MOVEP 2014

Efficient analysis on very large models

Maxime FOLSCHETTE

MeForBio / IRCCyN / École Centrale de Nantes (Nantes, France)

`maxime.folschette@irccyn.ec-nantes.fr`
`http://maxime.folschette.name/`

Context and Aims

MeForBio team:
Qualitative modelling to study
large dynamical biological systems

Context and Aims

MeForBio team:
Qualitative modelling to study
large dynamical biological systems

1) The object: **Gene regulations**

   Large discrete models to study gene interactions

Context and Aims

MeForBio team:
Qualitative modelling to study
large dynamical biological systems

1) The object: **Gene regulations**

   Large discrete models to study gene interactions

2) The method: **Static analysis**

   Efficient methods thanks to the **Process Hitting** framework

Context and Aims

MeForBio team:
Qualitative modelling to study
large dynamical biological systems

1) The object: **Gene regulations**

   Large discrete models to study gene interactions

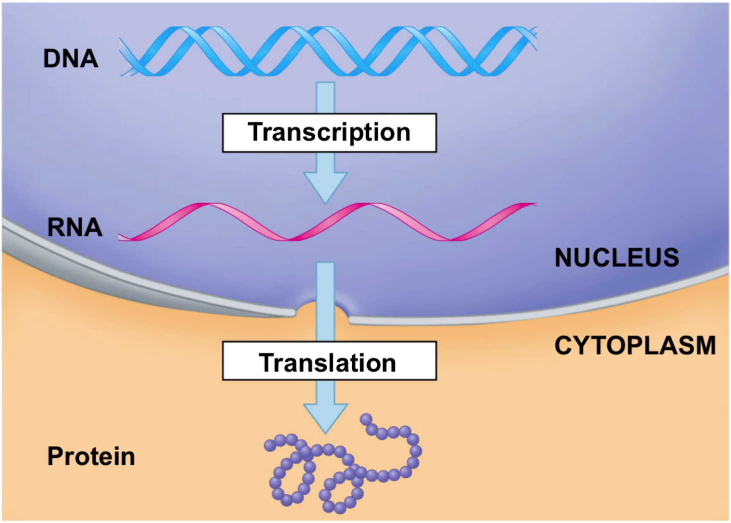
2) The method: **Static analysis**

   Efficient methods thanks to the **Process Hitting** framework

3) The result: **Applications**

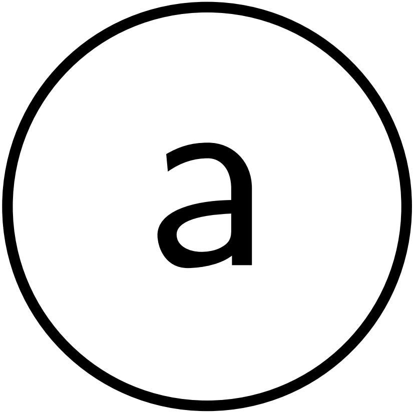
   The example of **gene therapies**

Gene regulations



© 2012 Pearson Education, Inc.

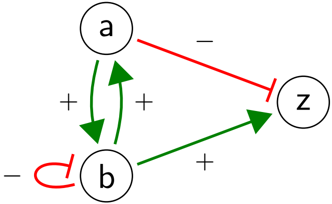
Gene regulations



Usual biological algebraic models

[De Jong, *Journal of Computational Biology*, 2002]

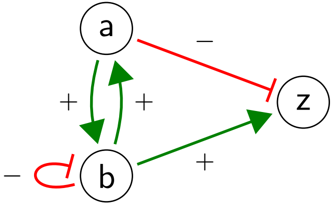
Modelling interacting genes/proteins: **Boolean Networks**



Usual biological algebraic models

[De Jong, *Journal of Computational Biology*, 2002]

Modelling interacting genes/proteins: **Boolean Networks**

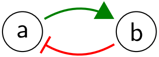


Questions:

- How does z **behave**?
- Is it **possible** to make a inactive?
- If I **knock-out** b, what changes?

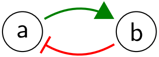
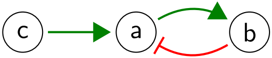
The combinatorial explosion

→ Problem: easy to understand but hard to study
exponential number of states

Model	**Possible states**
	4

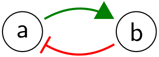
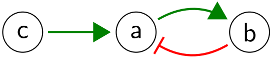
The combinatorial explosion

→ Problem: easy to understand but hard to study
exponential number of states

Model	Possible states
	4
	8

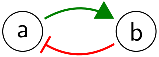
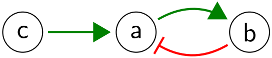
The combinatorial explosion

→ Problem: easy to understand but hard to study
exponential number of states

Model	**Possible states**
	4
	8
⋮	⋮
(10)	1024

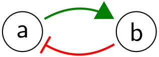
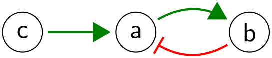
The combinatorial explosion

→ Problem: easy to understand but hard to study
exponential number of states

Model	Possible states
	4
	8
⋮	⋮
(10)	1024
(20)	1048576

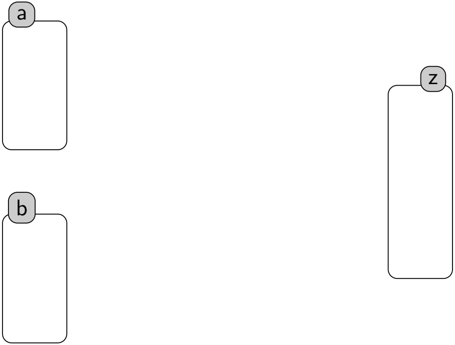
The combinatorial explosion

→ Problem: easy to understand but hard to study
exponential number of states

Model	Possible states
	4
	8
⋮	⋮
(10)	1024
(20)	1048576
(100)	1267650600000000000000000000000

