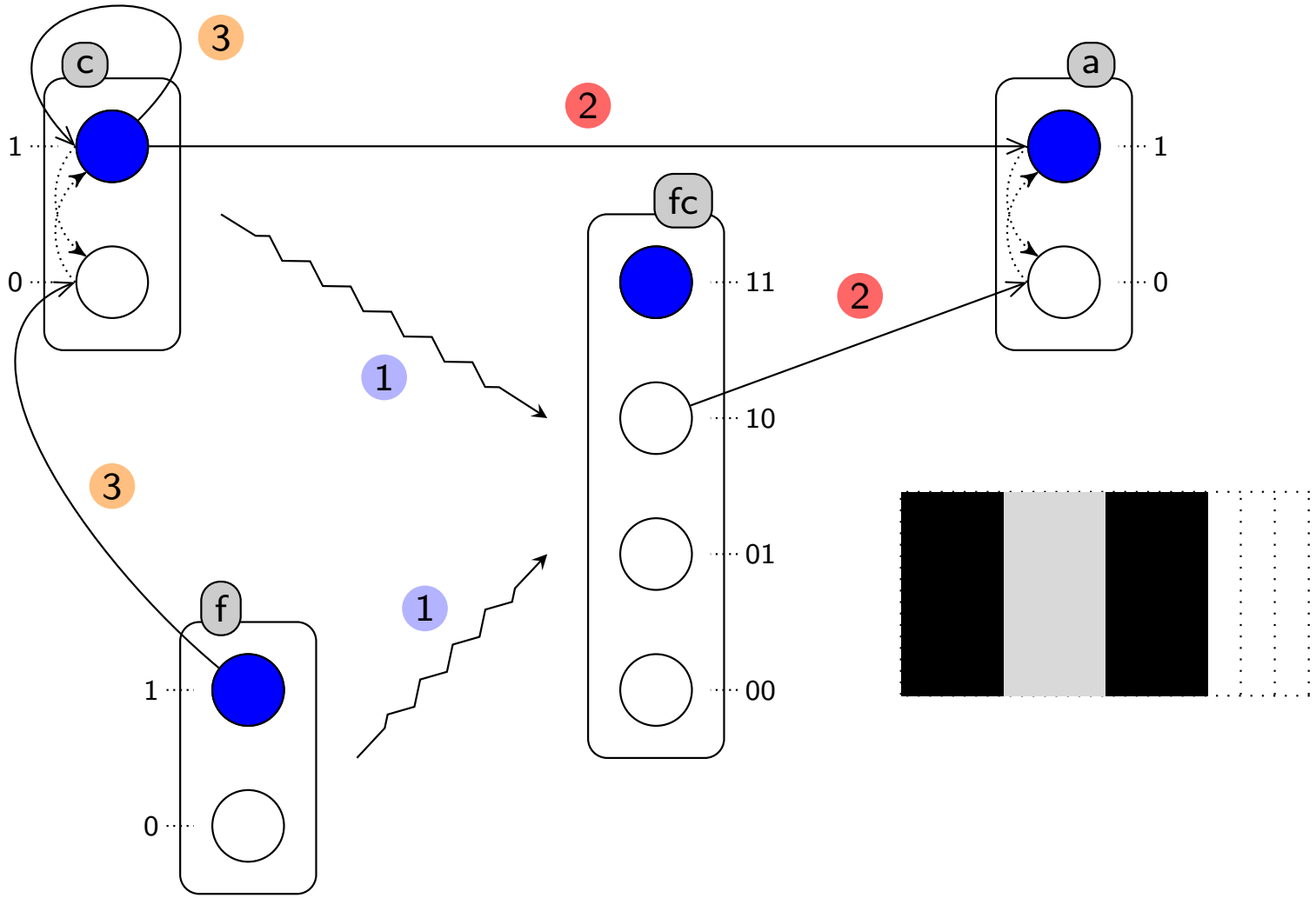
Metazoan Segmentation with Priorities

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



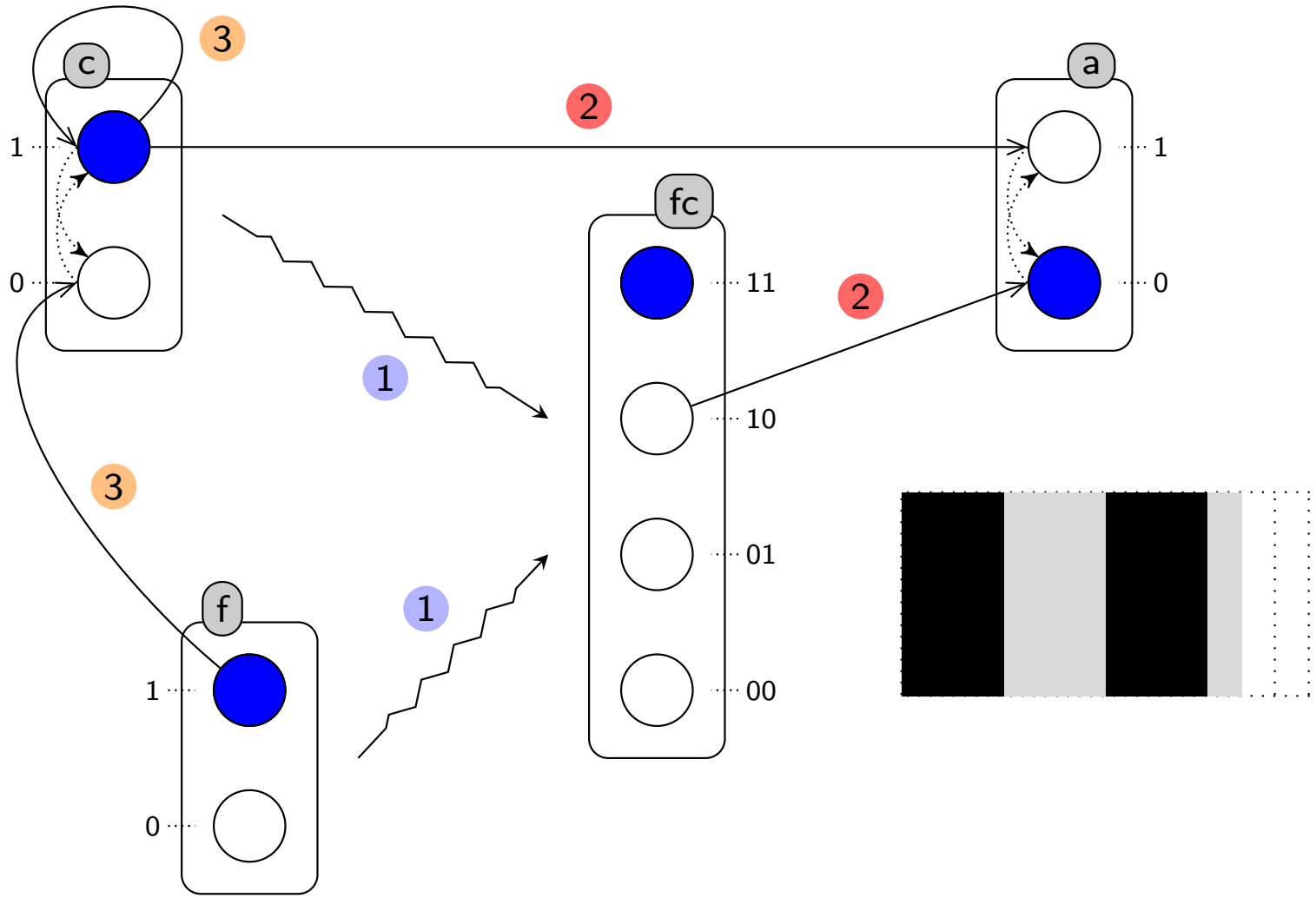
Metazoan Segmentation with Priorities

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



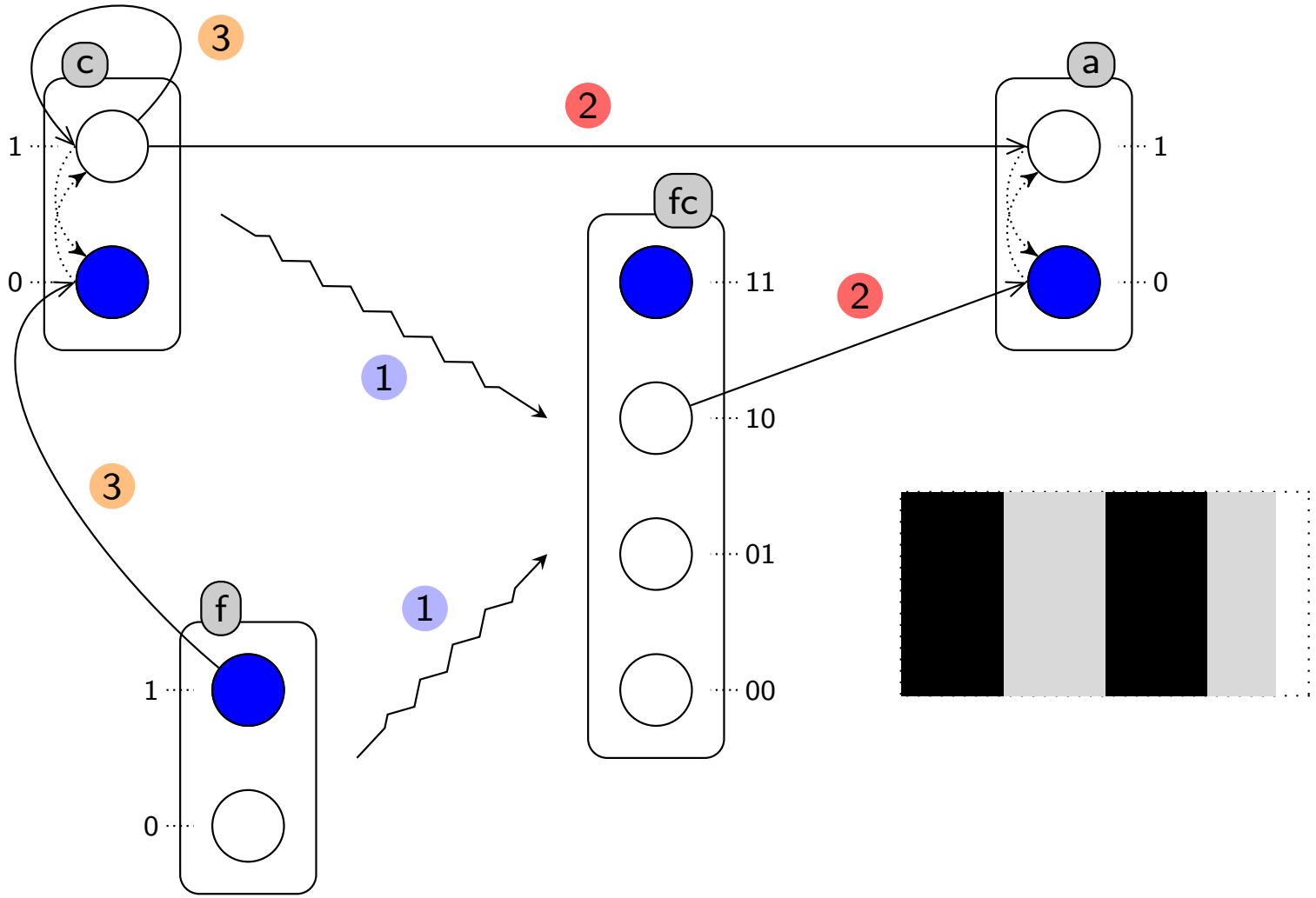
Metazoan Segmentation with Priorities

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



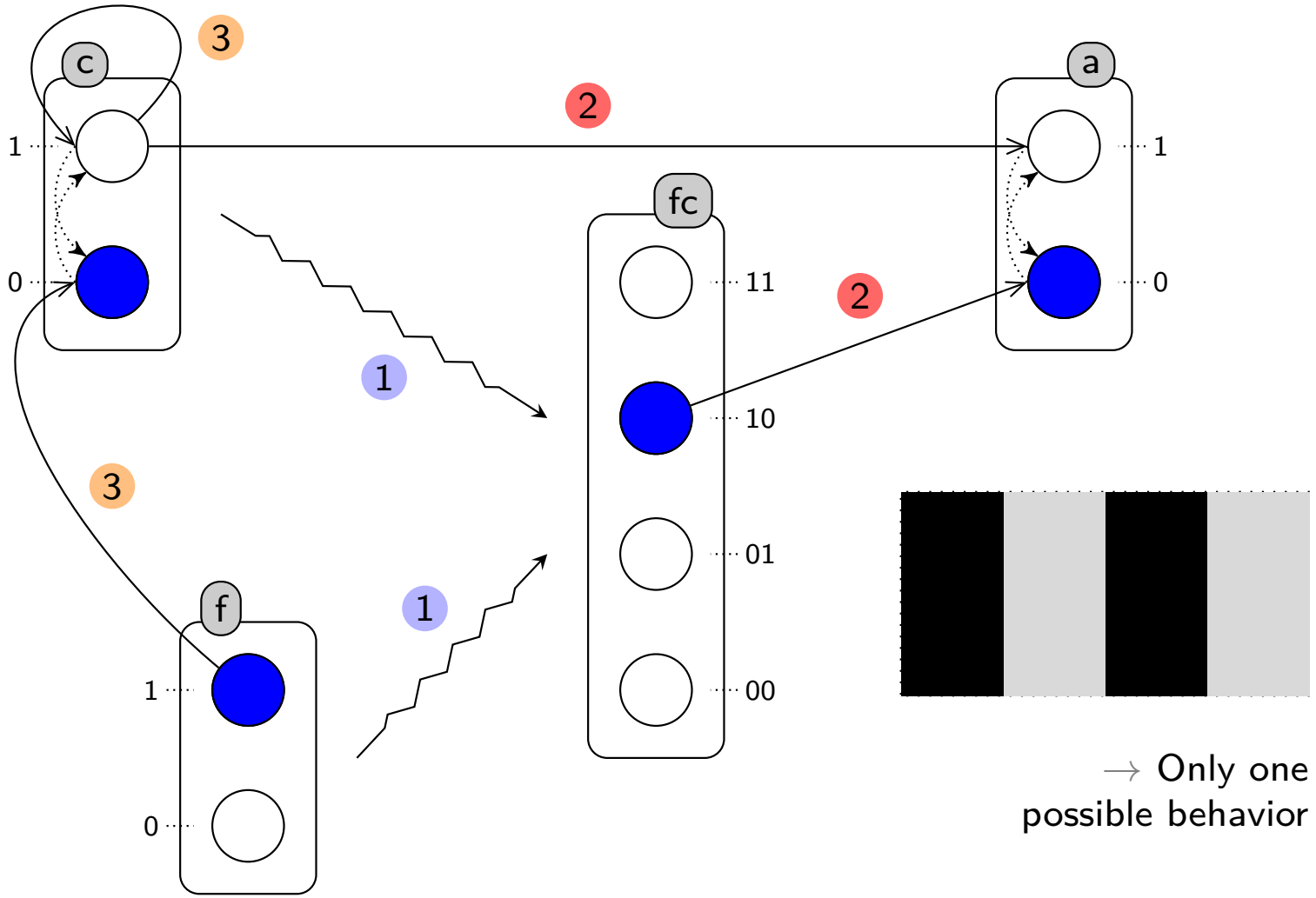
Metazoan Segmentation with Priorities

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



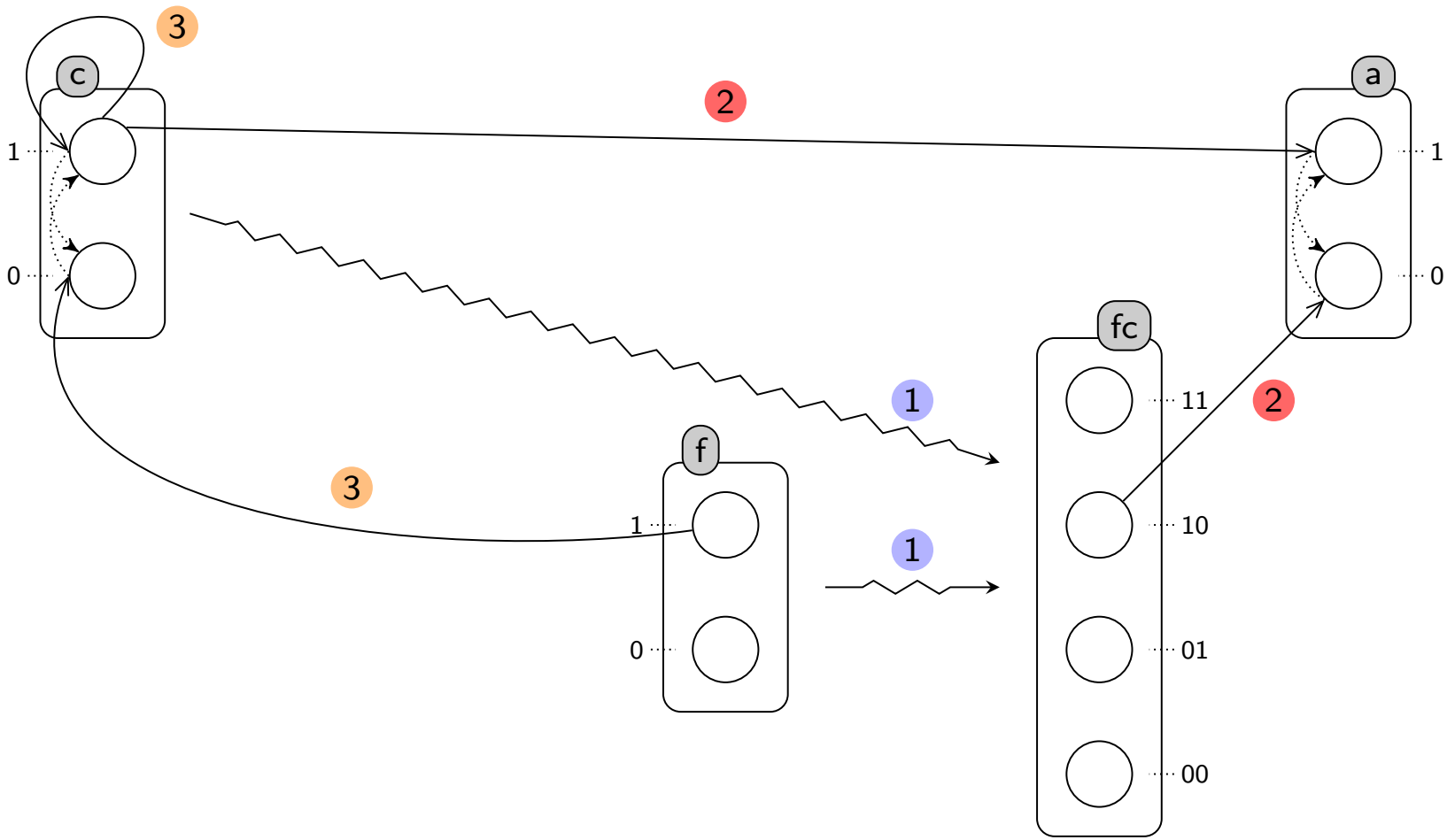
Metazoan Segmentation with Priorities

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

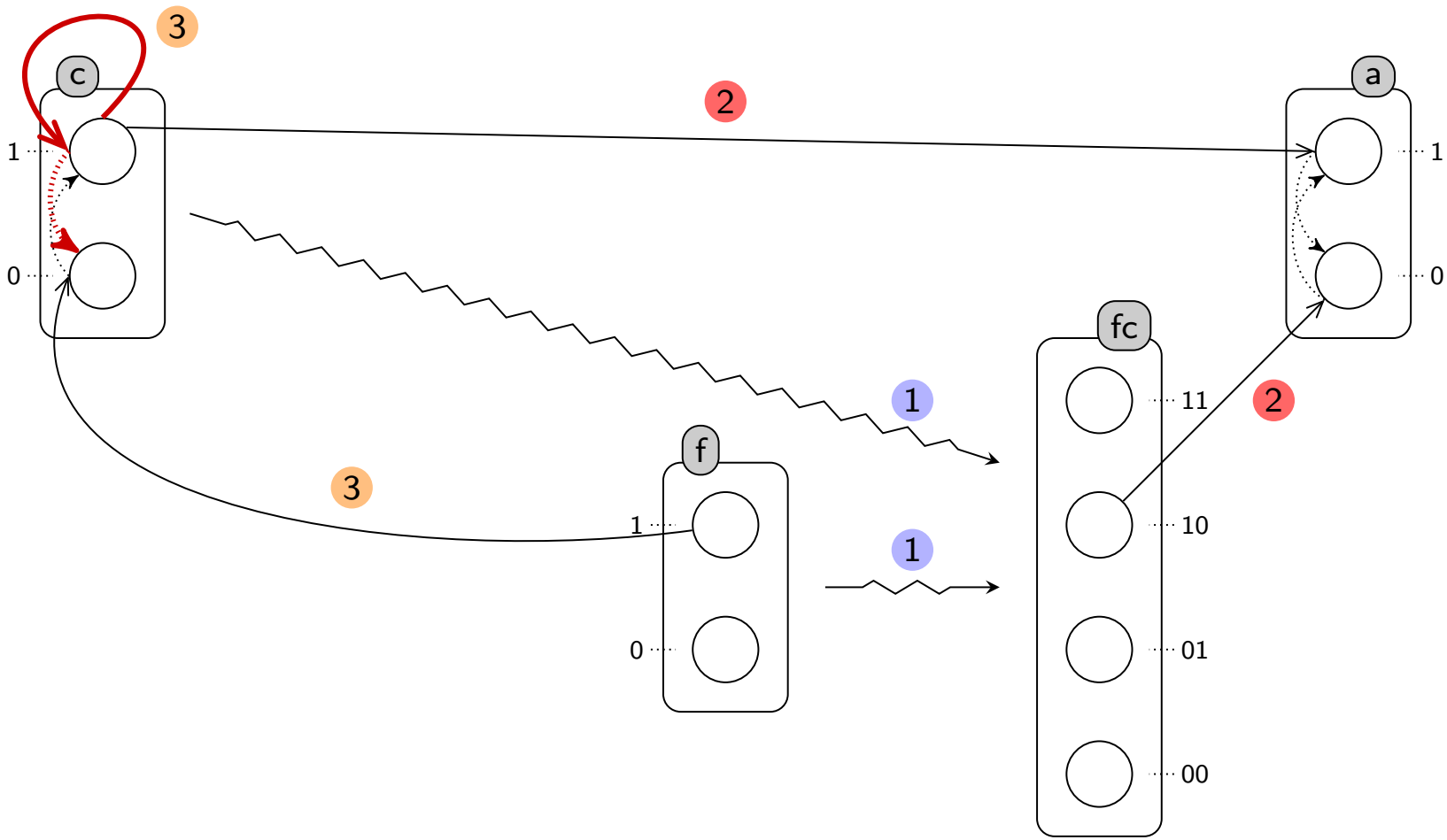


→ Only one possible behavior

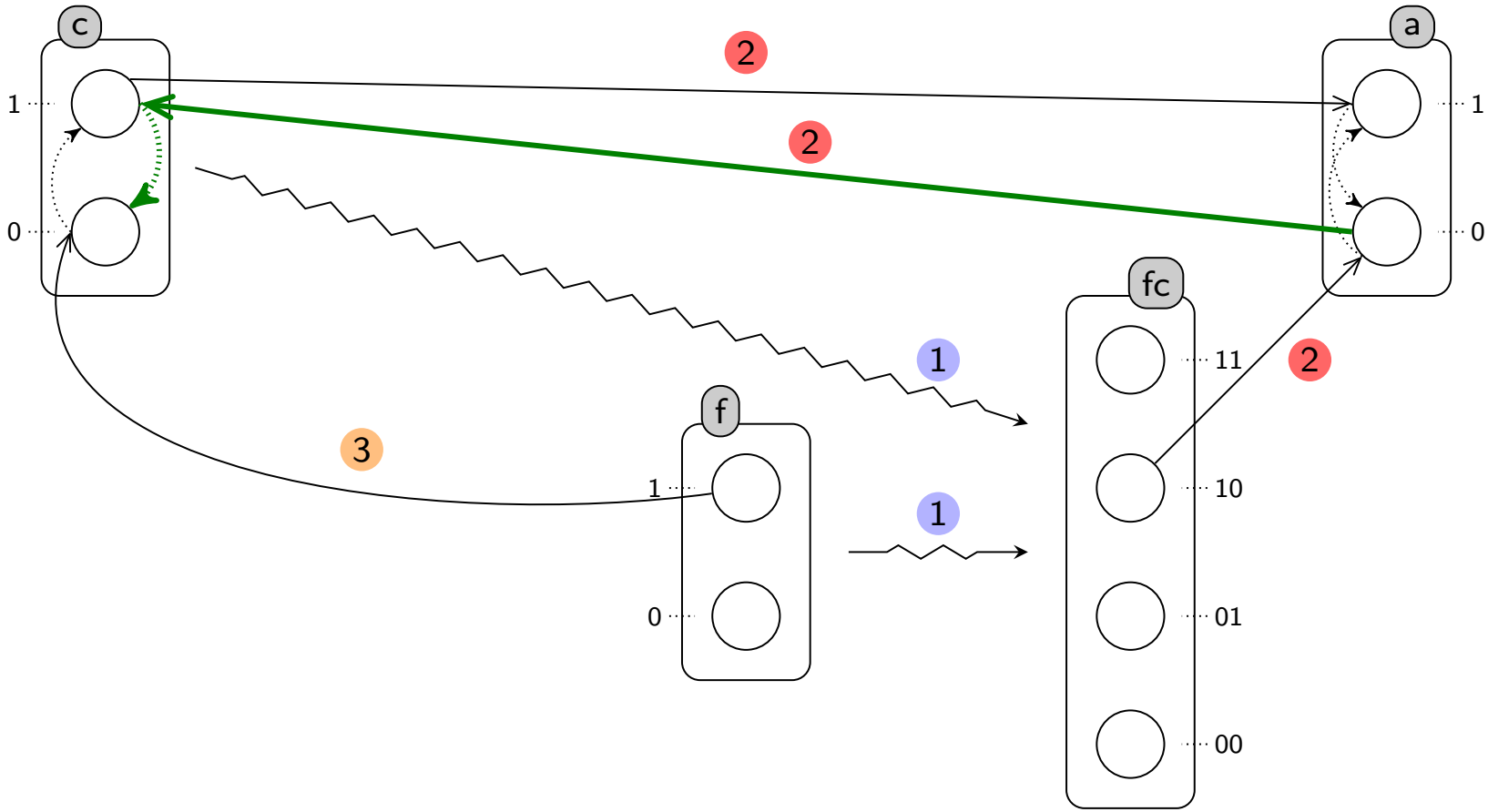
Metazoan Segmentation in Canonical Form



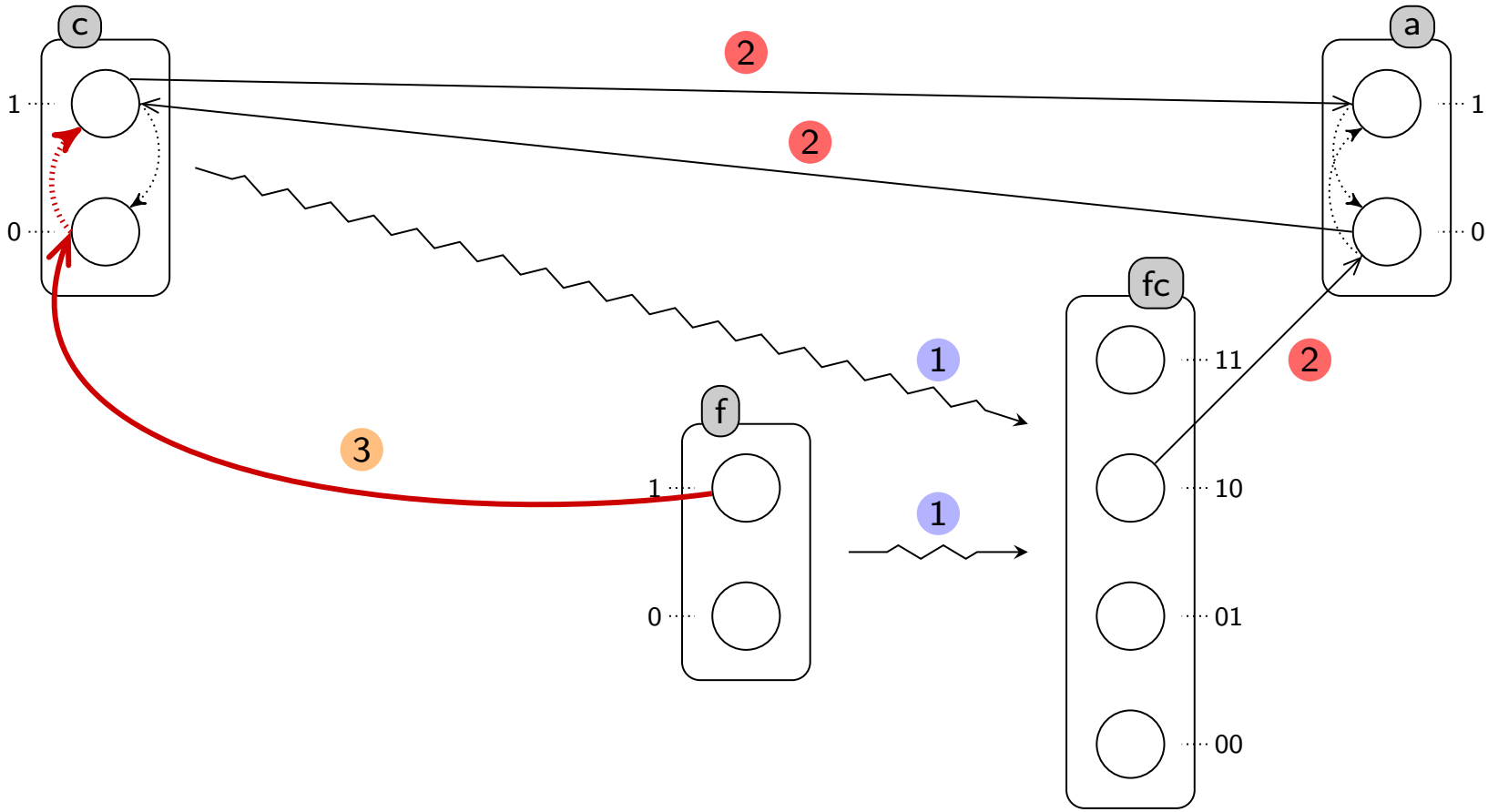
Metazoan Segmentation in Canonical Form