The Process Hitting modelling

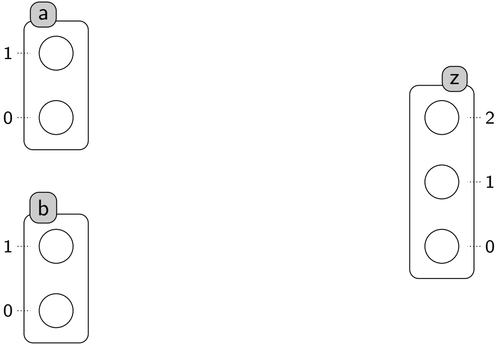
[Paulevé *et al.*, *Transactions on Computational Systems Biology*, 2011]



Sorts: components *a*, *b*, *z*

The Process Hitting modelling

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

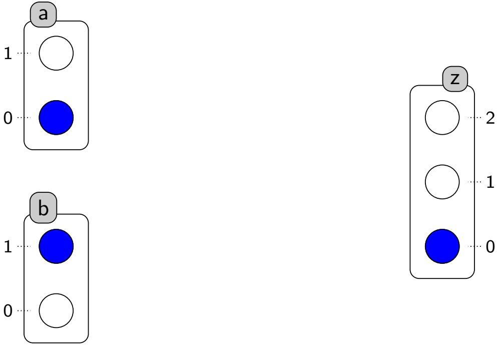


Sorts: components a, b, z

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

The Process Hitting modelling

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



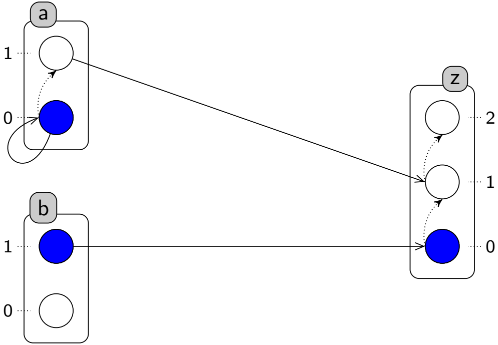
Sorts: components a, b, z

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 modelling

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



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

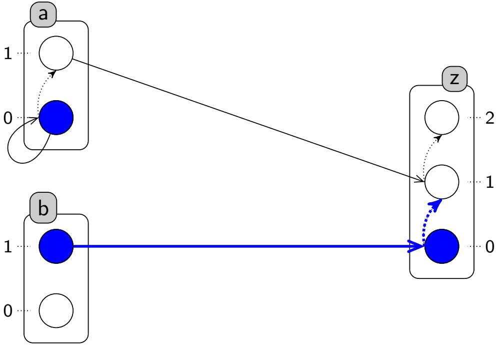
Processes: local states / levels of expression $z_0,\ z_1,\ z_2$

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

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

The Process Hitting modelling

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



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

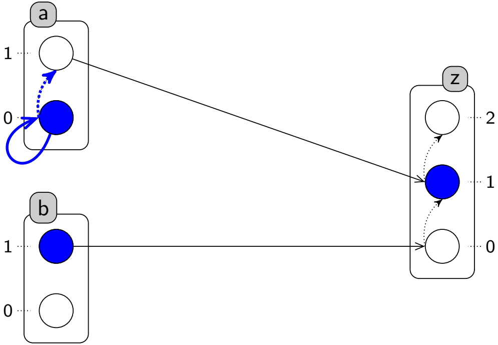
Processes: local states / levels of expression $z_0,\ z_1,\ z_2$

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

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

The Process Hitting modelling

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



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

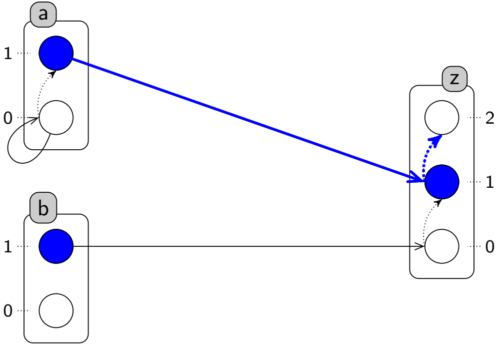
Processes: local states / levels of expression $z_0,\ z_1,\ z_2$

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

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

The Process Hitting modelling

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



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

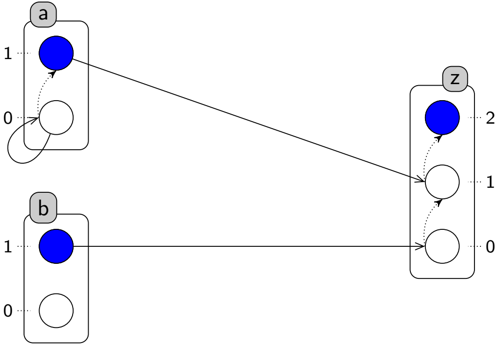
Processes: local states / levels of expression $z_0,\ z_1,\ z_2$