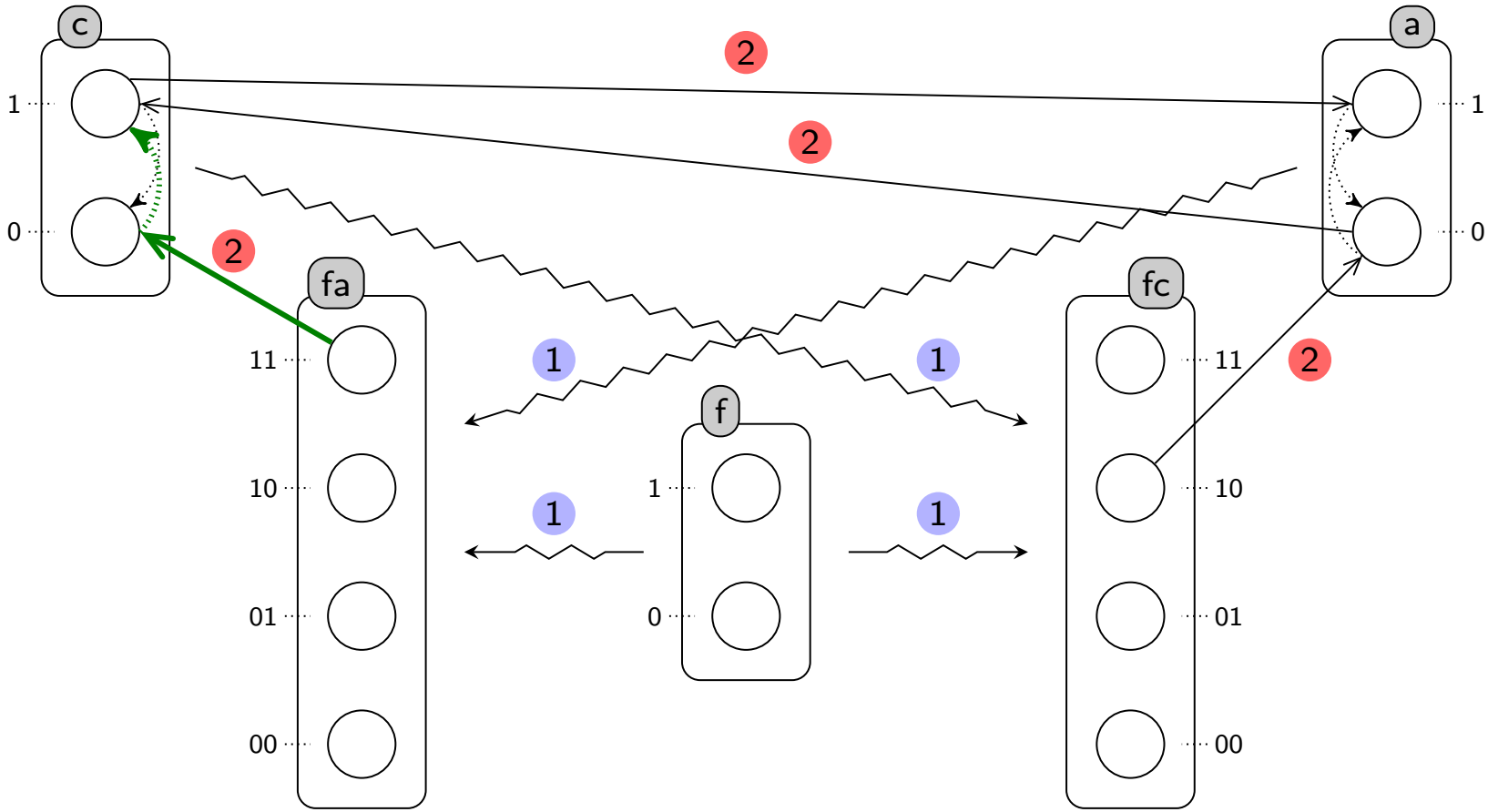
Metazoan Segmentation in Canonical Form



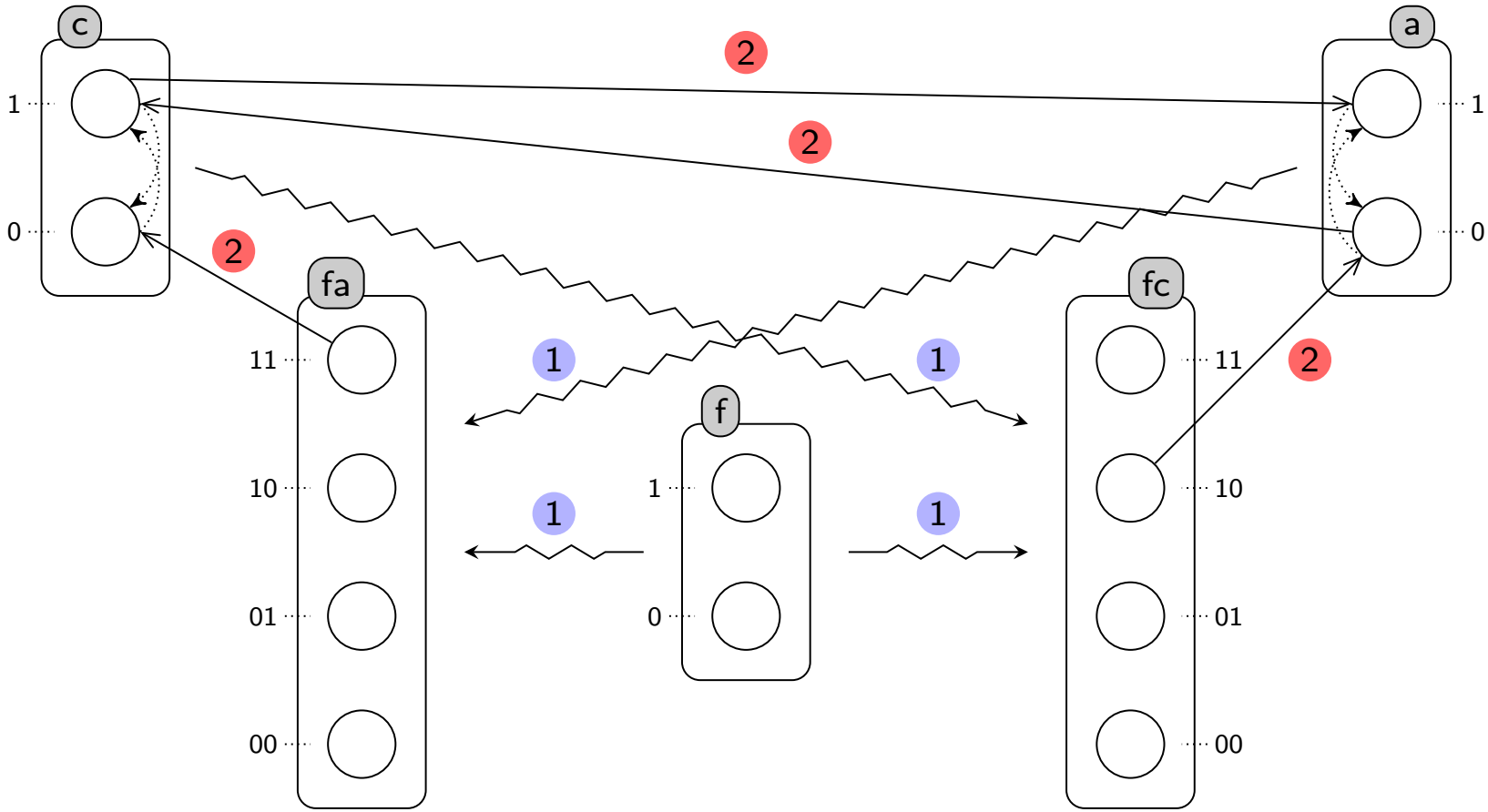
Metazoan Segmentation in Canonical Form



Metazoan Segmentation in Canonical Form



Metazoan Segmentation in Canonical Form



→ Same dynamics but only 2 priorities
→ Priority 1 is only for cooperative sorts

Pros and Cons of Classes of Priorities

Pros:

- Better expressivity (same as Boolean Networks!)
- Efficient static analysis

Model	Sorts	Procs	Actions	States	libddd[1]	PINT
tcrsig94	133	448	1124	2^{194}	∞	**0.008s – 0.060s**

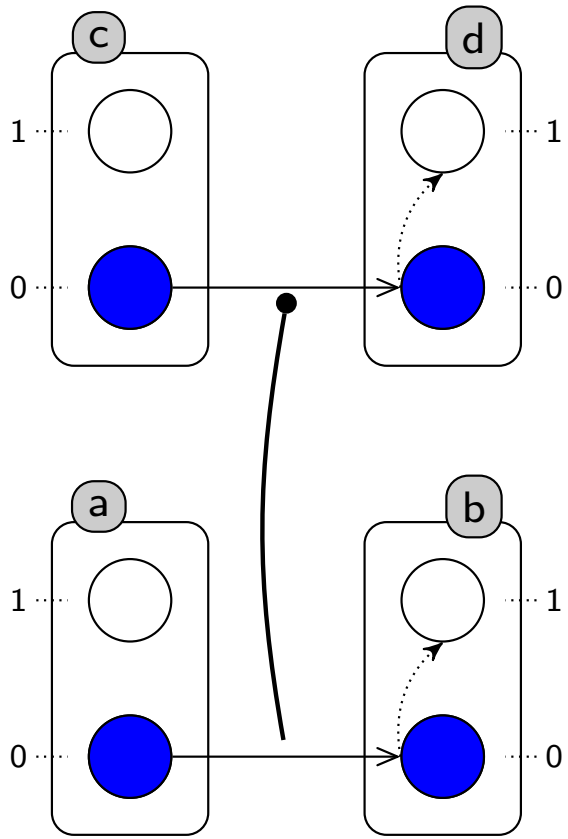
[1] LIP6/Move

tcrsig94: [T-Cell Receptor Signaling, by Julio Saez-Rodriguez *et al.*]
(Here with prioritized cooperative sorts)

Cons:

- No accumulation phenomenons
  Contrary to stochastic simulation
- Translation to canonical form is exponential
  → For each action of priority n, exponential in the number of actions of priority $[\![2; n-1]\!]$