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

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

The Process Hitting modelling

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



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

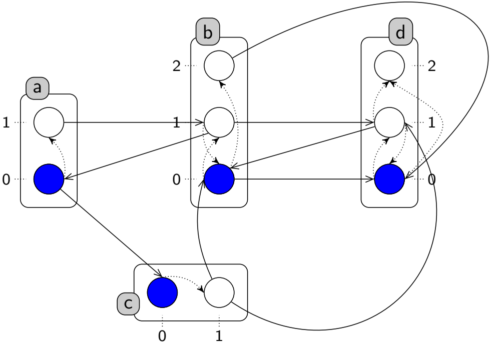
Processes: local states / levels of expression $z_0,\ z_1,\ z_2$

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

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

Static analysis: successive reachability

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

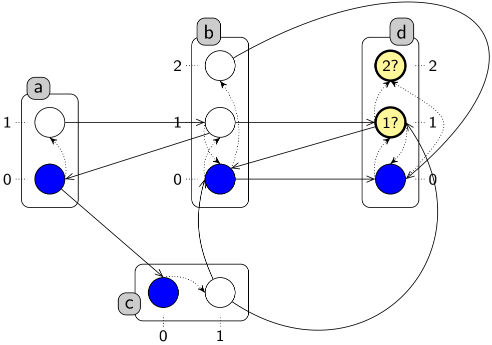


- Initial state

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

Static analysis: successive reachability

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



- Initial state

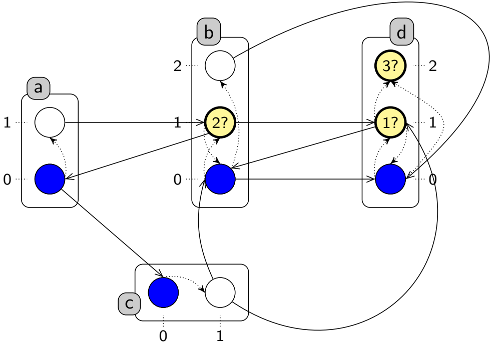
$$\langle a_1, b_0, c_0, d_0 \rangle$$

- Objectives

$$[\ \upharpoonright d_1\ ::\ \upharpoonright d_2\ ]$$

Static analysis: successive reachability

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



- Initial state

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

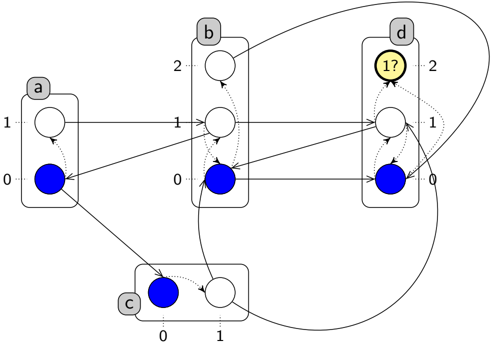
- Objectives

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

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

Static analysis: successive reachability

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



- Initial state

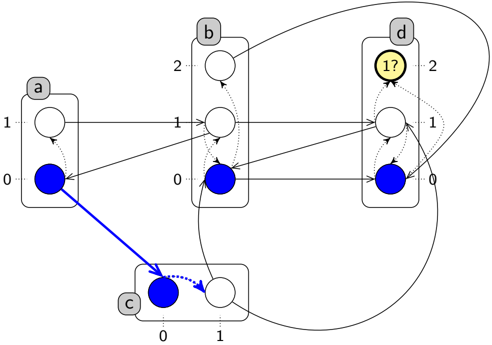
$$\langle a_1, b_0, c_0, d_0 \rangle$$

- 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.*, *Mathematical Structures in Computer Science*, 2012]



- Initial state

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

- Objectives

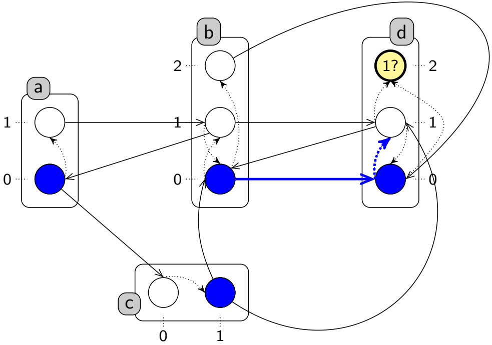
$$[\ \Rsh d_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_2\ ]$$

→ Concretization of the objective = scenario

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

Static analysis: successive reachability

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



- Initial state

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

- Objectives

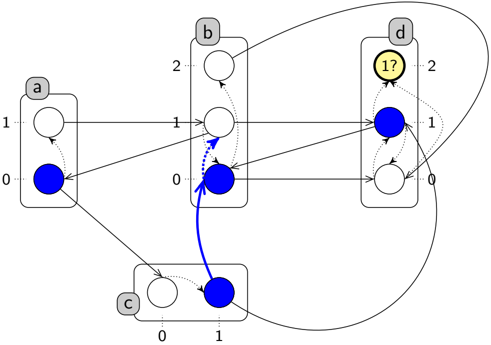
$$[\ \Rsh d_1 \ :: \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_2\ ]$$

$\rightarrow$ 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.*, *Mathematical Structures in Computer Science*, 2012]



- Initial state

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

- Objectives

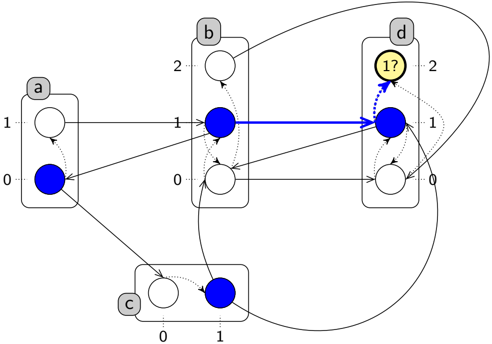
$$[\ \Rsh d_1 \ :: \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_2\ ]$$

$\rightarrow$ 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.*, *Mathematical Structures in Computer Science*, 2012]



- Initial state

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

- Objectives

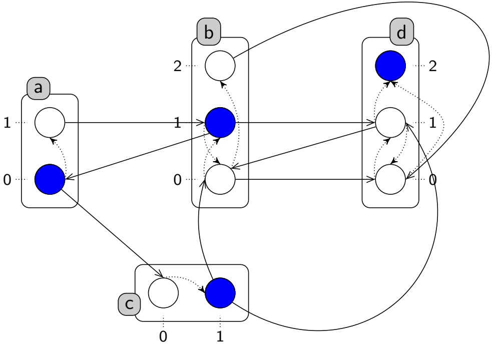
$$[\ \Rsh d_1\ :: \Rsh d_2\ ]$$
$$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$$
$$[\ \Rsh d_2\ ]$$

→ Concretization of the objective = scenario

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

Static analysis: successive reachability

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



- 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

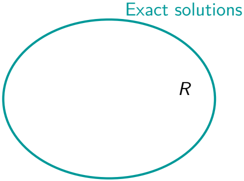
$$a_0 \rightarrow c_0 \Rsh c_1 :: b_0 \rightarrow d_0 \Rsh d_1 :: c_1 \rightarrow b_0 \Rsh b_1 :: b_1 \rightarrow d_1 \Rsh d_2$$

Over- and Under-approximations

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

$\rightarrow$ Directly checking R is hard (**exponential**)

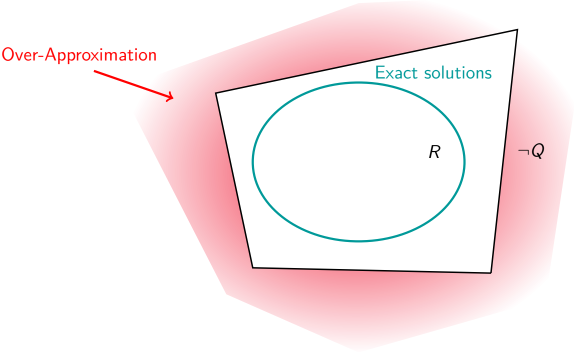
$\rightarrow$ Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$



Over- and Under-approximations

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

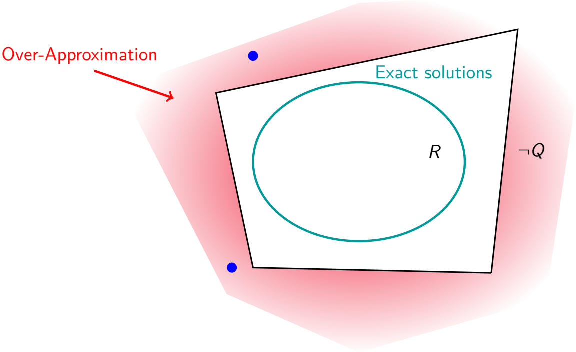
- → Directly checking R is hard (**exponential**)
- → Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$



Over- and Under-approximations

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

- → Directly checking R is hard (**exponential**)
- → Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$

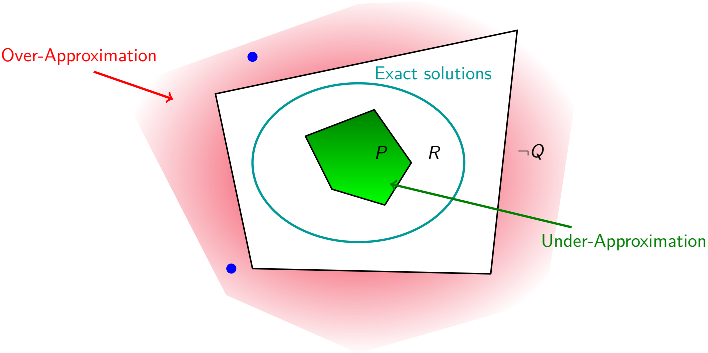


Over- and Under-approximations

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

→ Directly checking R is hard (**exponential**)

→ Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$

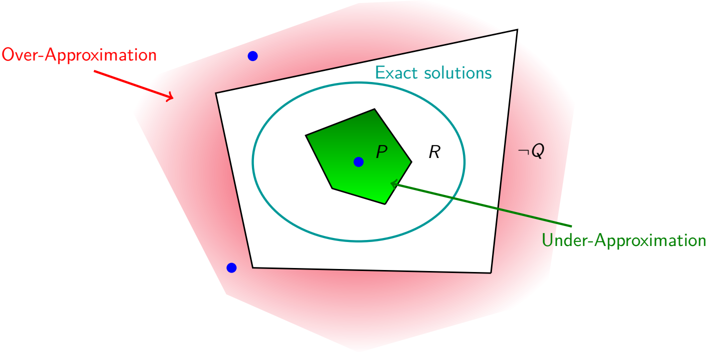


Over- and Under-approximations

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

→ Directly checking R is hard (**exponential**)

→ Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$

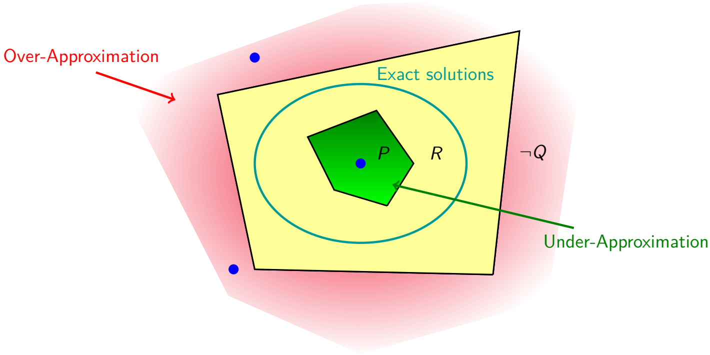


Over- and Under-approximations

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

→ Directly checking R is hard (**exponential**)

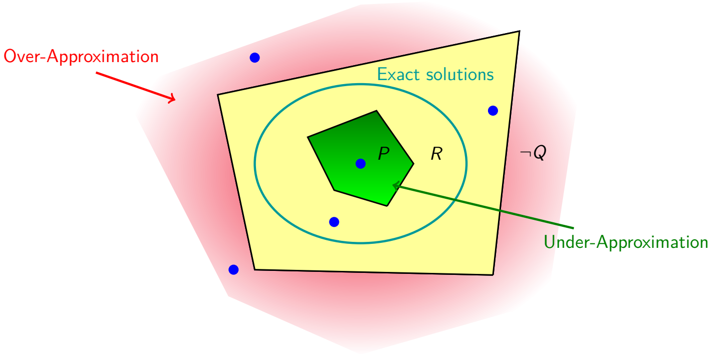
→ Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$



Over- and Under-approximations

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

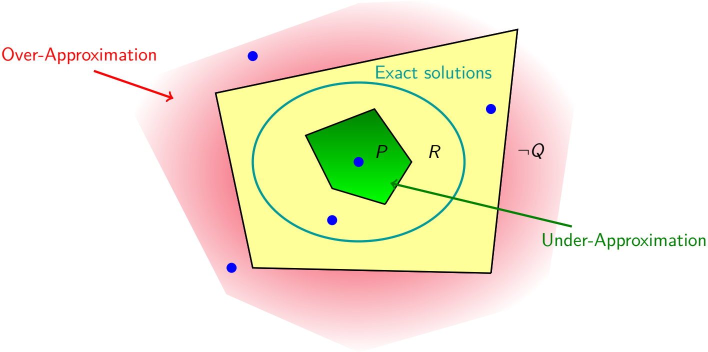
- → Directly checking R is hard (**exponential**)
- → Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$



Over- and Under-approximations

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

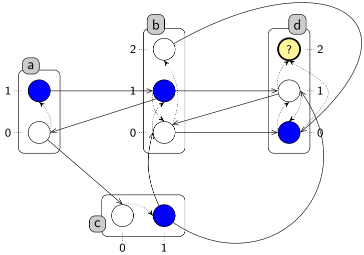
- → Directly checking R is hard (**exponential**)
- → Rather check **approximations** P and Q so that: $\underline{P \Rightarrow R \Rightarrow Q}$

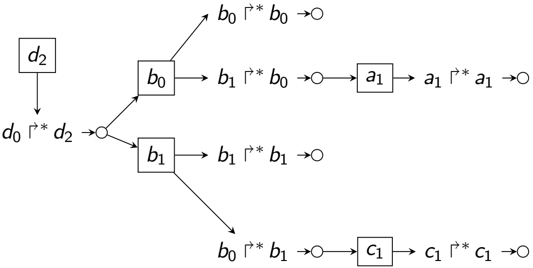


Computing P or Q is much simpler (roughly **polynomial**)

→ Efficient for big models → **Hundredths of seconds**

Under-approximation





$\boxed{d_2}$

$d_0 \Rsh^* d_2 \rightarrow ○$

$\boxed{b_0}$

$b_0 \Rsh^* b_0 \rightarrow ○$

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

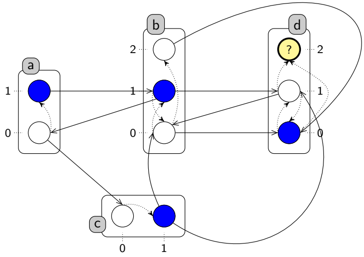
$\boxed{b_1}$

$b_1 \Rsh^* b_1 \rightarrow ○$

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

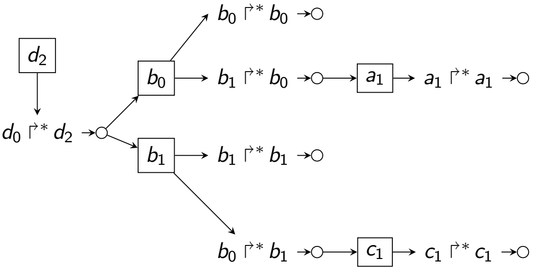
$\boxed{d_2}$	Required process
$d_0 \Rsh^* d_2$	Objective
○	Solution to an objective

Under-approximation



Sufficient condition:

- no cycle
- each objective has a solution



$\boxed{d_2}$

$d_0 \Rsh^* d_2 \rightarrow ○$

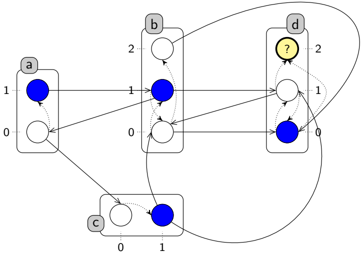
$\boxed{b_0}$ $\rightarrow$ $b_0 \Rsh^* b_0 \rightarrow ○$