Neutralizing Edges

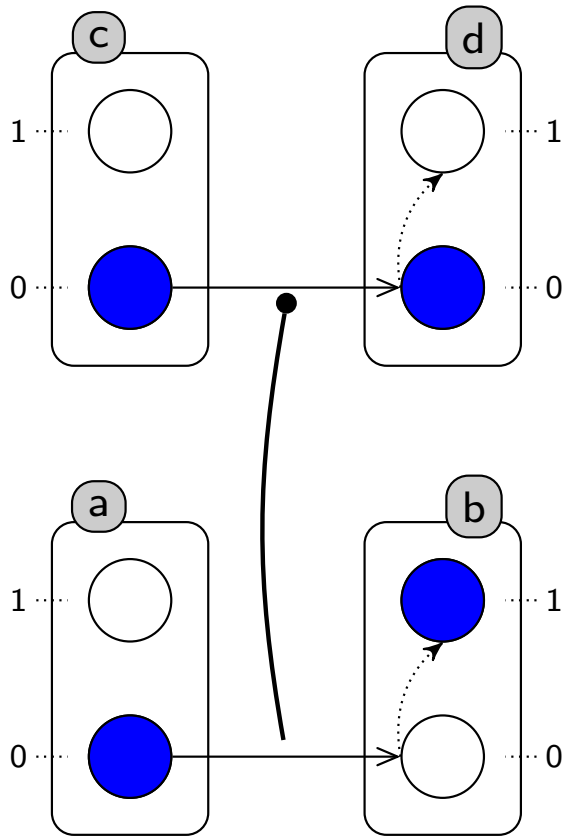


$c_0 \rightarrow d_0 \nearrow d_1$ cannot be played **while**

$a_0 \rightarrow b_0 \nearrow b_1$ is playable

$\rightarrow$ Here, only one possible behavior

Neutralizing Edges

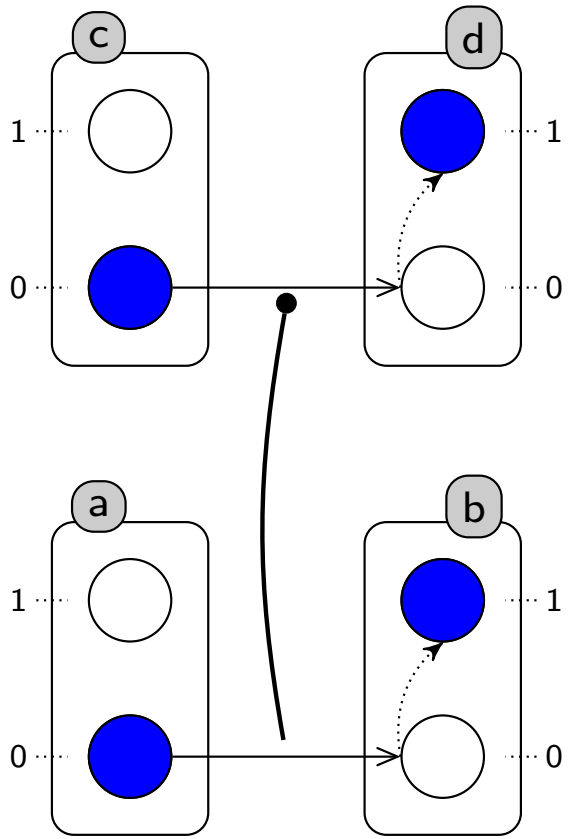


$c_0 \rightarrow d_0 \nearrow d_1$ cannot be played **while**

$a_0 \rightarrow b_0 \nearrow b_1$ is playable

$\rightarrow$ Here, only one possible behavior

Neutralizing Edges

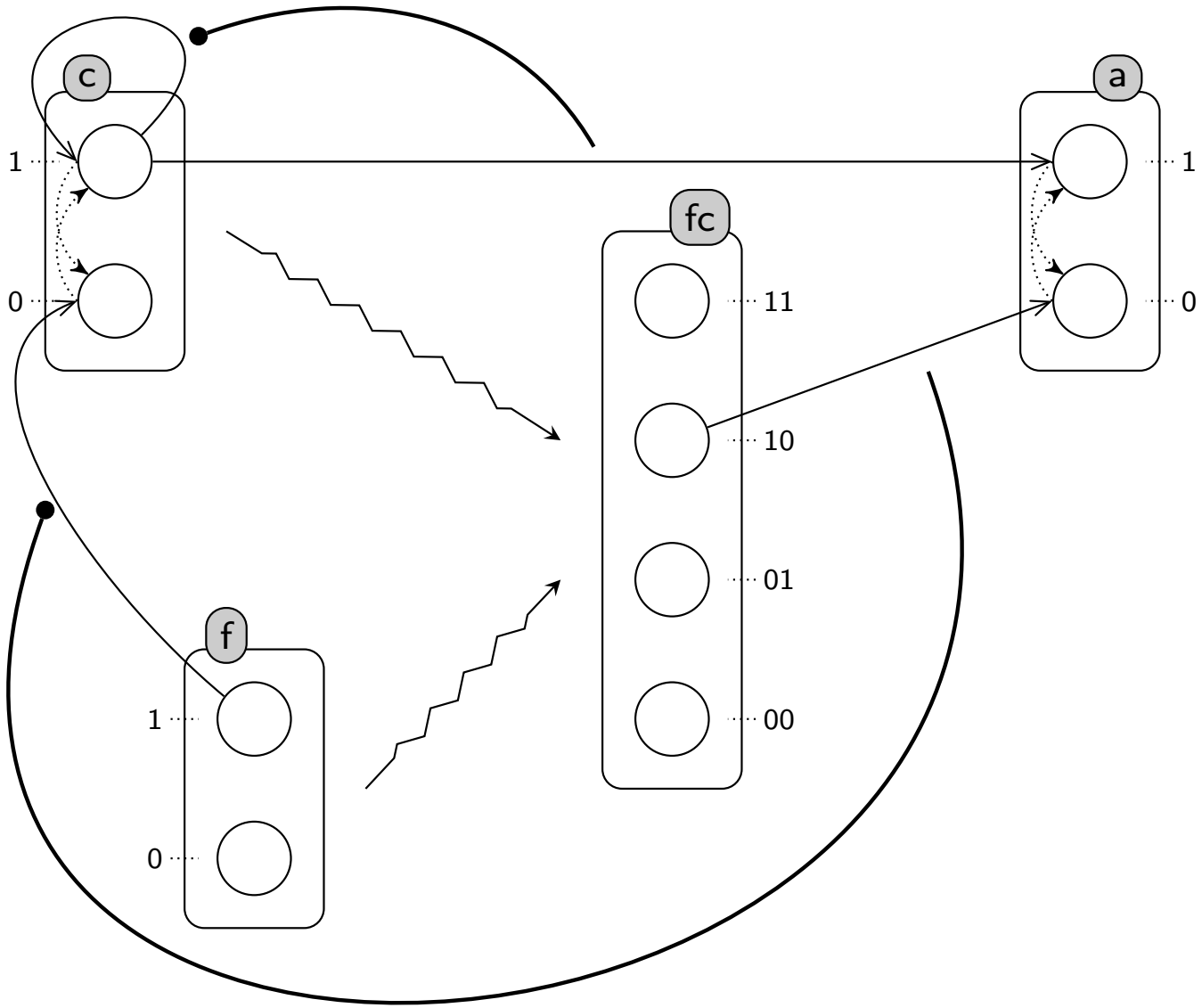


$c_0 \rightarrow d_0 \Rsh d_1$ cannot be played **while**

$a_0 \rightarrow b_0 \Rsh b_1$ is playable

$\rightarrow$ Here, only one possible behavior

Metazoan Segmentation with Neutralizing Edges



Pros and Cons of Neutralizing Edges

Pros:

- Same expressivity than Priorities
  - → Can be translated to the canonical form
- Finer preemption relations
  - → Easier modeling in some cases
- Sparser constraints
  - → More efficient translation to canonical form

Summary & Conclusion

Process Hitting: an atomistic modeling with powerful static analysis

1. Stochastic parameters:
   - To model systems with chronometric features
   - **Continuous time**
   - But **hard to analyze**
2. Classes of priorities:
   - Allows to reproduce the same behaviors
   - Efficient **static analysis**
   - But the translation to canonical form faces **combinatorial explosion**
3. Neutralizing edges:
   - Alternative to priorities
   - Closer to reality in some cases
   - **Lighter translation** to canonical form

Thank you

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*, volume 6575 of Lecture Notes in Computer Science, 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.
- Maxime Folschette, Loïc Paulevé, Morgan Magnin, Olivier Roux. Under-approximation of Reachability in Multivalued Asynchronous Networks. In E. Merelli and A. Troina, editors, *4th International Workshop on Interactions between Computer Science and Biology (CS2Bio'13)*, Electronic Notes in Theoretical Computer Science, Volume 299, 33–51. June 2013.
- Loïc Paulevé. PhD thesis: *Modélisation, Simulation et Vérification des Grands Réseaux de Régulation Biologique*, October 2011, Nantes, France.
- Paul François, Vincent Hakim, Eric D Siggia. Deriving structure from evolution : metazoan segmentation. In *Molecular Systems Biology*, Volume 3, Issue 1. 2007.

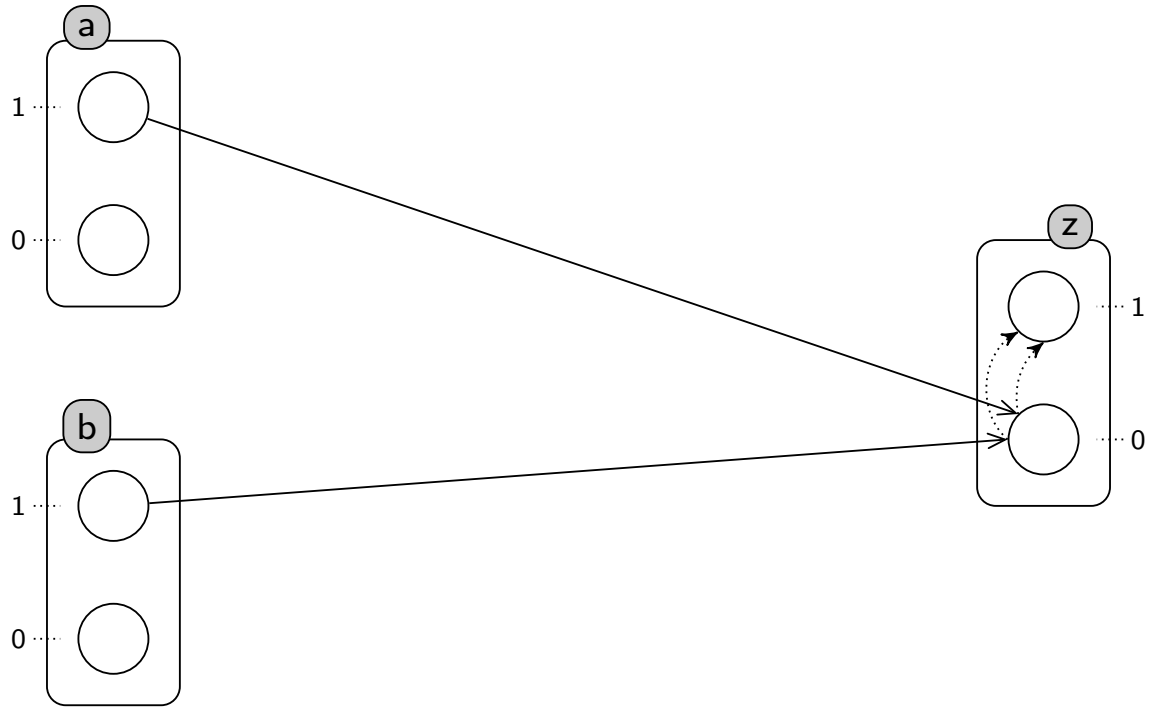
Adding cooperations

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

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

Adding cooperations

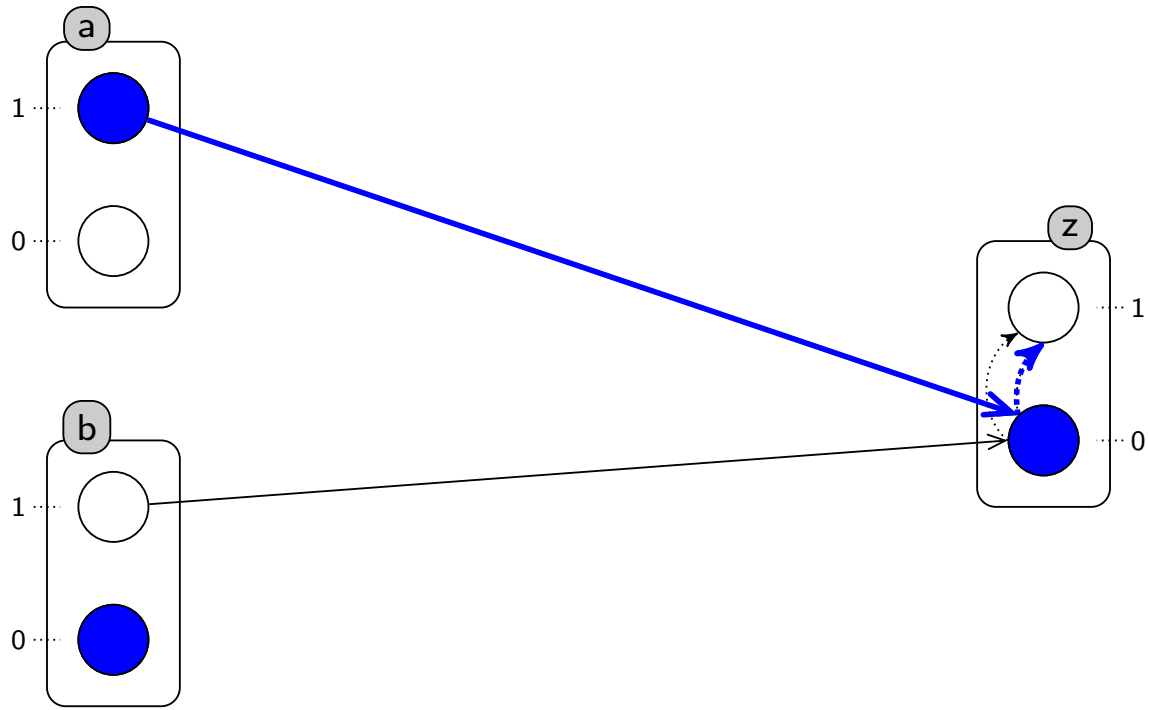
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Adding cooperations

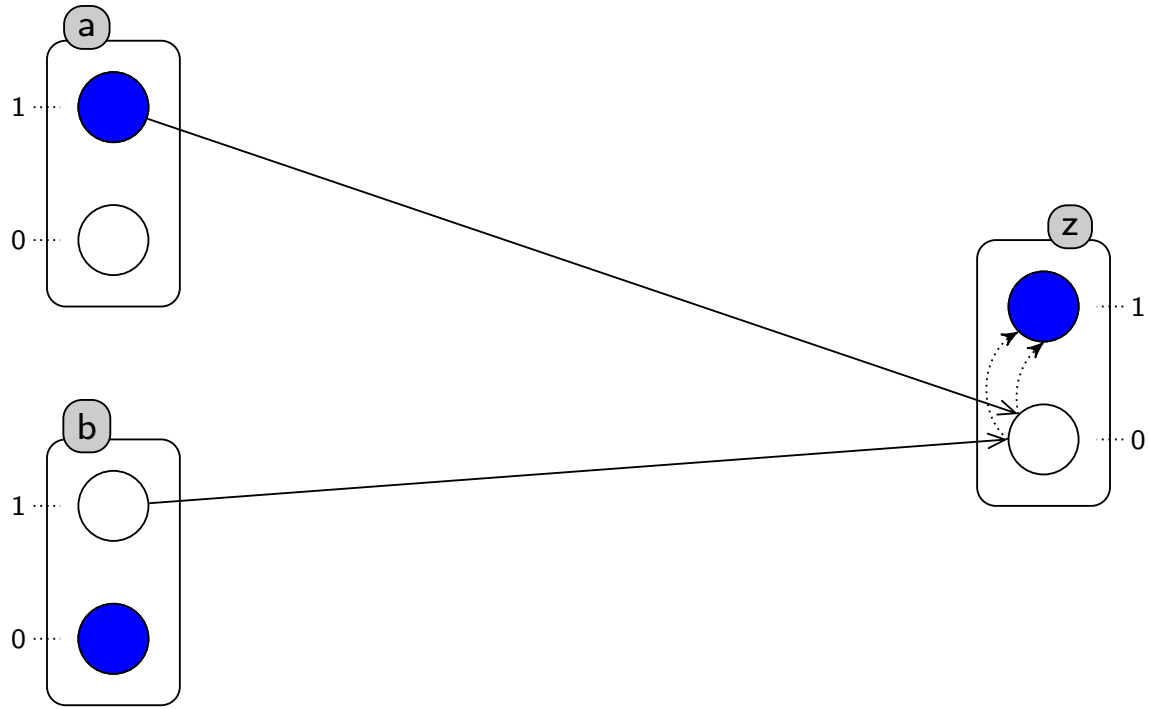
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Adding cooperations

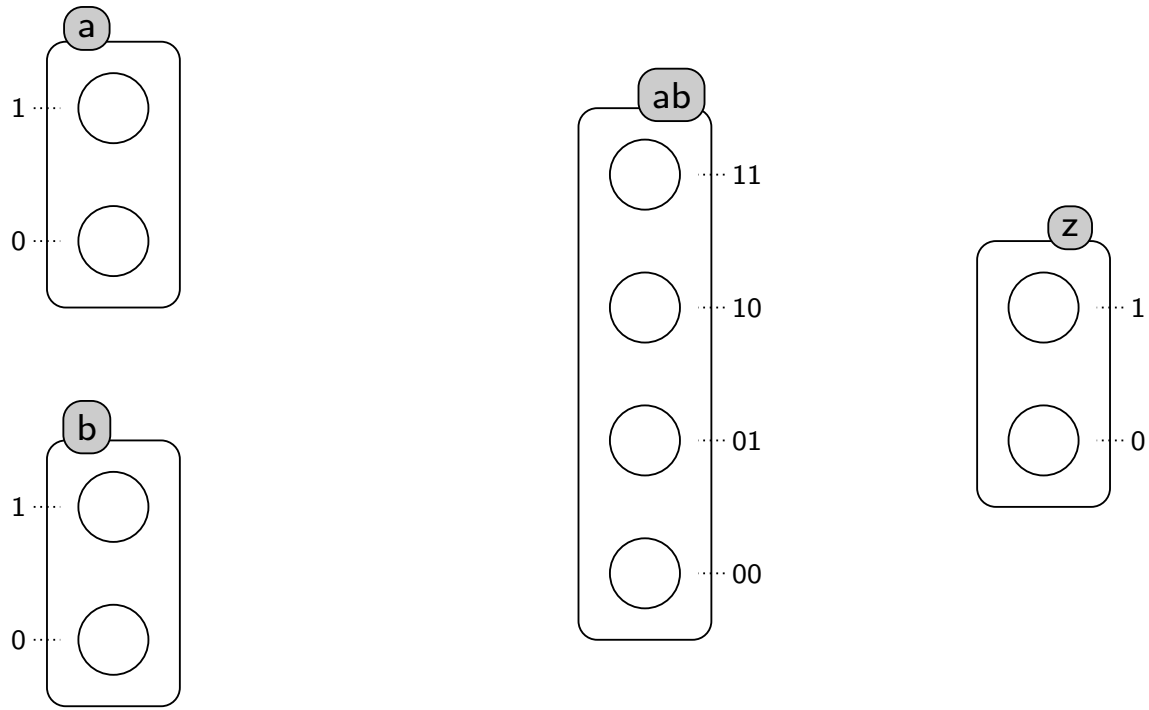
[Paulevé *et al.* in Transactions on Computational Systems Biology, 2011]



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

Adding cooperations

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

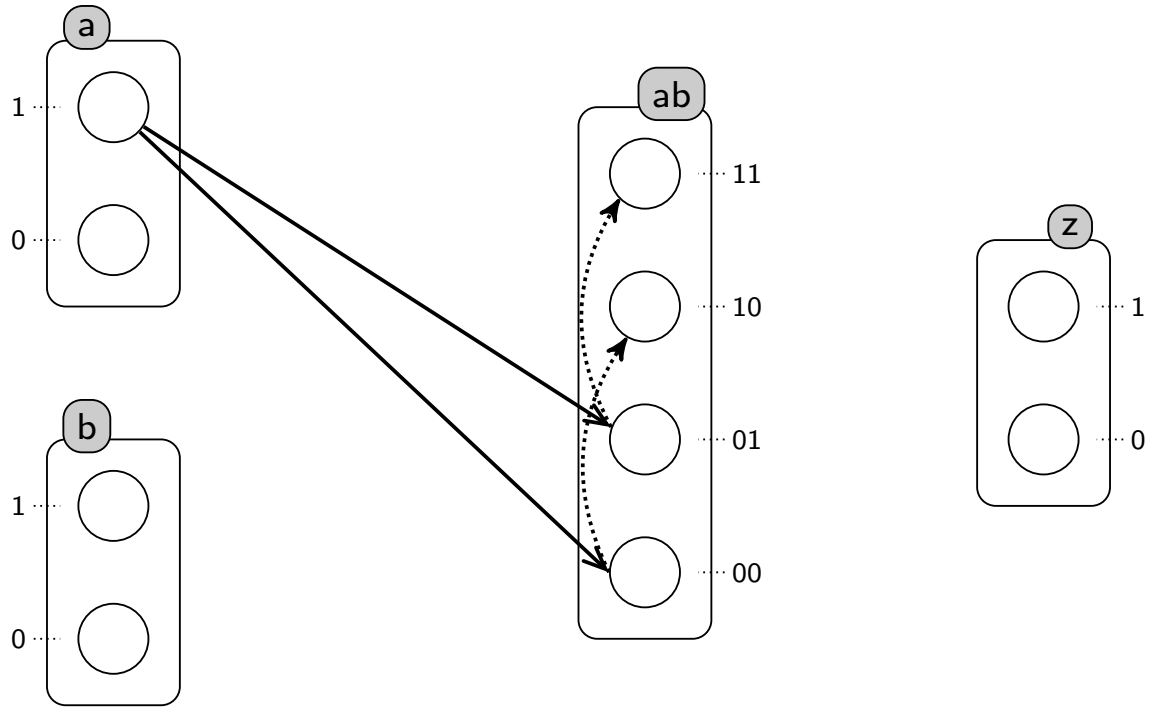


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

Adding cooperations

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

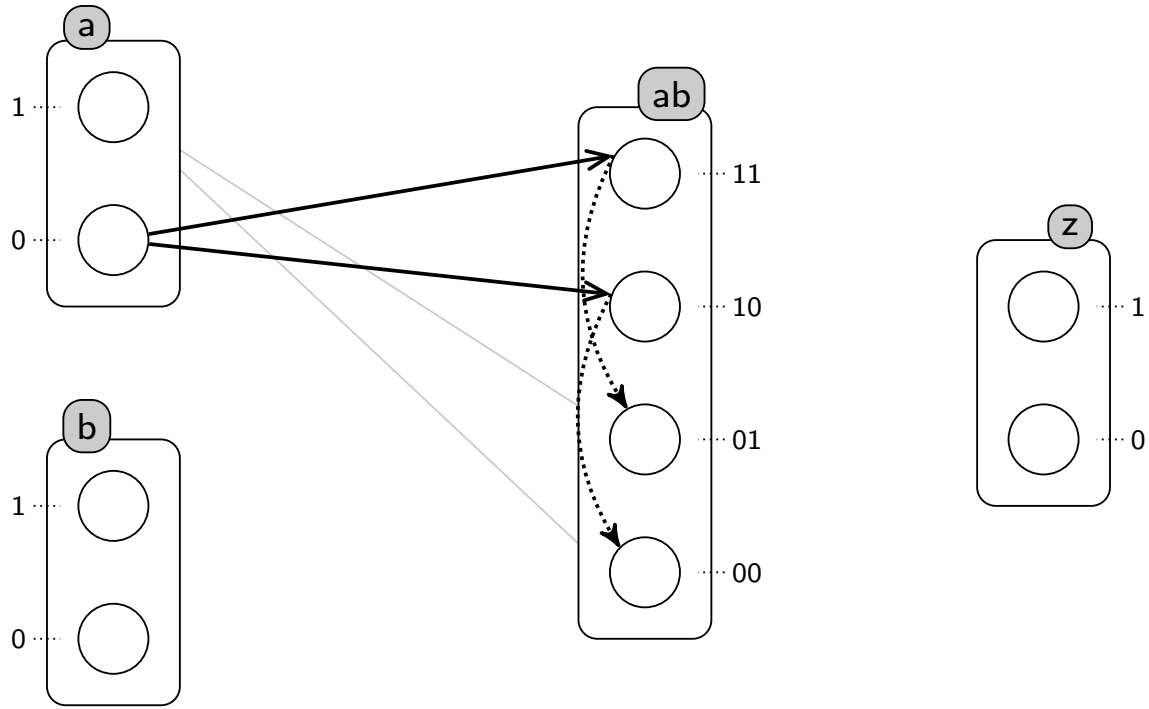


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

Adding cooperations

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

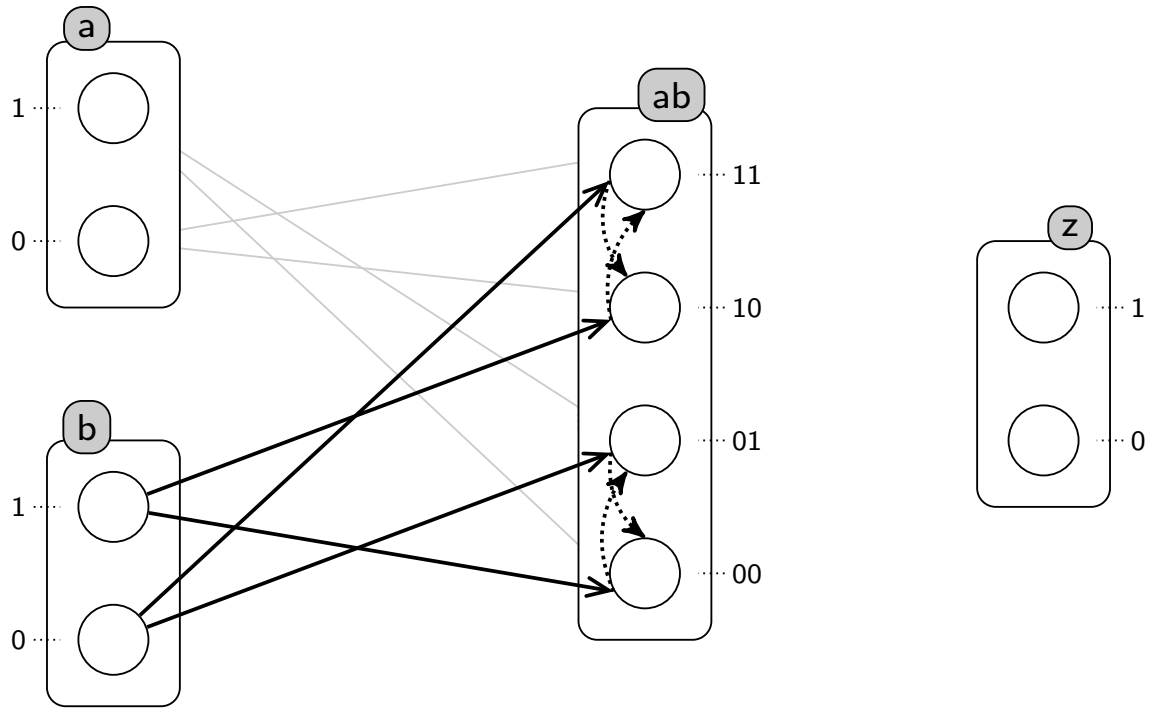


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

Adding cooperations

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

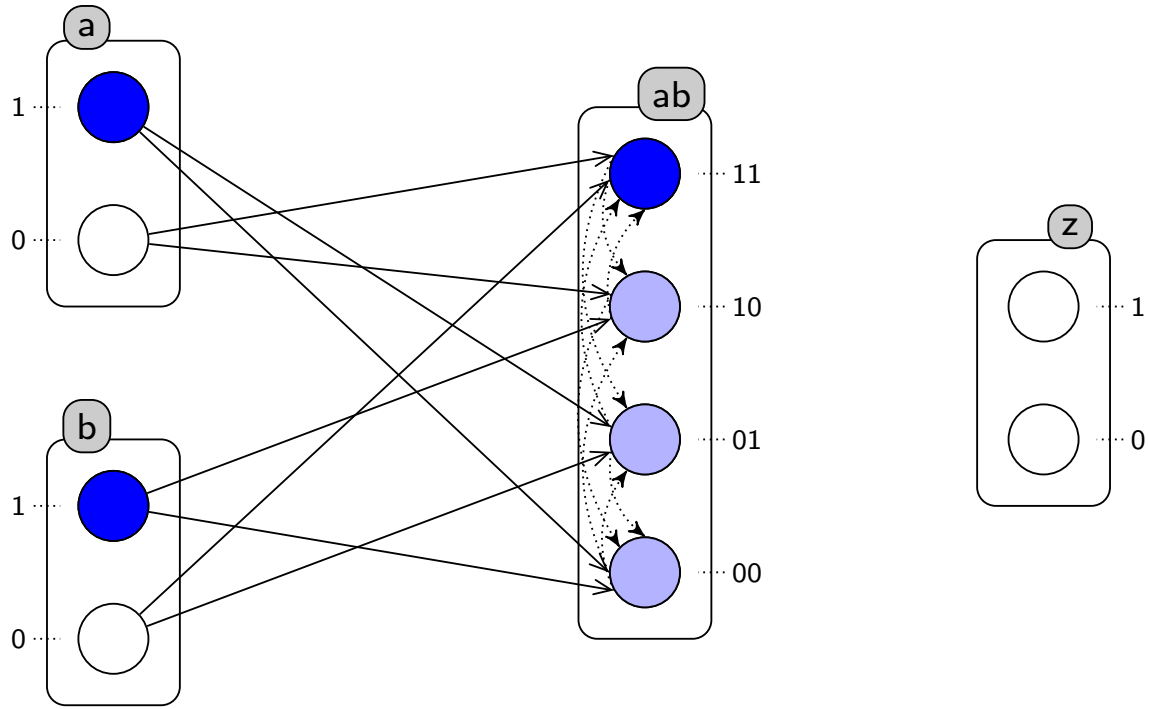


Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

Adding cooperations

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



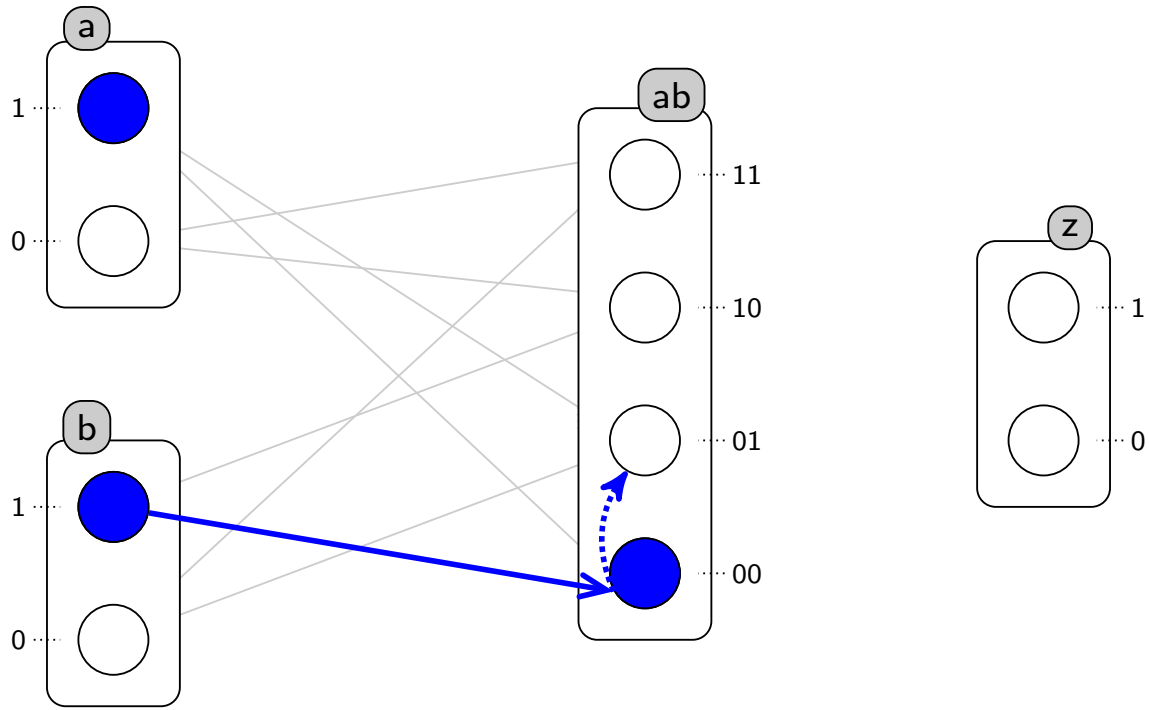
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

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

Adding cooperations

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



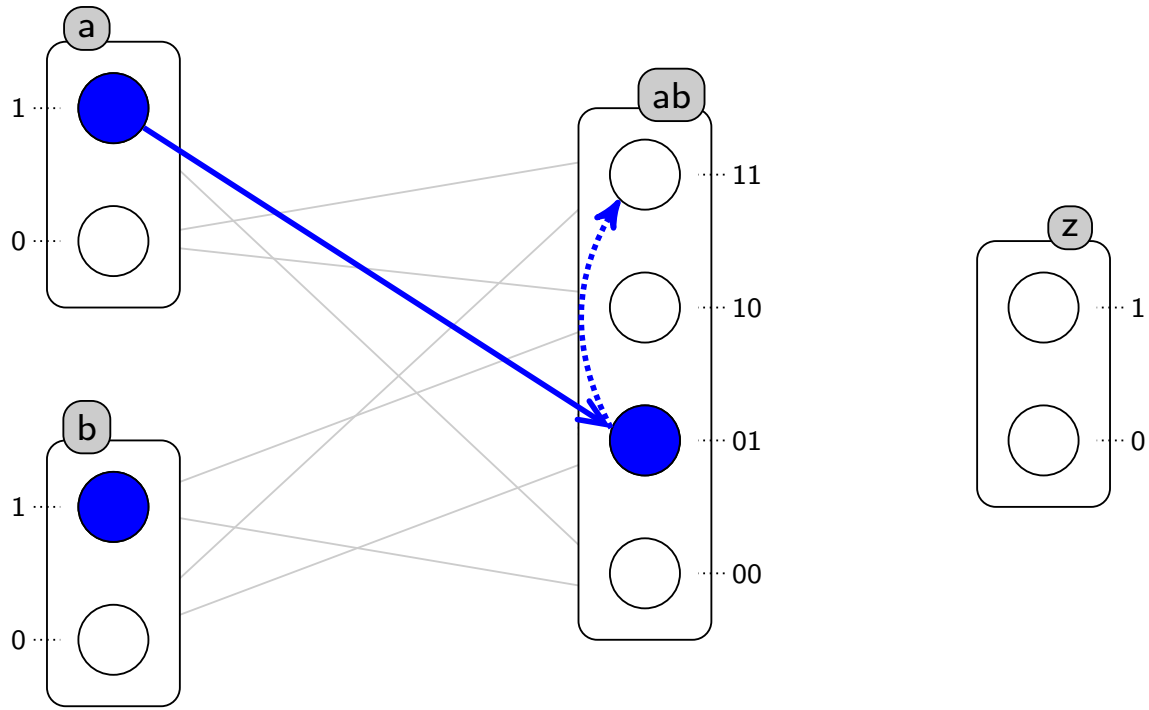
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

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

Adding cooperations

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



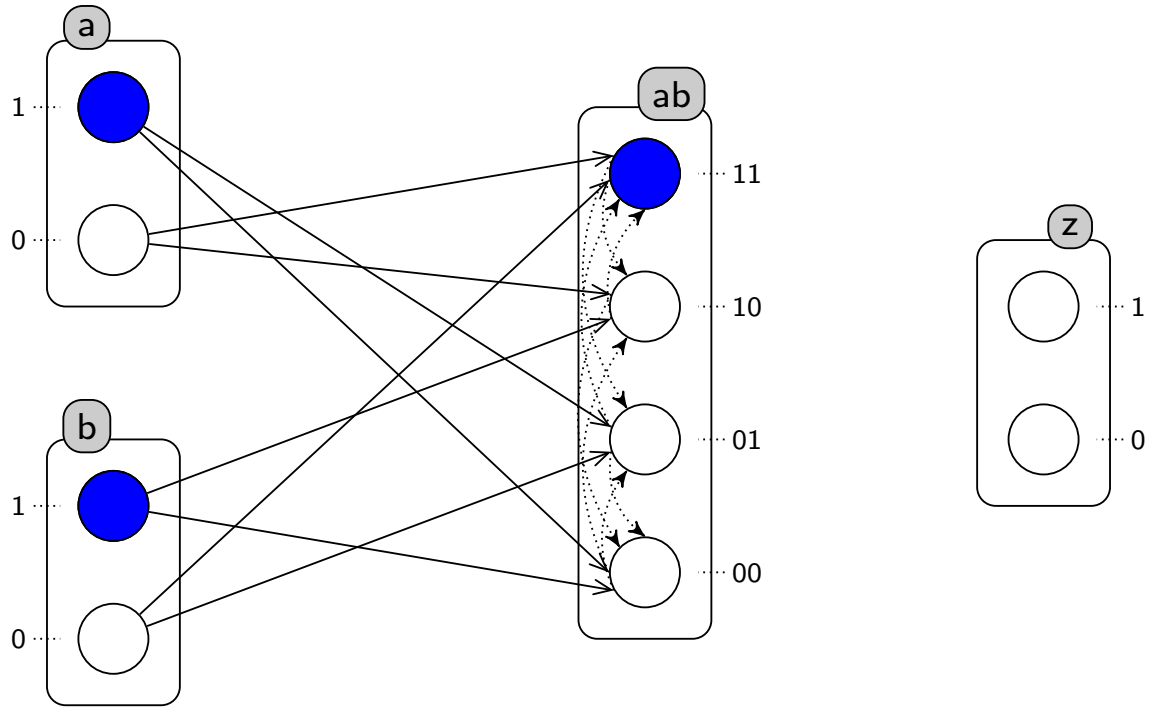
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

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

Adding cooperations

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



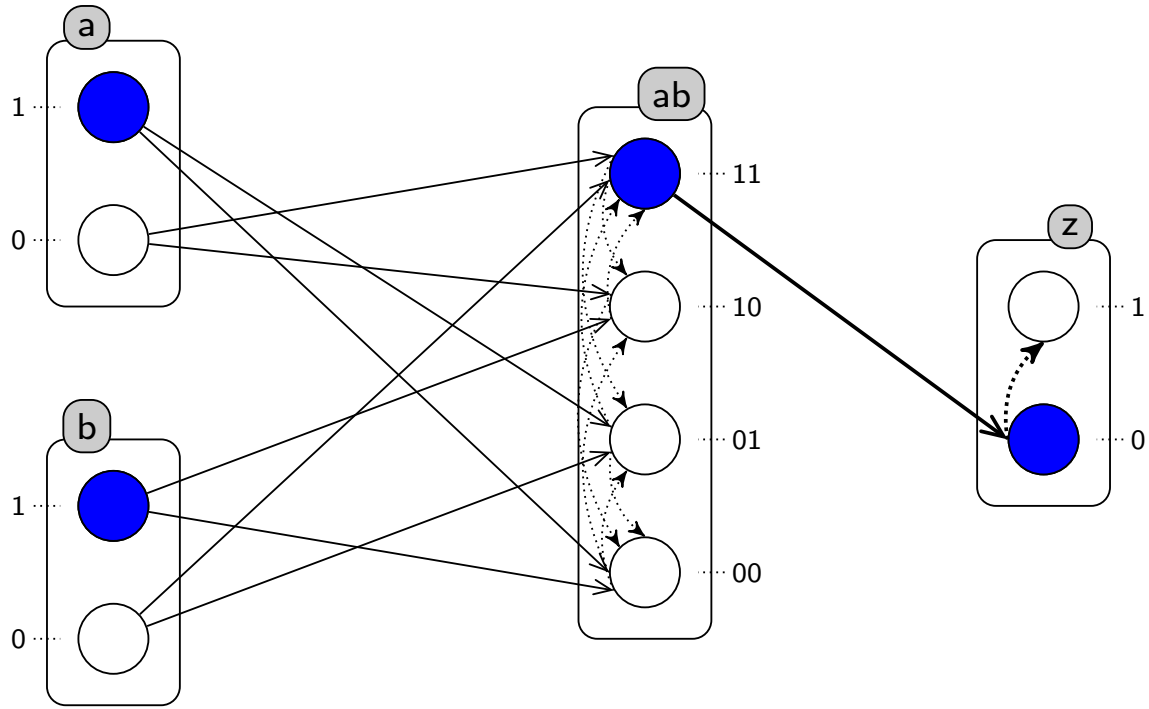
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \Rsh z_1$

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

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

Adding cooperations

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



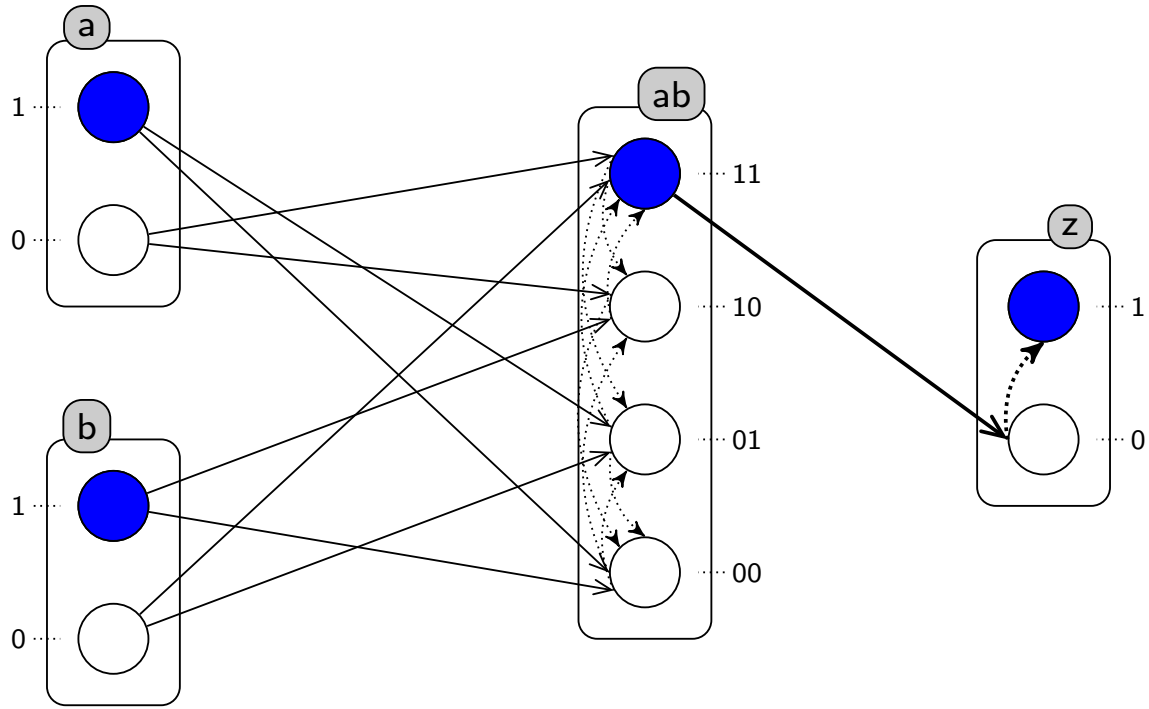
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \not\uparrow z_1$

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

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

Adding cooperations

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



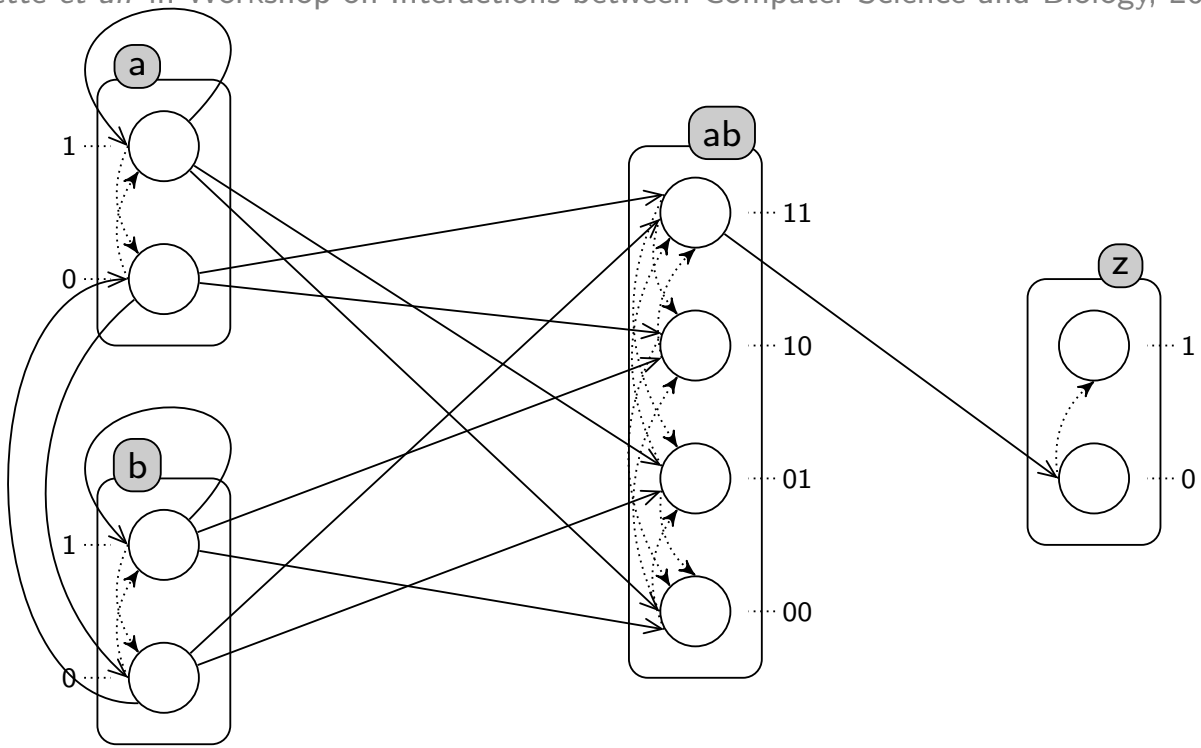
Cooperation between a_1 and b_1: $\underline{a_1 \wedge b_1} \rightarrow z_0 \uparrow z_1$

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

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

Adapting the expressivity of PH

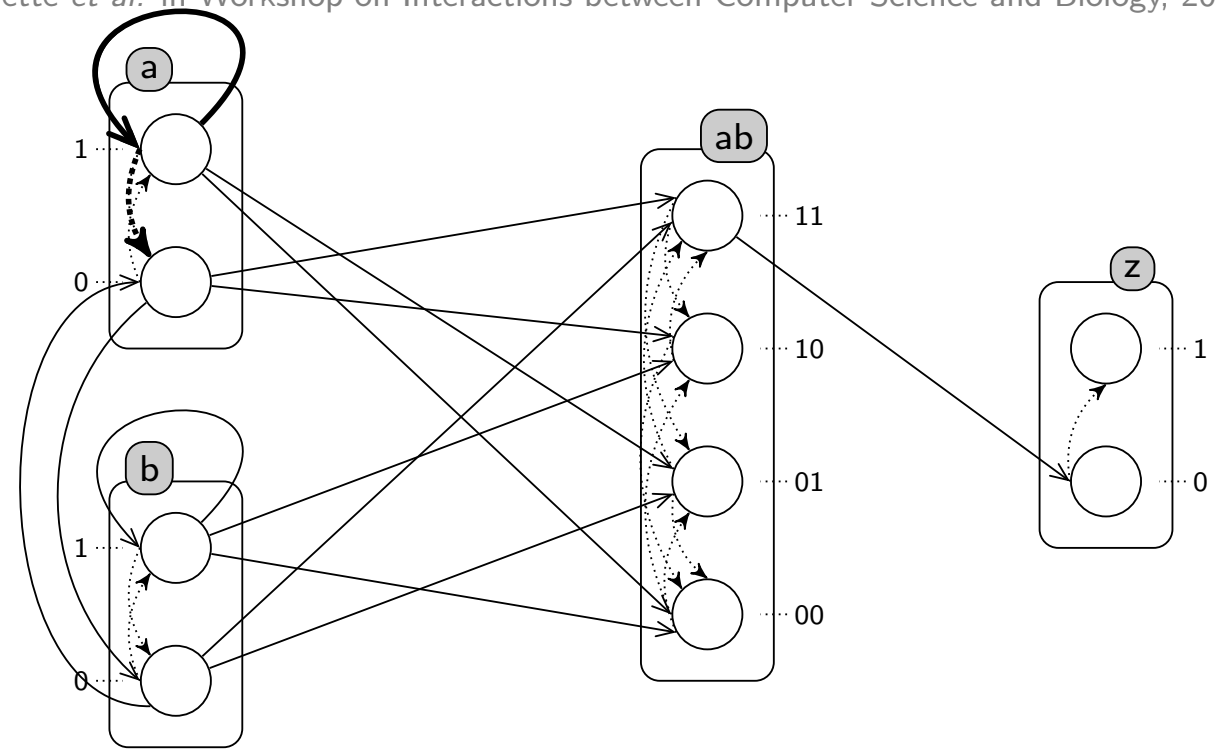
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]



Drawback: Cooperations are too "loose" to be as expressive as ADN.

Adapting the expressivity of PH

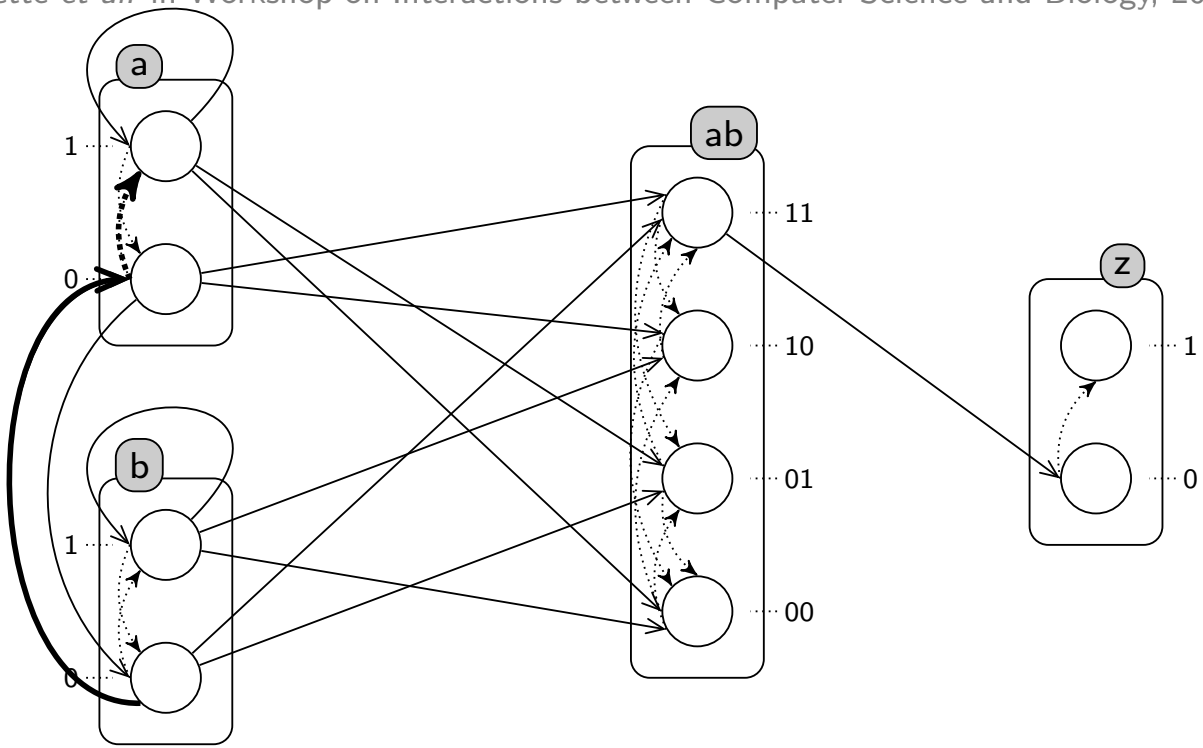
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]



Drawback: Cooperations are too "loose" to be as expressive as ADN.

Adapting the expressivity of PH

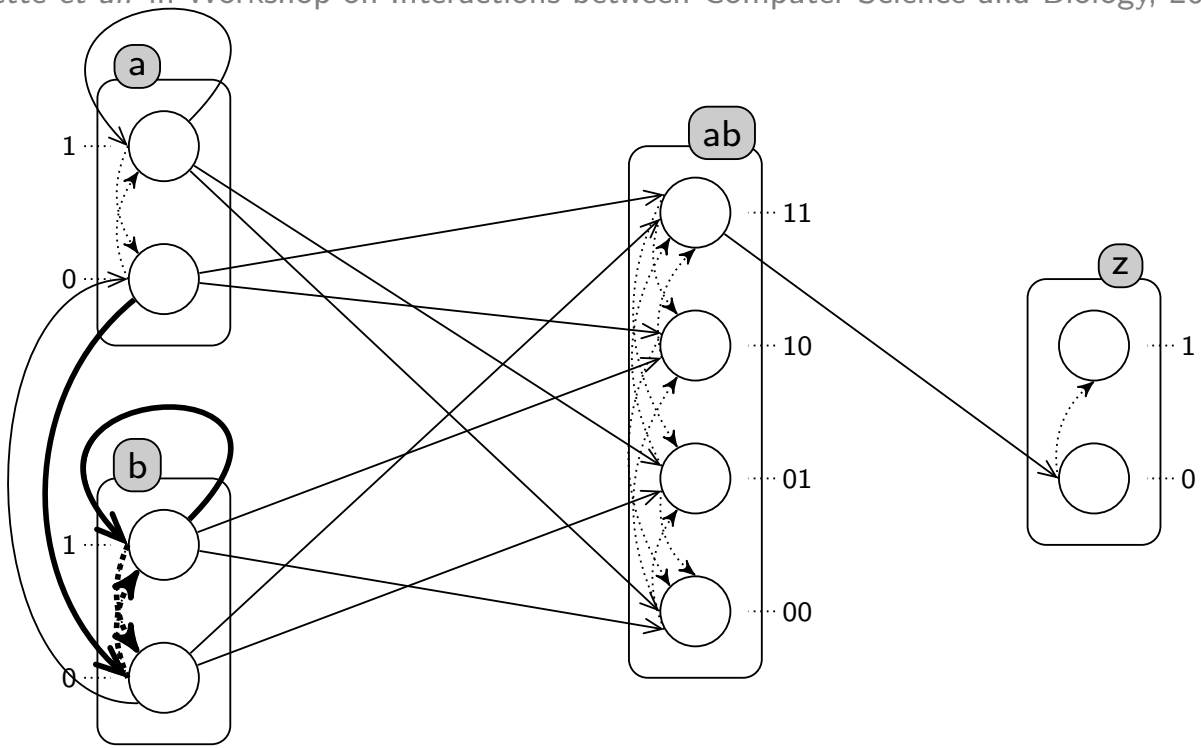
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]



Drawback: Cooperations are too "loose" to be as expressive as ADN.

Adapting the expressivity of PH

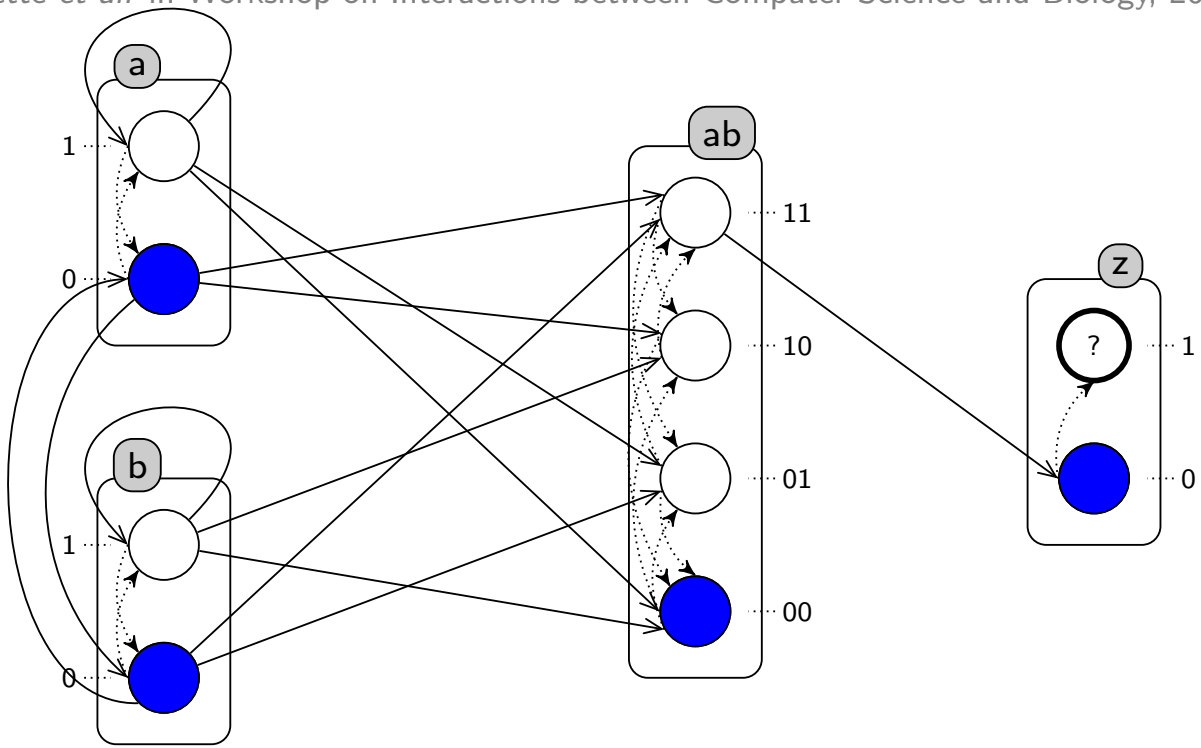
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]



Drawback: Cooperations are too "loose" to be as expressive as ADN.

Adapting the expressivity of PH

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

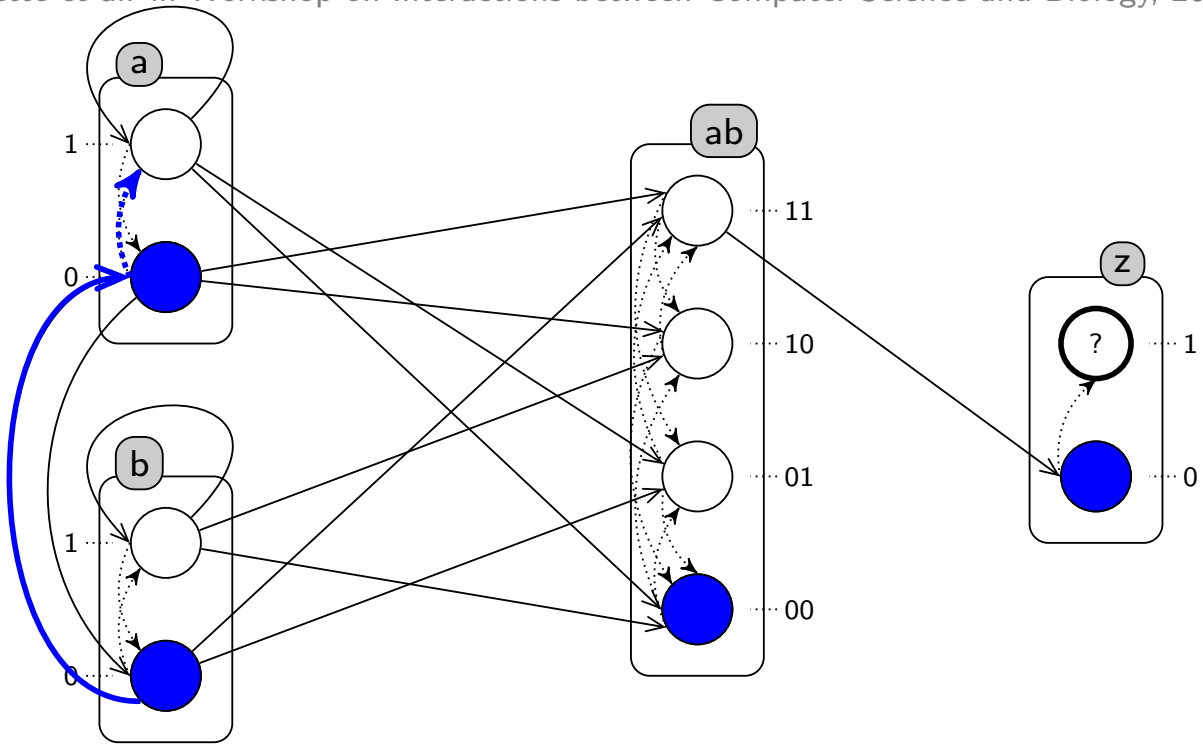


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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

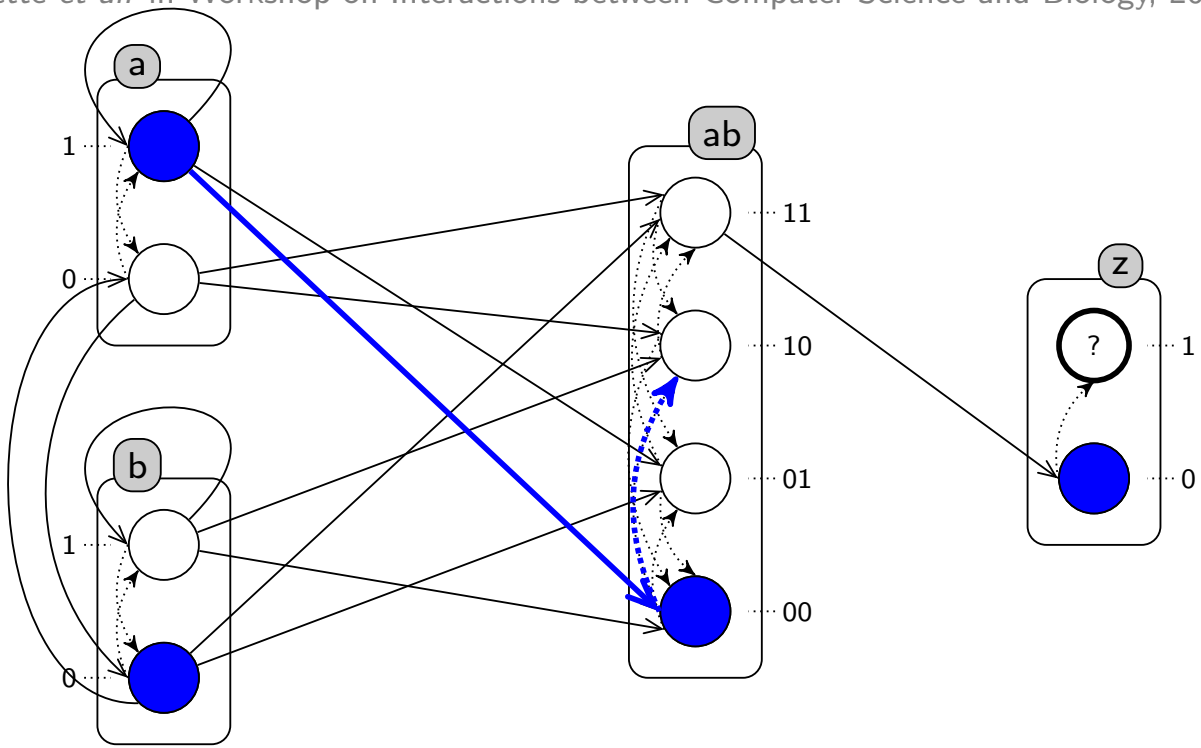


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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

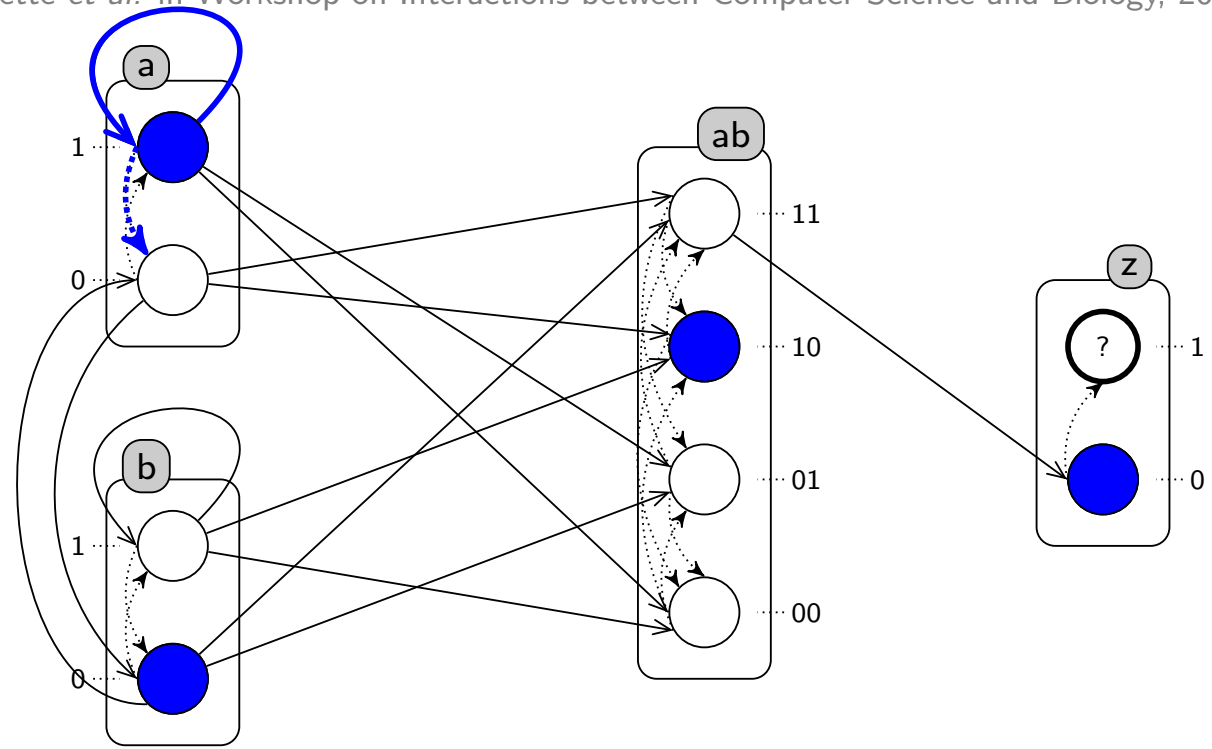


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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

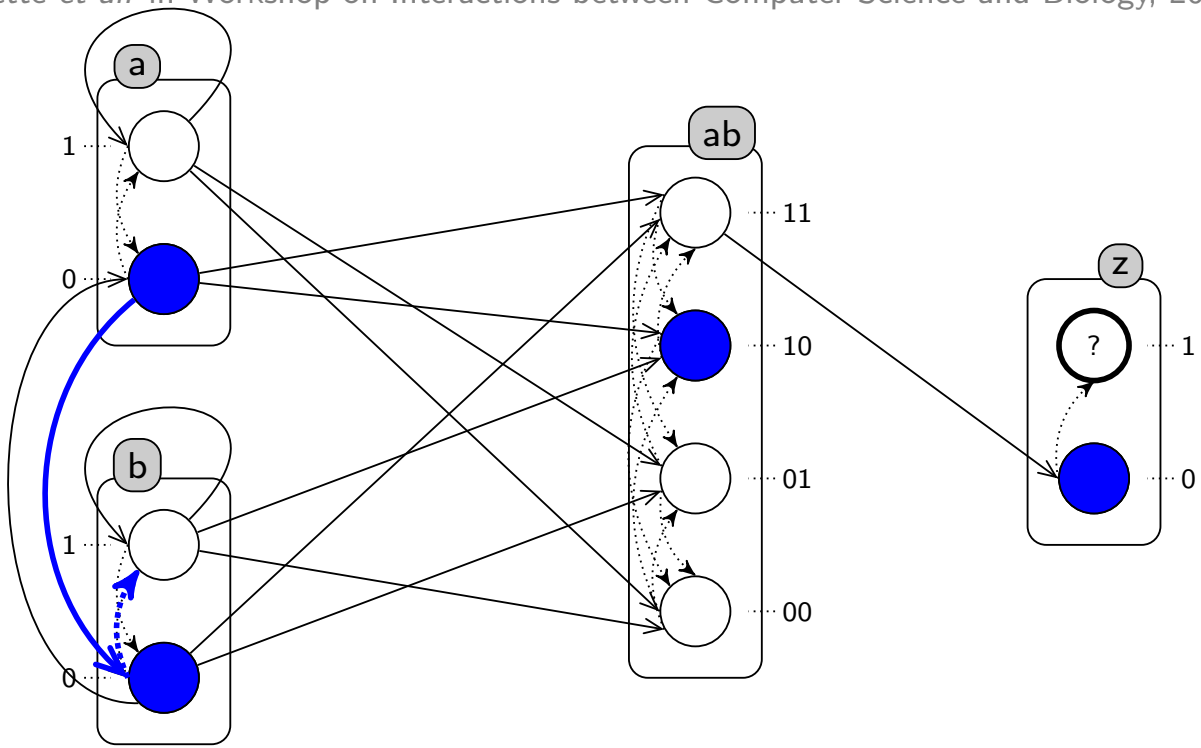


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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

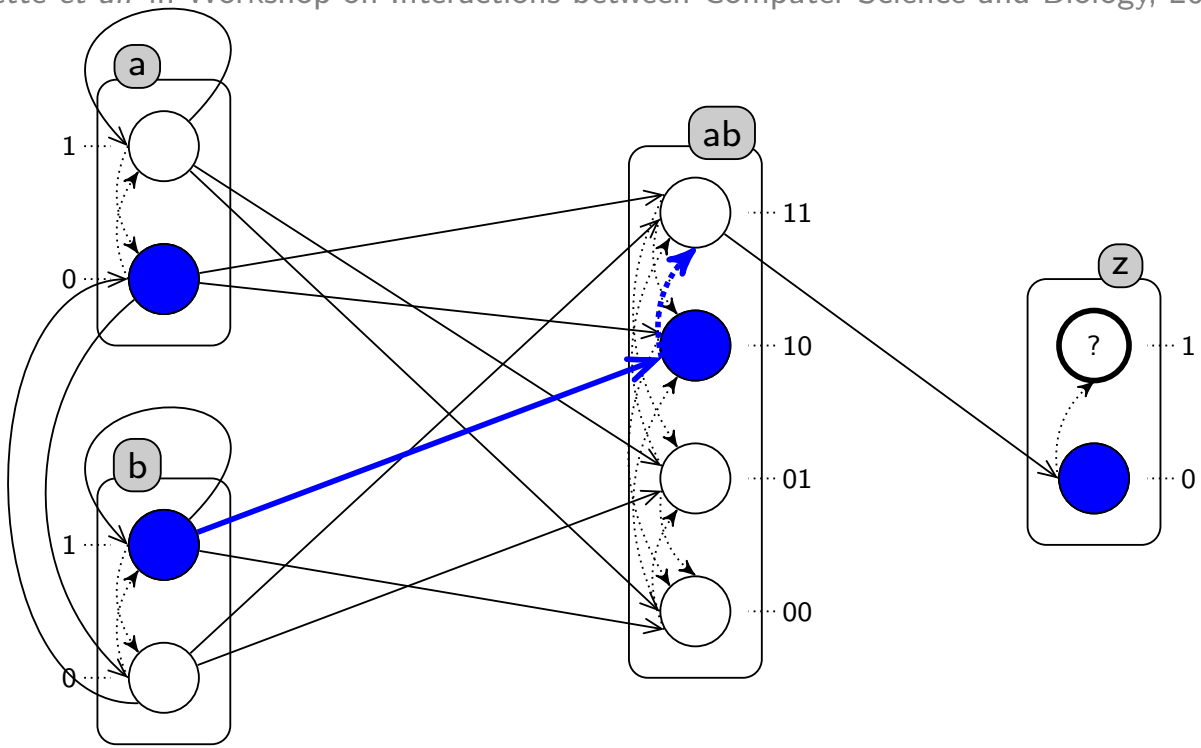


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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

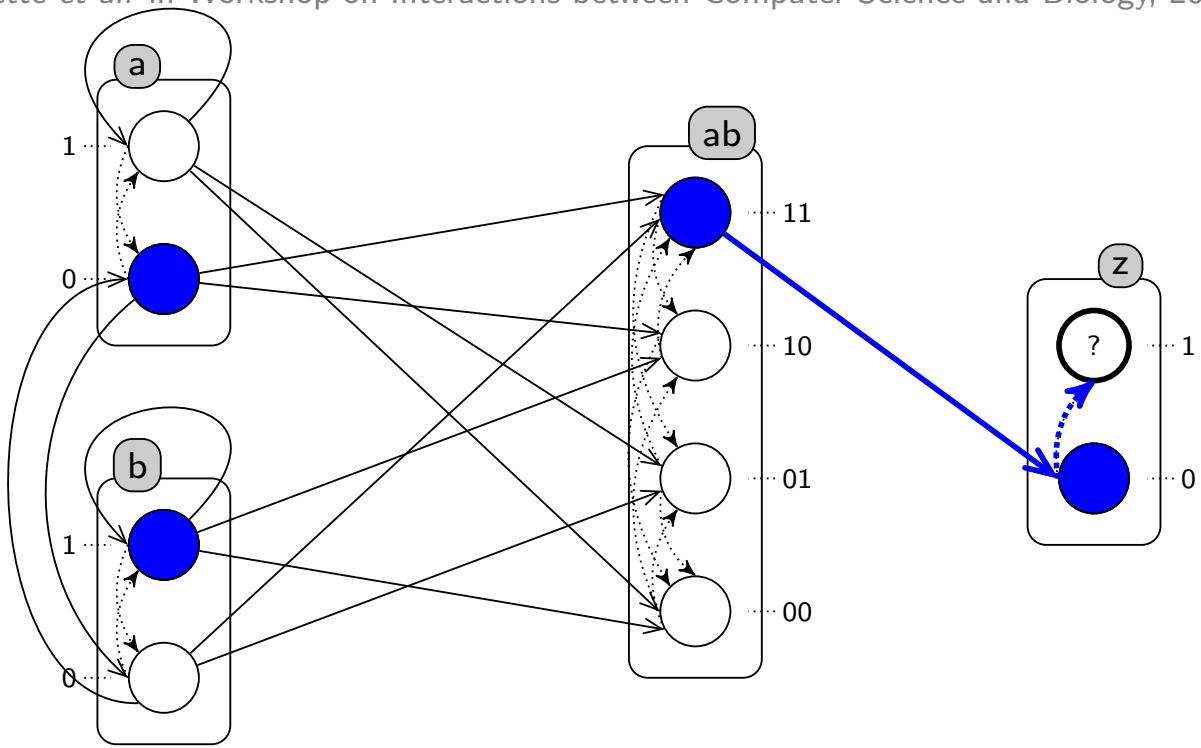


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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

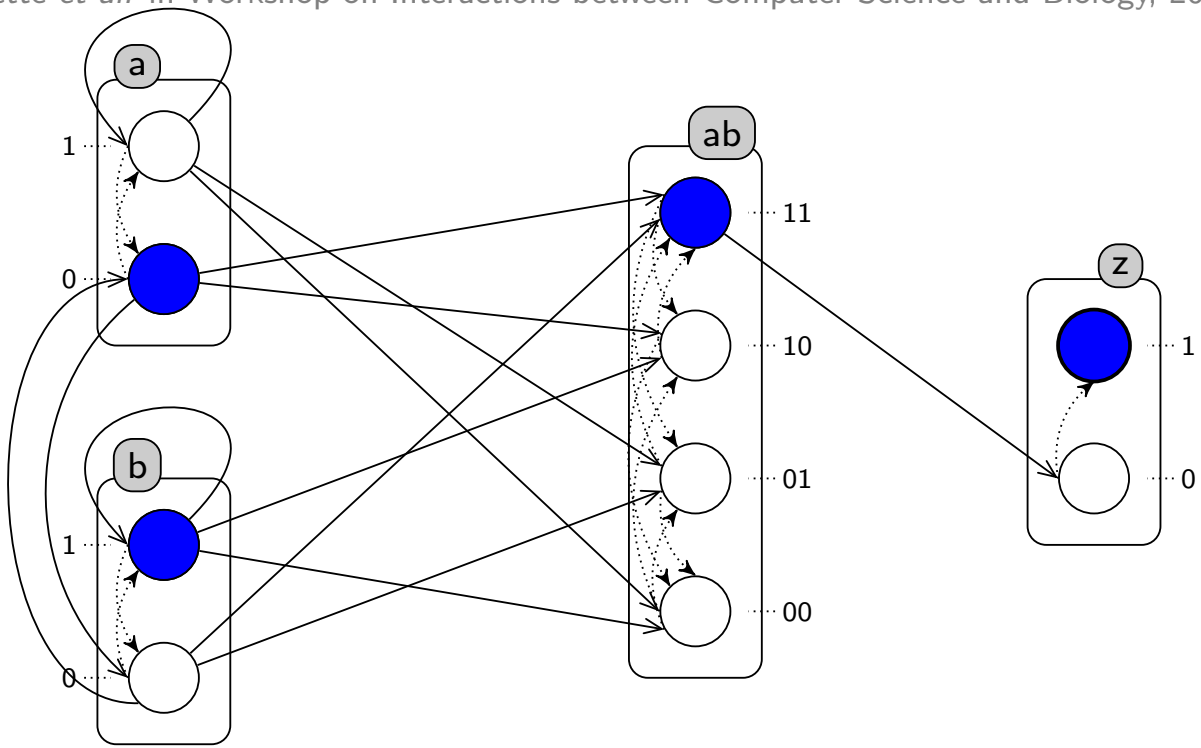


Drawback: Cooperations are too "loose" to be as expressive as ADN.

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

Adapting the expressivity of PH

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



Drawback: Cooperations are too "loose" to be as expressive as ADN.

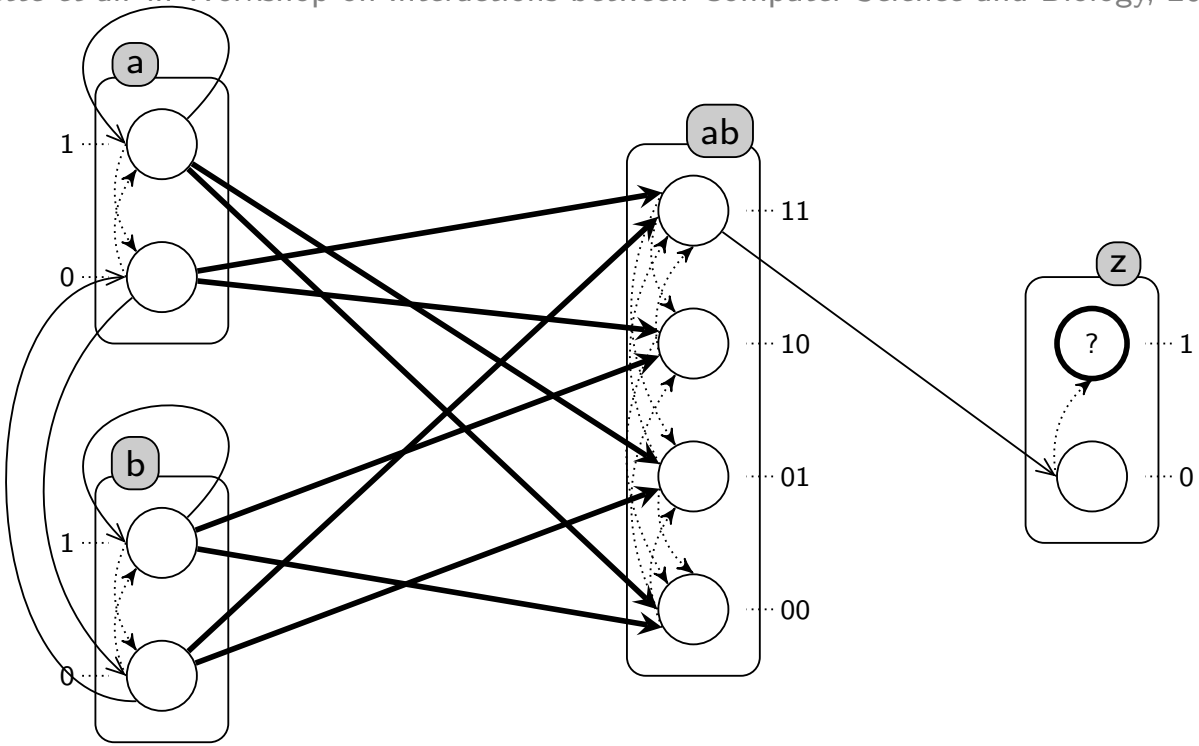
$\langle a_0, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{00}, z_0 \rangle \to \langle a_1, b_0, ab_{10}, z_0 \rangle \to \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\to \langle a_0, b_1, ab_{10}, z_0 \rangle \to \langle a_0, b_1, ab_{11}, z_0 \rangle \to \langle a_0, b_1, ab_{11}, z_1 \rangle$

The cooperativity should be: $a_1 \wedge b_1$ **simultaneously** *i.e.* "in the same state"

but the model behaves like: $\mathbf{P}(a_1) \wedge \mathbf{P}(b_1)$ with $\mathbf{P}$ = "previously"

Adapting the expressivity of PH

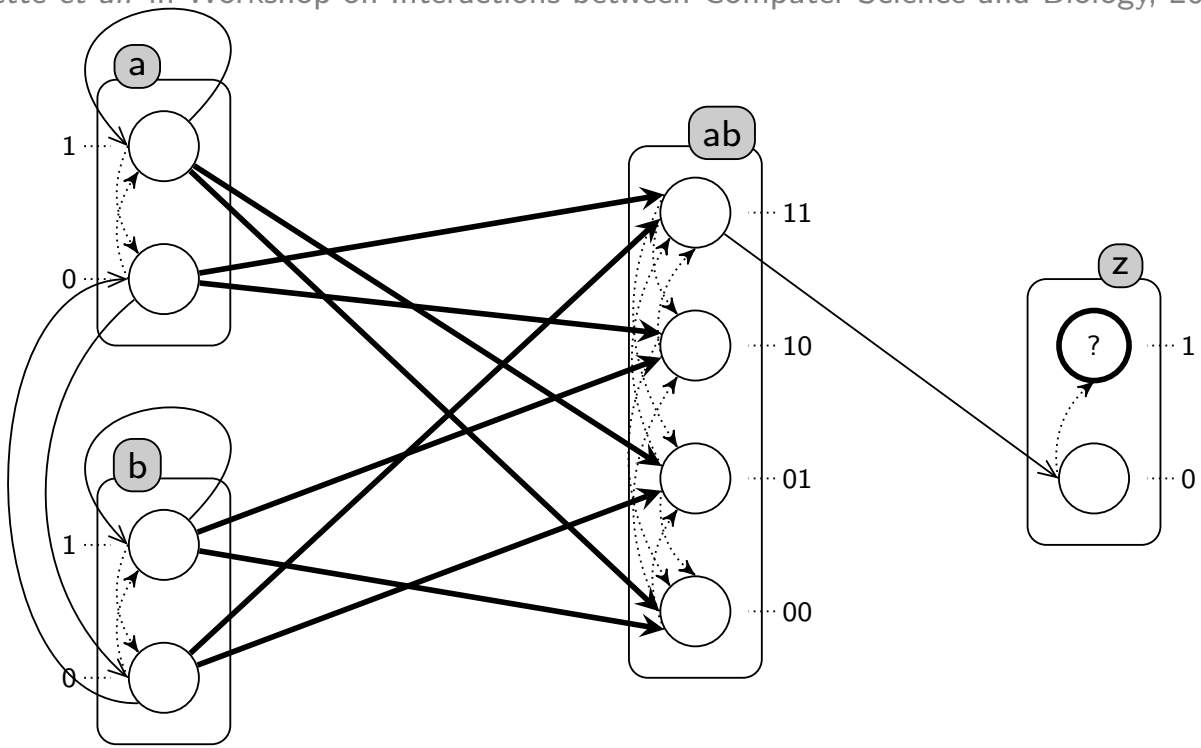
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]



- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

Adapting the expressivity of PH

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

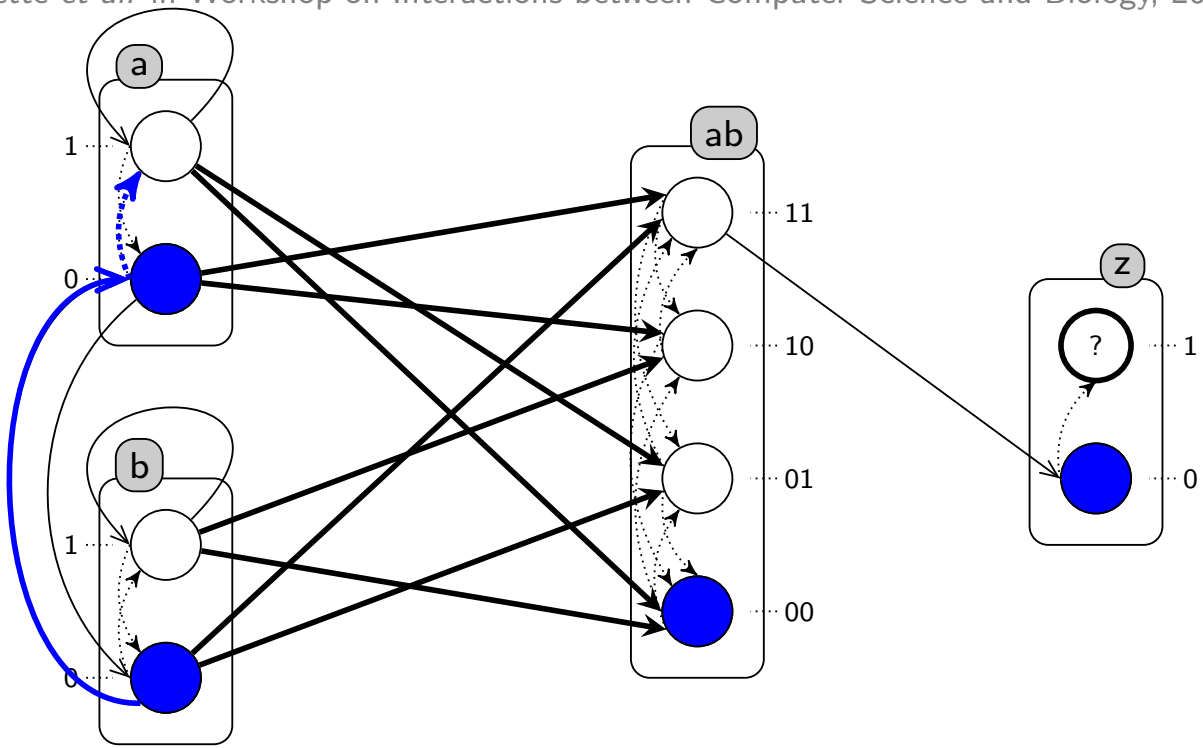


- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

Adapting the expressivity of PH

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



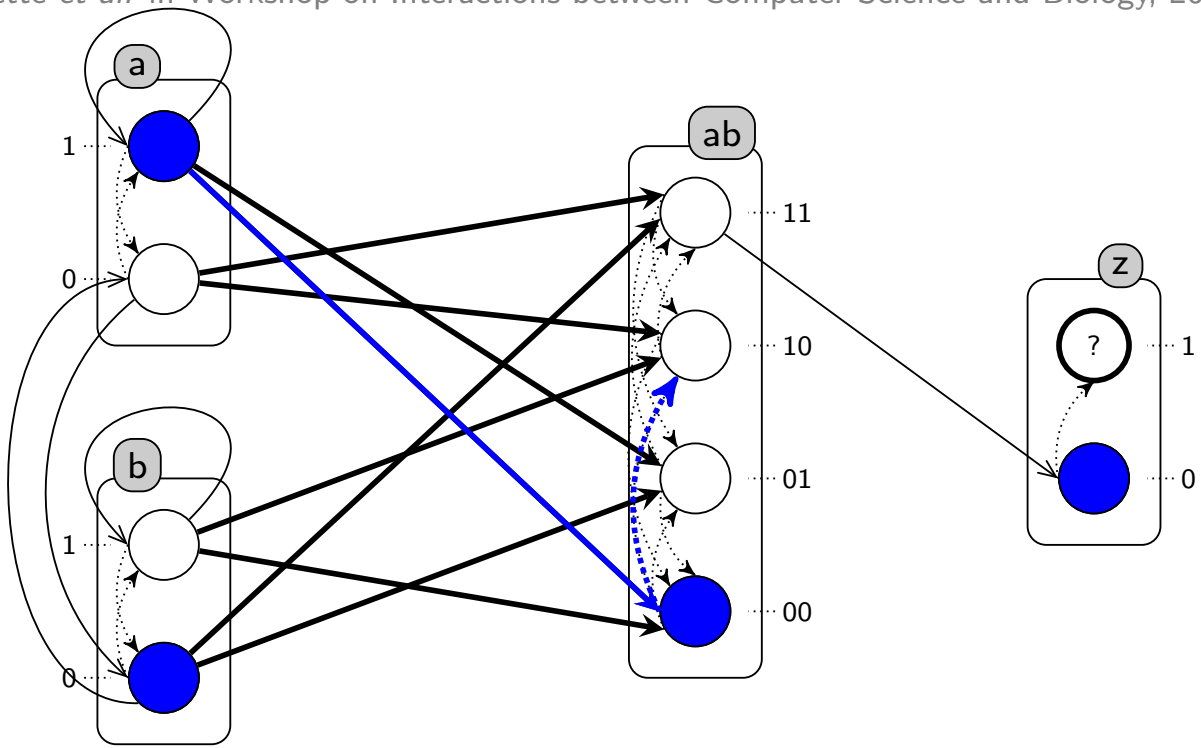
- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

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

Adapting the expressivity of PH

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



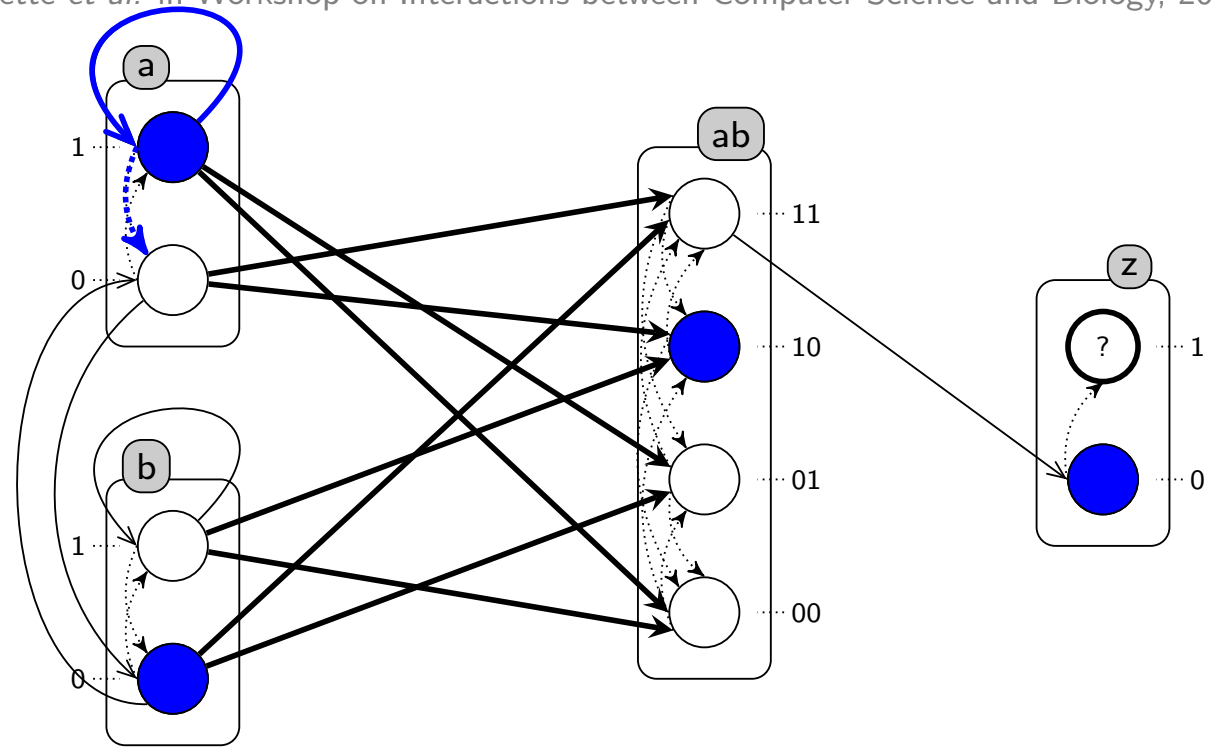
- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

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

Adapting the expressivity of PH

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



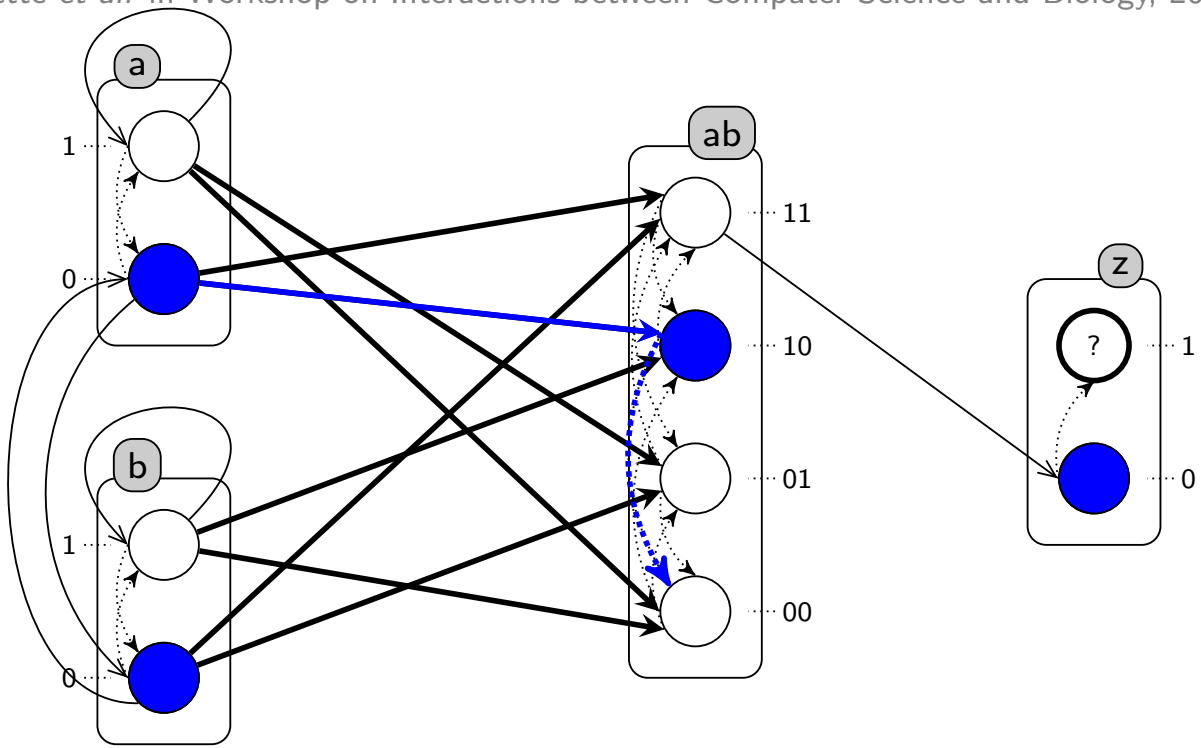
- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

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

Adapting the expressivity of PH

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



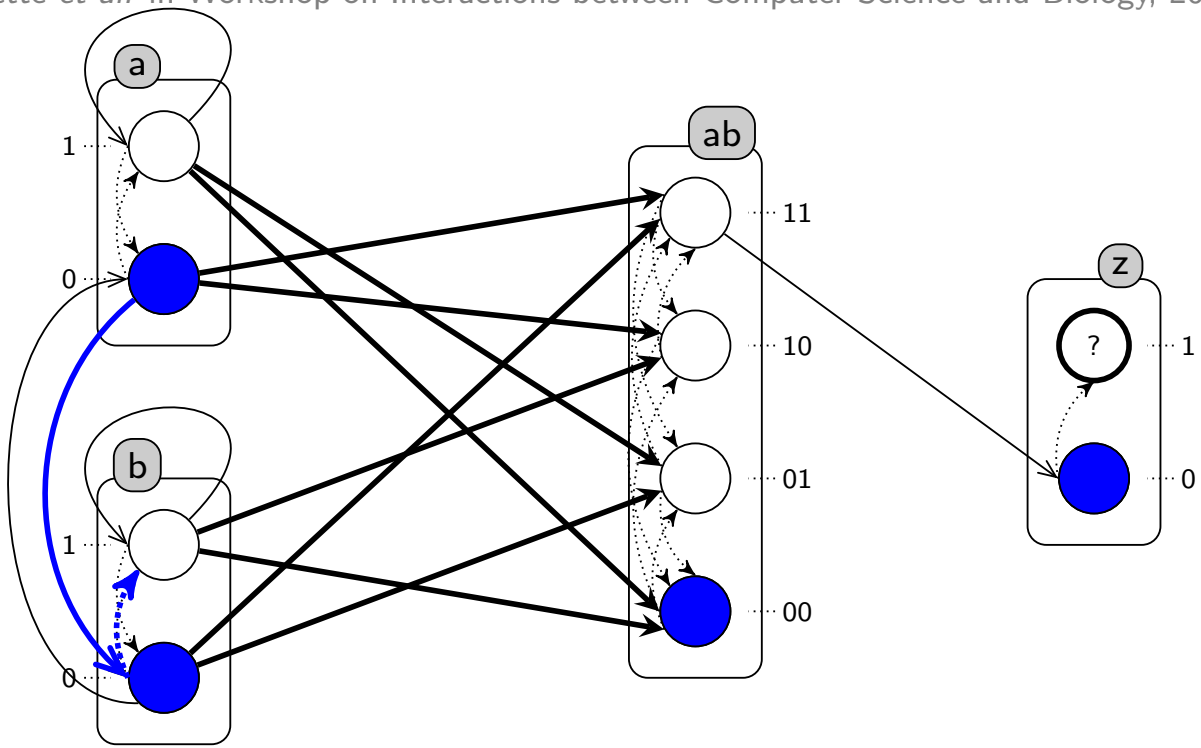
- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

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

Adapting the expressivity of PH

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



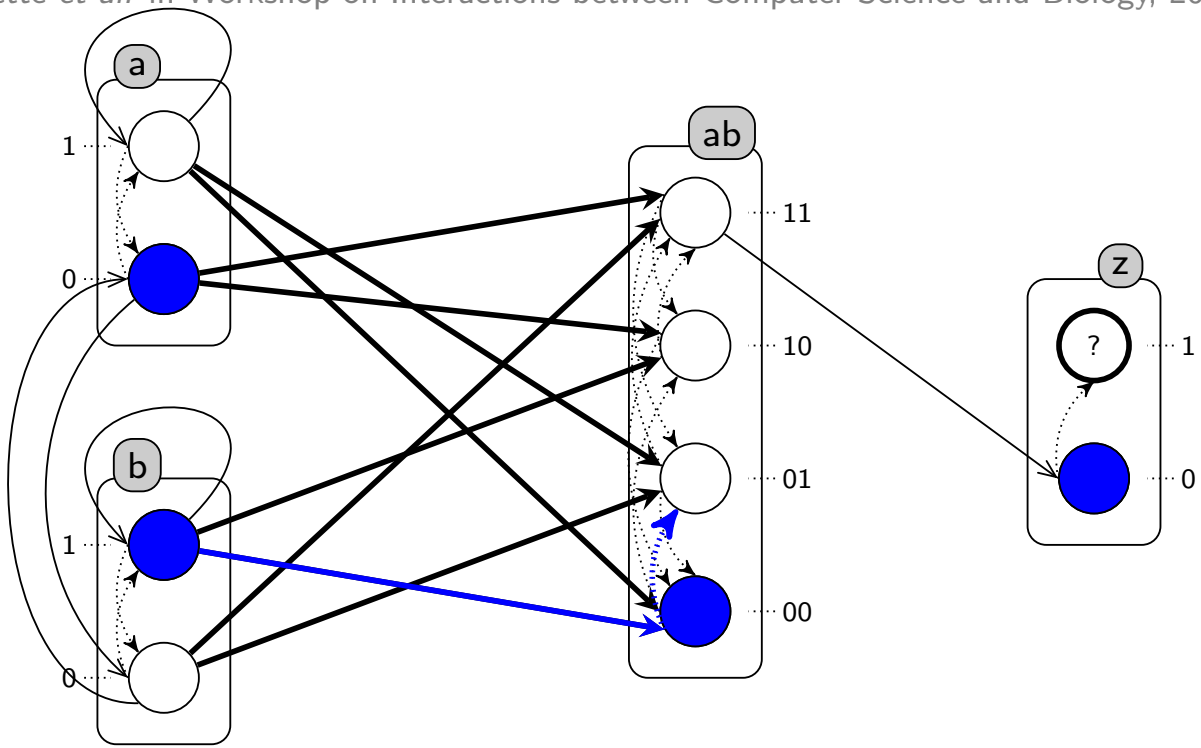
- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

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

Adapting the expressivity of PH

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



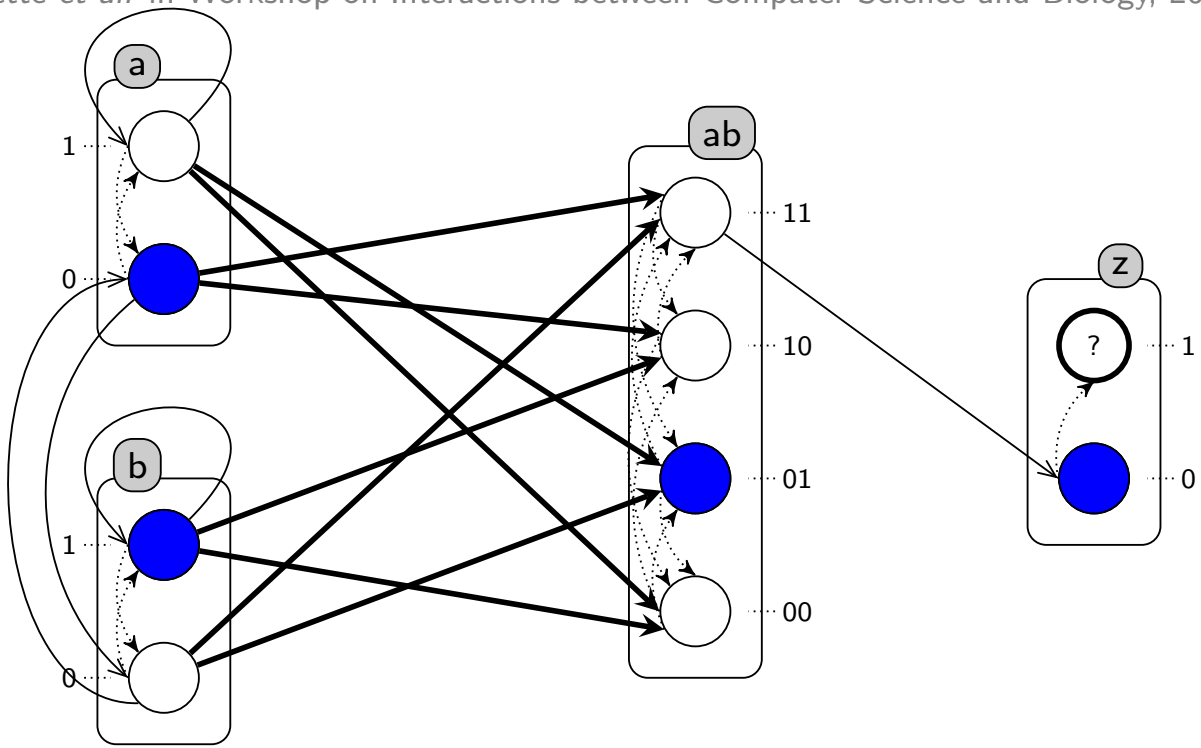
- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

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

Adapting the expressivity of PH

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



- Prioritise actions updating cooperative sorts (non-biological actions)
- All other actions remain unprioritised (evolutions with delays)

$\Rightarrow$ Whenever a regular action is played, all cooperative sorts are already updated

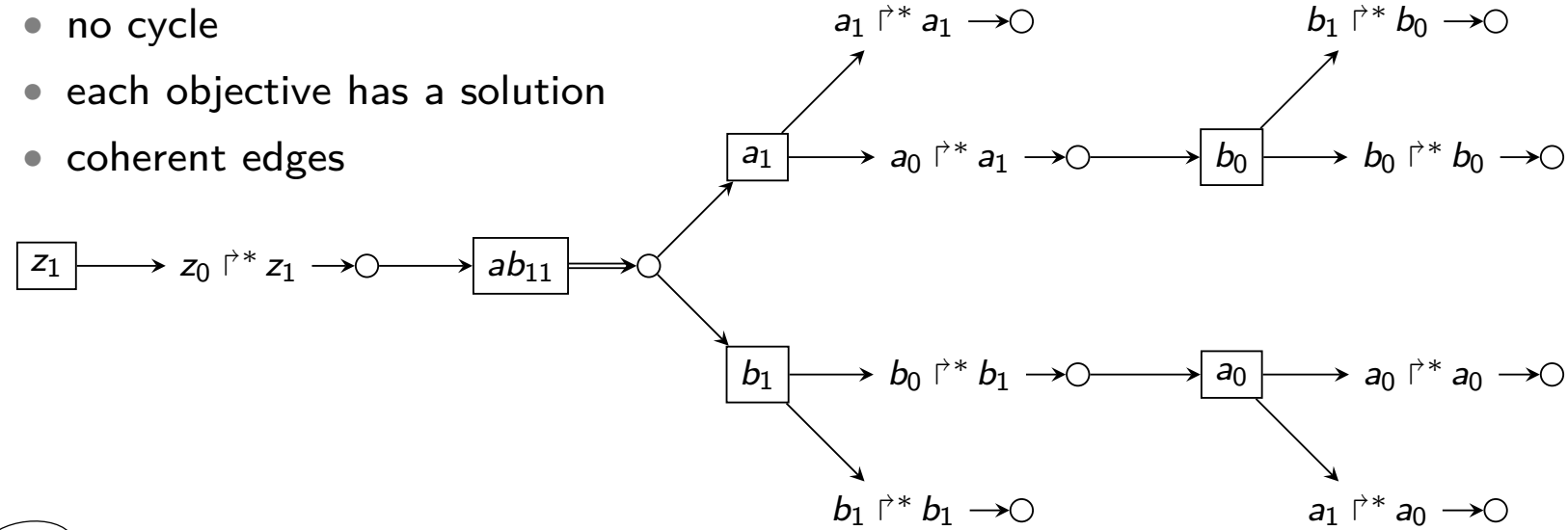
$\langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_1, b_0, ab_{10}, z_0 \rangle \rightarrow \langle a_0, b_0, ab_{10}, z_0 \rangle$
$\rightarrow \langle a_0, b_0, ab_{00}, z_0 \rangle \rightarrow \langle a_0, b_1, ab_{00}, z_0 \rangle \rightarrow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

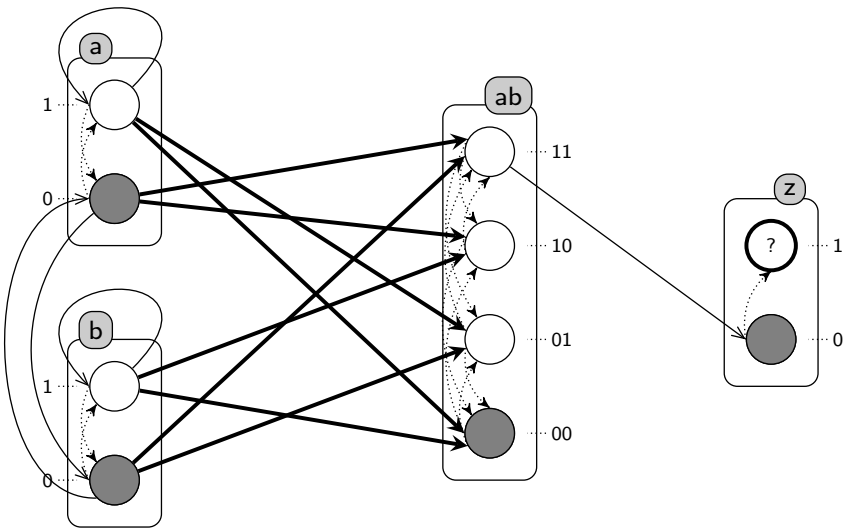
Static analysis with prioritised actions

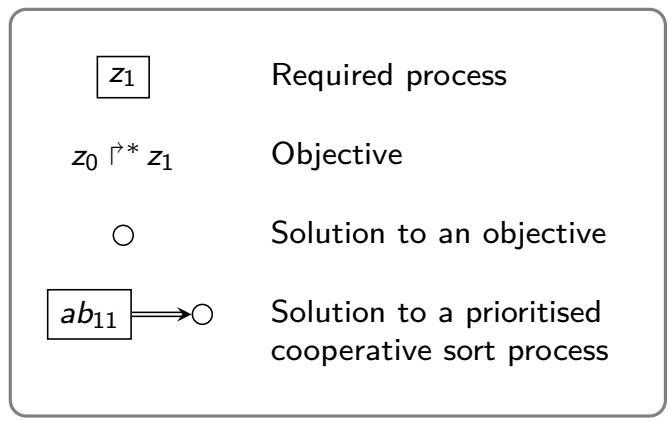
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- coherent edges





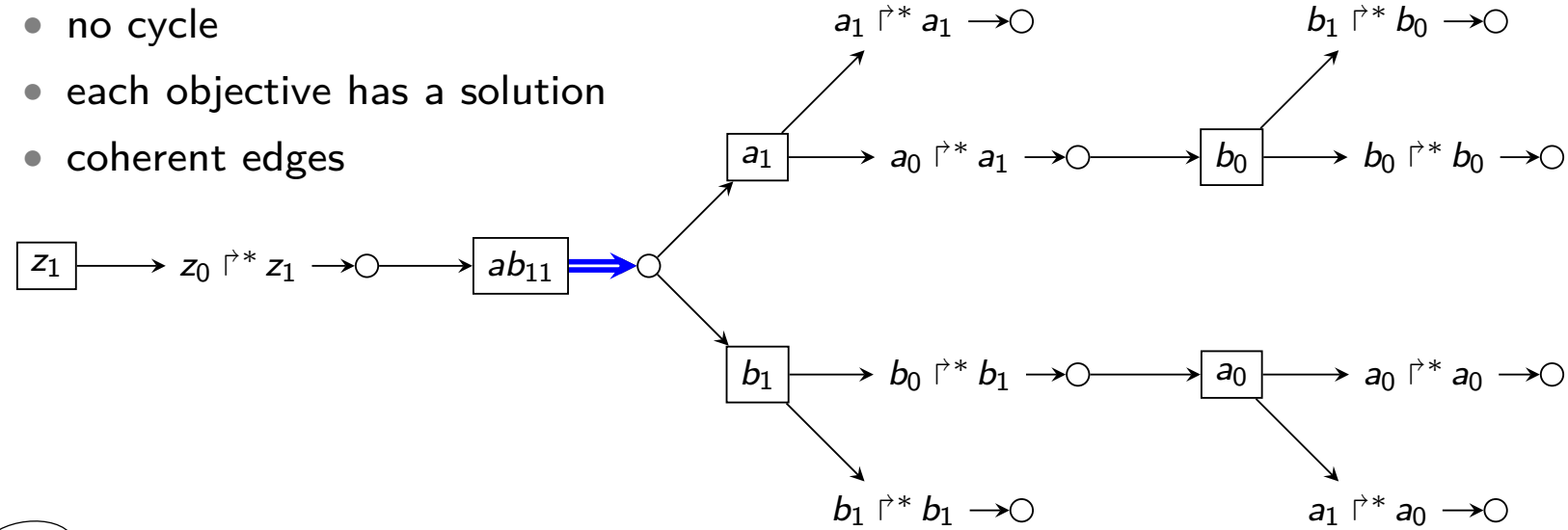


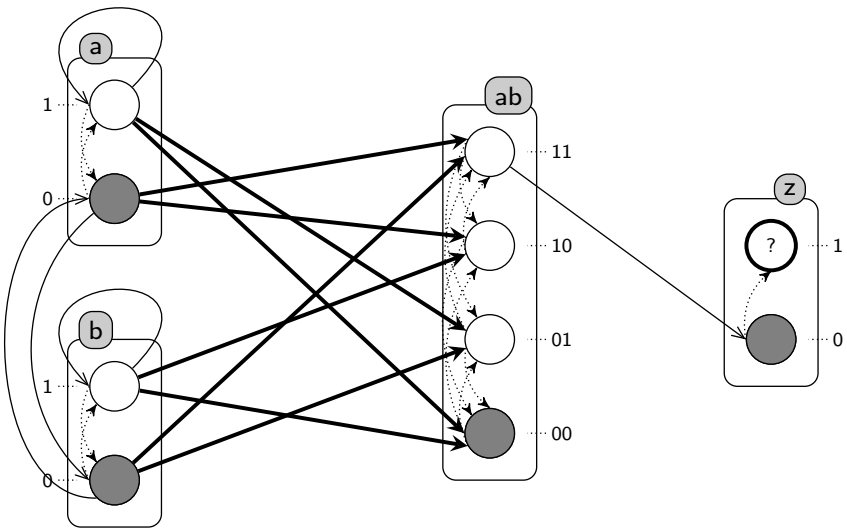
Static analysis with prioritised actions

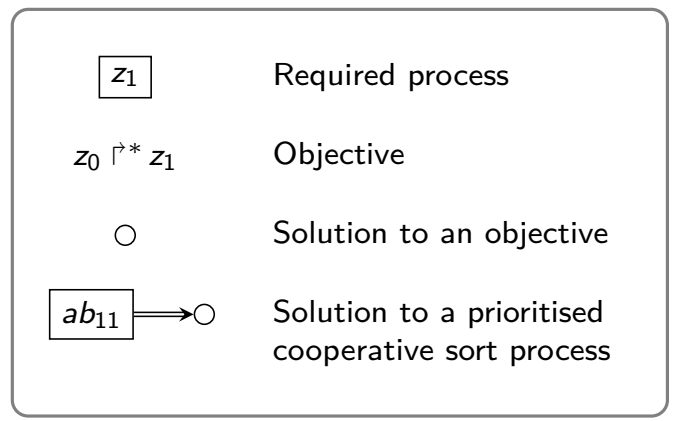
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- coherent edges





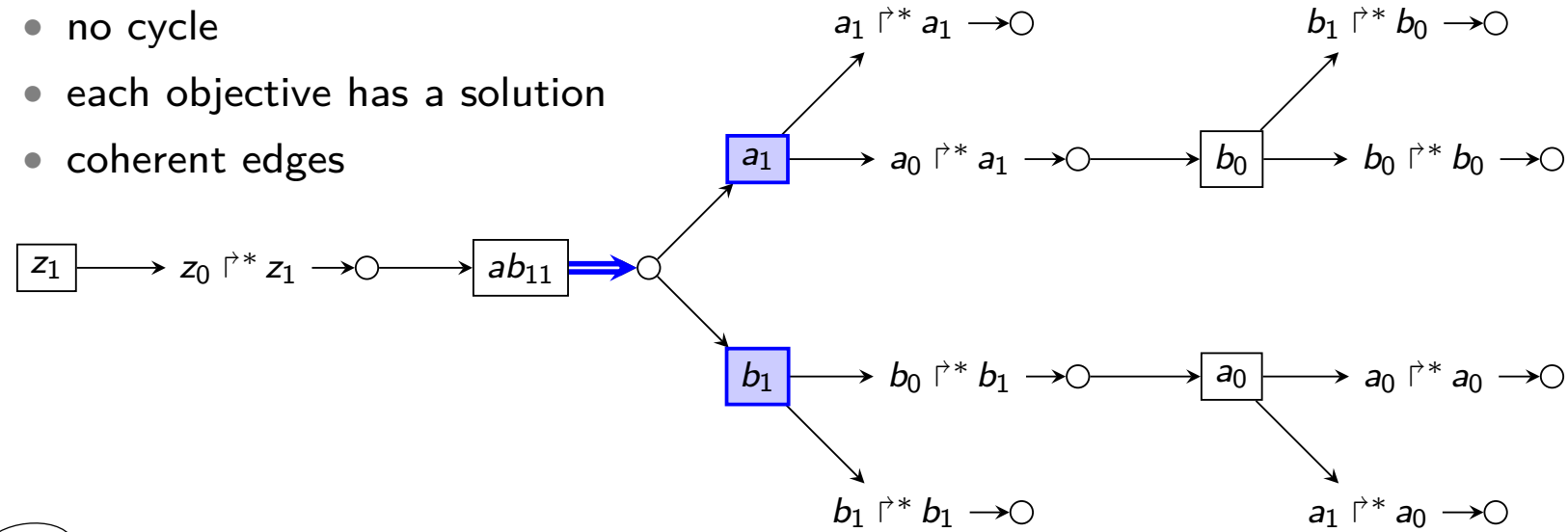


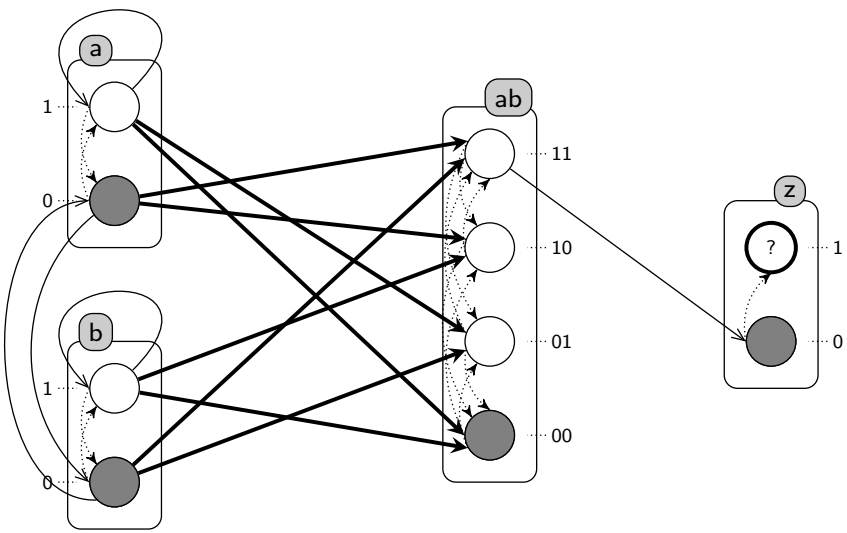
Static analysis with prioritised actions

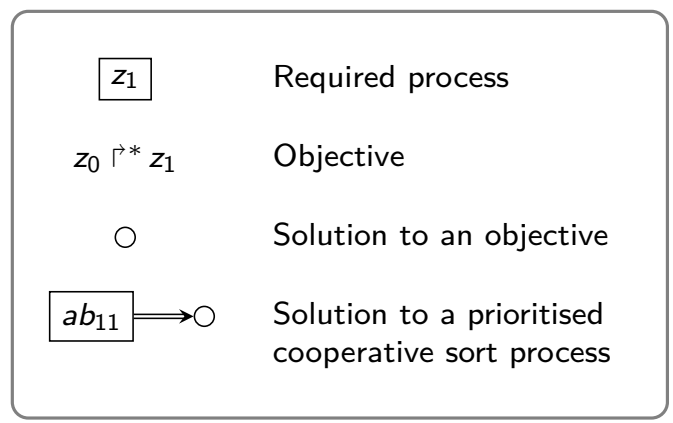
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- coherent edges





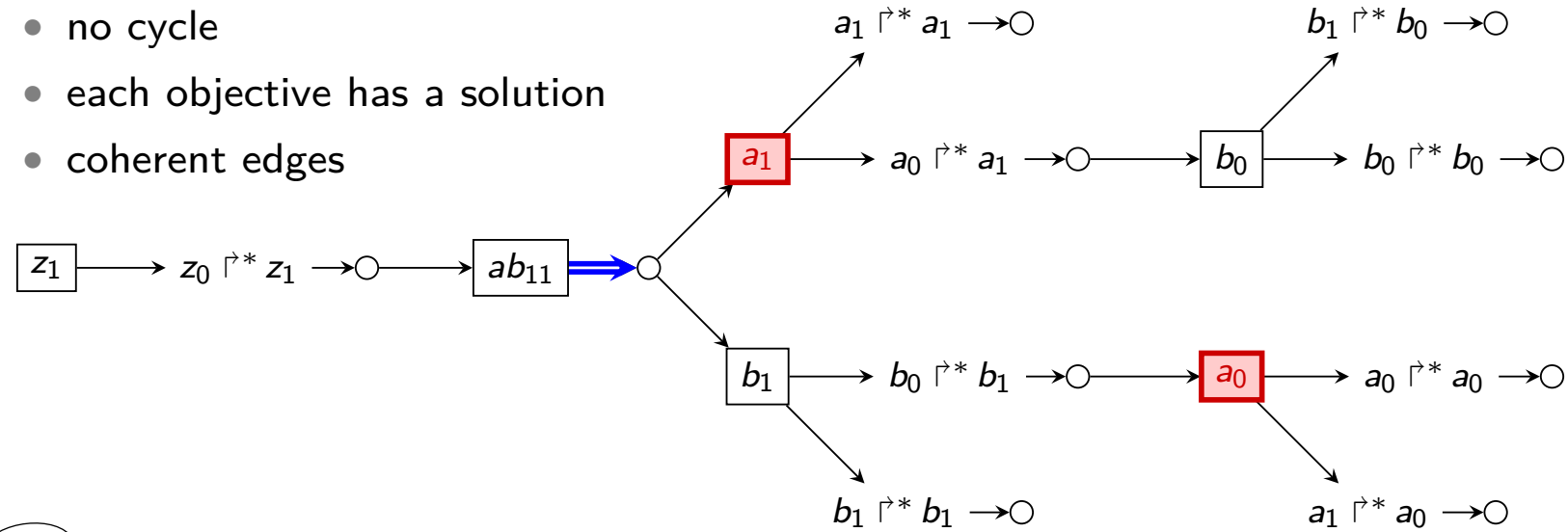


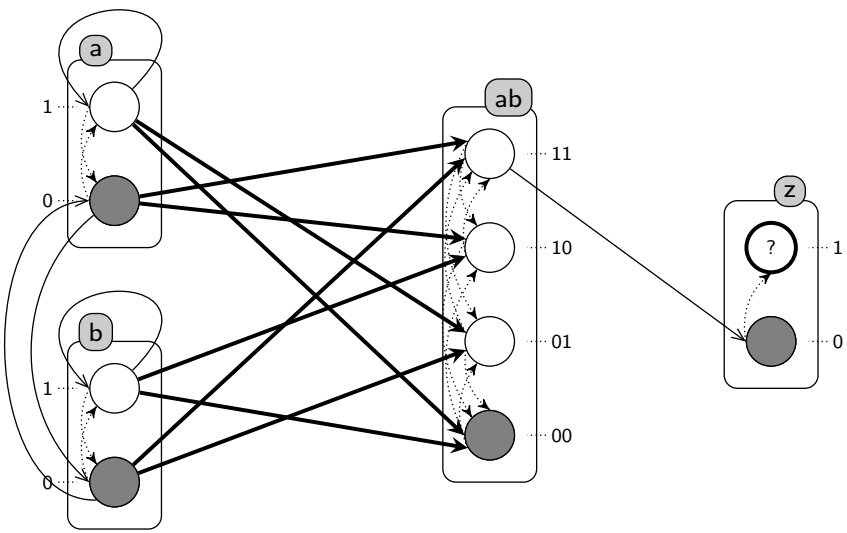
Static analysis with prioritised actions

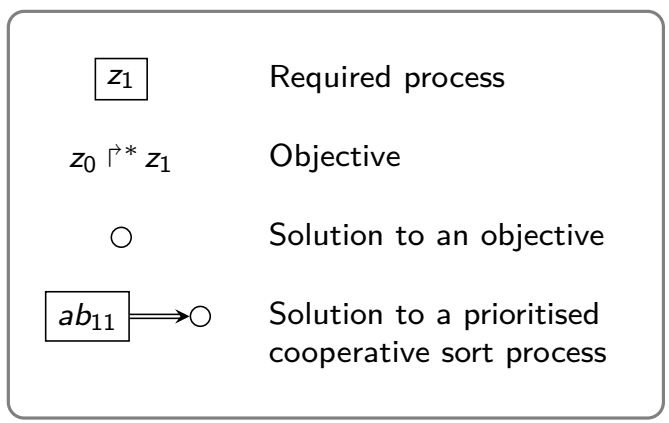
[Folschette *et al.* in Workshop on Interactions between Computer Science and Biology, 2013]

Sufficient condition:

- no cycle
- each objective has a solution
- coherent edges





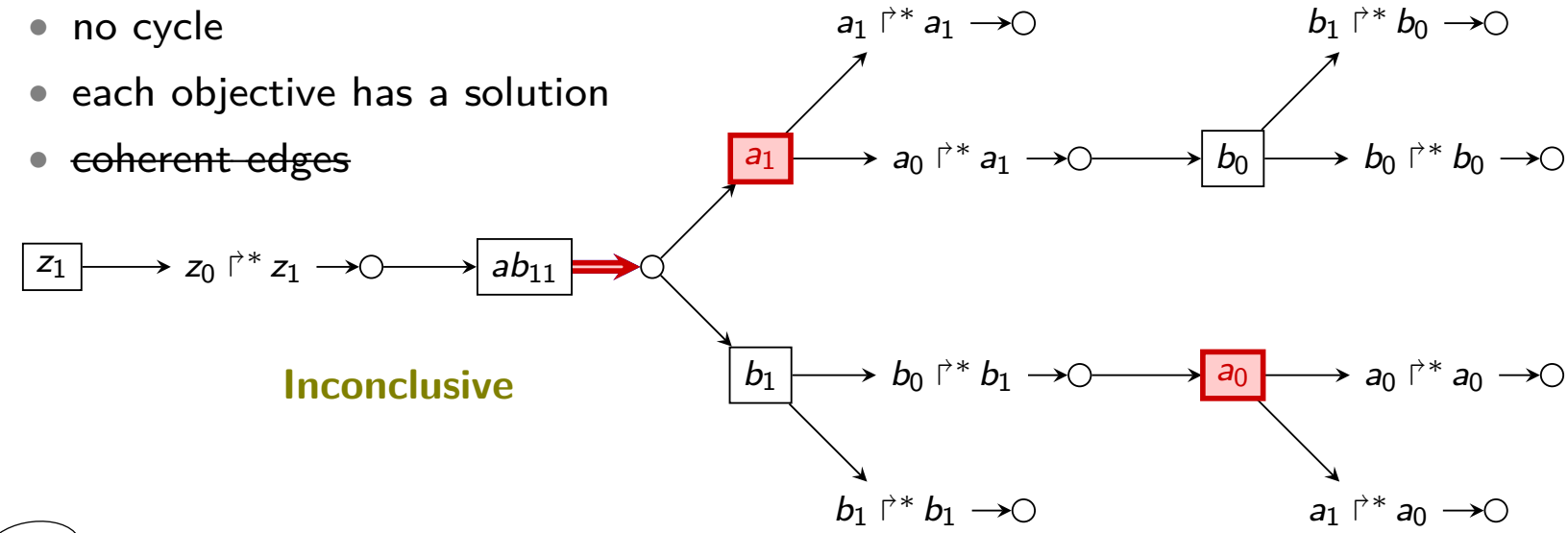


Static analysis with prioritised actions

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

Sufficient condition:

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



Inconclusive