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

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

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

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

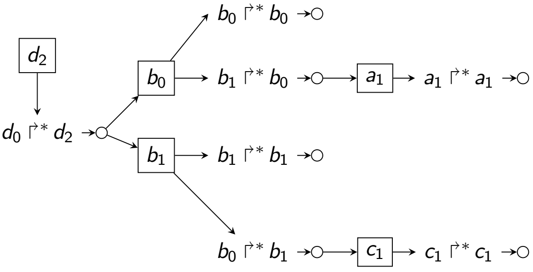


Under-approximation

Sufficient condition:

- no cycle
- each objective has a solution

P is **true** ⇒ R is **true**


$\boxed{d_2}$

$d_0 \Rsh^* d_2$ →○

$\boxed{b_0}$

$b_0 \Rsh^* b_0$ →○

$b_1 \Rsh^* b_0$ →○

$\boxed{a_1}$

$a_1 \Rsh^* a_1$ →○

$\boxed{b_1}$

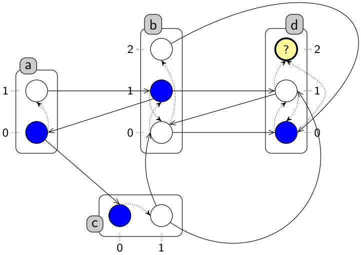
$b_1 \Rsh^* b_1$ →○

$b_0 \Rsh^* b_1$ →○

$\boxed{c_1}$

$c_1 \Rsh^* c_1$ →○

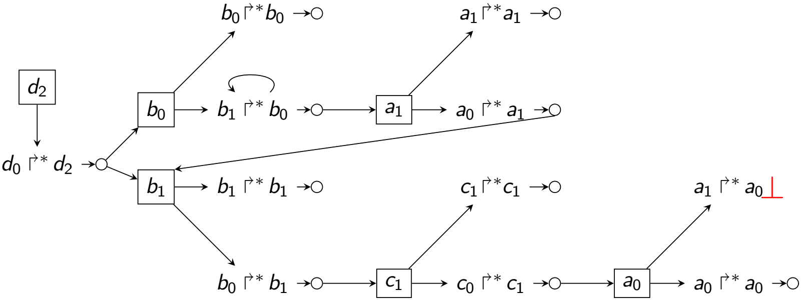
$\boxed{d_2}$	Required process
$d_0 \Rsh^* d_2$	Objective
○	Solution to an objective



Under-approximation

Sufficient condition:

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



d_2

$d_0 \Rsh^* d_2$

b_0

b_1

$b_0 \Rsh^* b_0$

$b_1 \Rsh^* b_0$

a_1

$a_1 \Rsh^* a_1$

$a_0 \Rsh^* a_1$

$b_1 \Rsh^* b_1$

$b_0 \Rsh^* b_1$

c_1

$c_1 \Rsh^* c_1$

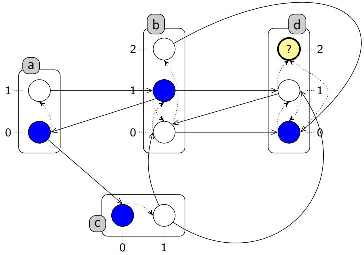
$c_0 \Rsh^* c_1$

a_0

$a_1 \Rsh^* a_0 \bot$

$a_0 \Rsh^* a_0$

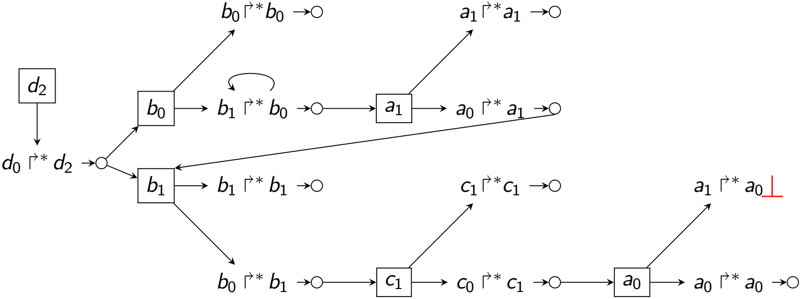
Under-approximation



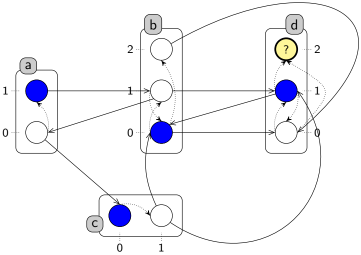
Sufficient condition:

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

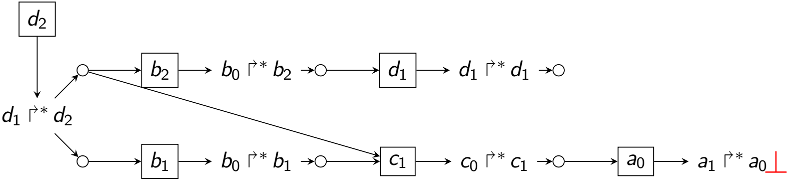
P is **false** ⇒ **Inconclusive**



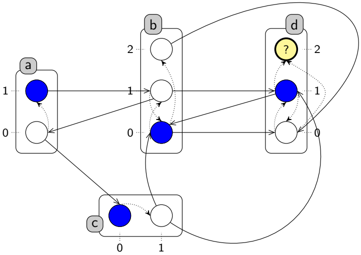
Over-approximation



Necessary condition:



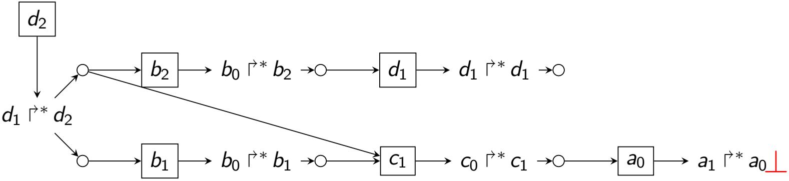
Over-approximation



Necessary condition:

There exists a traversal with no cycle

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



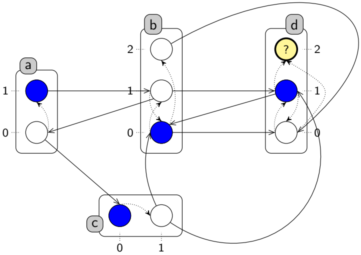
$\boxed{d_2}$

$d_1 \Rsh^* d_2$

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

$\boxed{b_1}$ $b_0 \Rsh^* b_1$ $\boxed{c_1}$ $c_0 \Rsh^* c_1$ $\boxed{a_0}$ $a_1 \Rsh^* a_0 \bot$

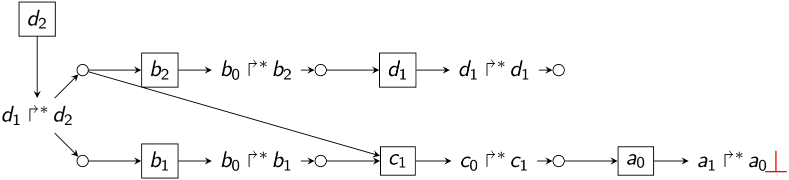
Over-approximation



Necessary condition:

~~There exists a traversal~~ with no cycle

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



d_2

$d_1 \Rsh^* d_2$

b_2

$b_0 \Rsh^* b_2$

d_1

$d_1 \Rsh^* d_1$

b_1

$b_0 \Rsh^* b_1$

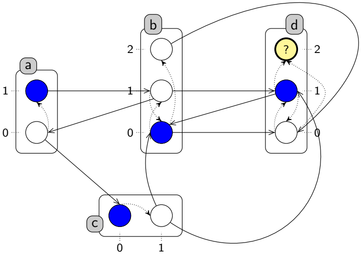
c_1

$c_0 \Rsh^* c_1$

a_0

$a_1 \Rsh^* a_0 \bot$

Over-approximation

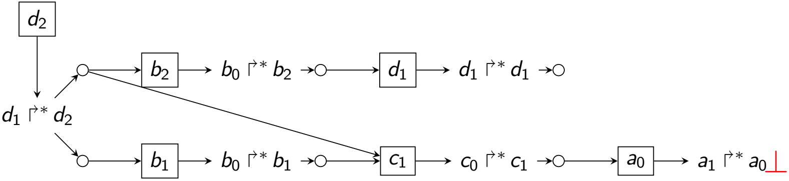


Necessary condition:

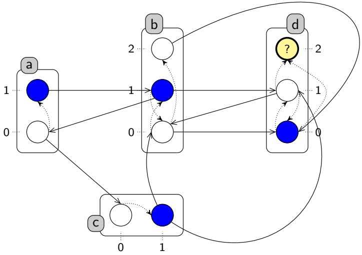
~~There exists a traversal~~ with no cycle

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

Q is **false** ⇒ R is **false**


d_2
$d_1 \upharpoonright^* d_2$
b_2
$b_0 \upharpoonright^* b_2$
d_1
$d_1 \upharpoonright^* d_1$
b_1
$b_0 \upharpoonright^* b_1$
c_1
$c_0 \upharpoonright^* c_1$
a_0
$a_1 \upharpoonright^* a_0 \perp$

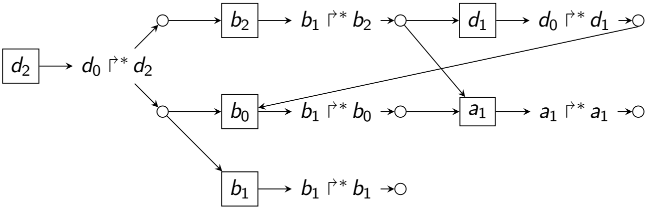
Over-approximation



Necessary condition:

There exists a traversal with no cycle

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



$\boxed{d_2}$ → $d_0 \Rsh^* d_2$

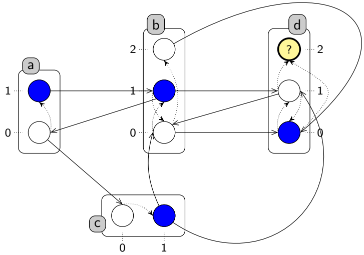
$\boxed{b_2}$ → $b_1 \Rsh^* b_2$

$\boxed{d_1}$ → $d_0 \Rsh^* d_1$

$\boxed{b_0}$ → $b_1 \Rsh^* b_0$

$\boxed{a_1}$ → $a_1 \Rsh^* a_1$

$\boxed{b_1}$ → $b_1 \Rsh^* b_1$



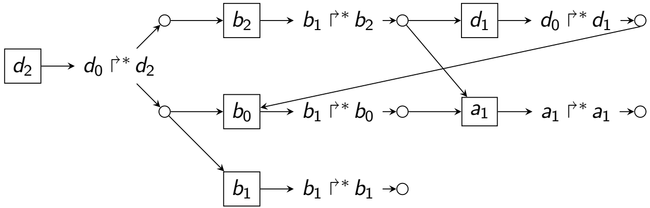
Over-approximation

Necessary condition:

There exists a traversal with no cycle

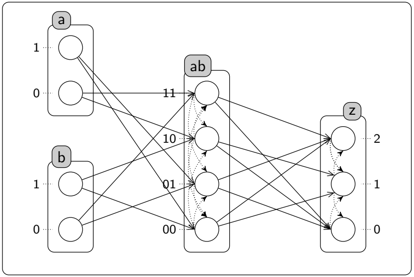
- objective → follow **one** solution
- solution → follow **all** processes
- process → follow **all** objectives

***R* is true ⇒ Inconclusive**


d_2
$d_0 \upharpoonright^* d_2$
b_2
$b_1 \upharpoonright^* b_2$
d_1
$d_0 \upharpoonright^* d_1$
b_0
$b_1 \upharpoonright^* b_0$
a_1
$a_1 \upharpoonright^* a_1$
b_1
$b_1 \upharpoonright^* b_1$

Translation of PH models

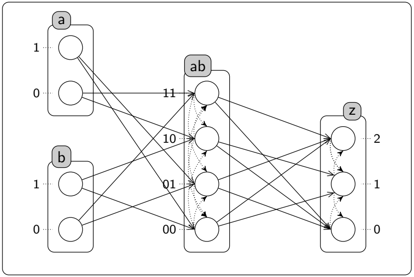
[Folschette *et al.*, *Computational Methods in Systems Biology*, 2012]



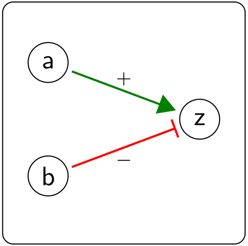
Process Hitting
Efficient but recent

Translation of PH models

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



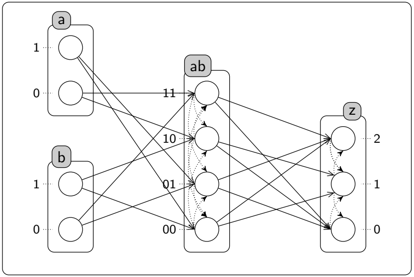
Process Hitting
Efficient but recent

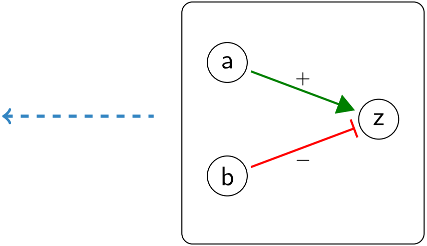


Boolean Networks
Widespread & readable

Translation of PH models

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



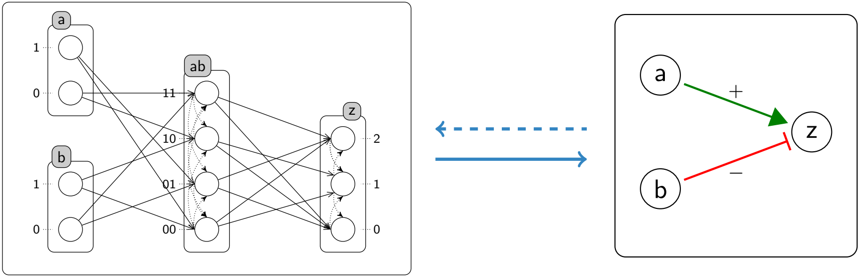


Process Hitting
Efficient but recent

Boolean Networks
Widespread & readable

Translation of PH models

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

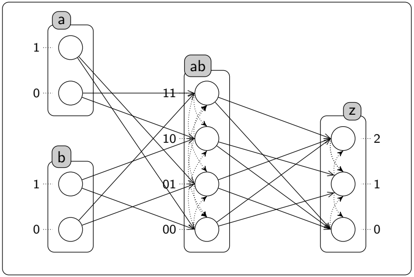


Process Hitting
Efficient but recent

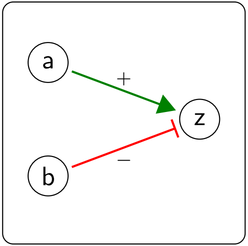
Boolean Networks
Widespread & readable

Enrichment of PH semantics

[Folschette *et al.*, *CS2Bio'13*, 2013]



≠

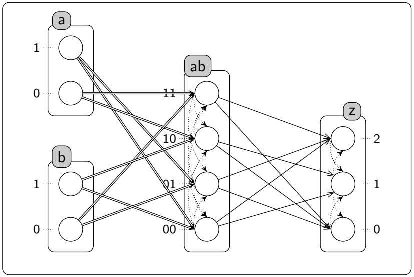


Process Hitting
Loose behaviour

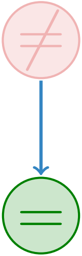
Boolean Networks
Accurate behaviour

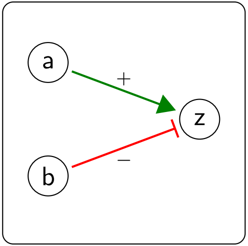
Enrichment of PH semantics

[Folschette *et al.*, *CS2Bio'13*, 2013]



Process Hitting
Accurate behaviour



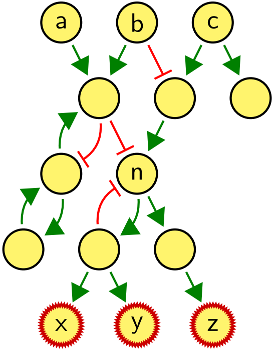


Boolean Networks
Accurate behaviour

Gene therapies

Modify DNA to cure a disease

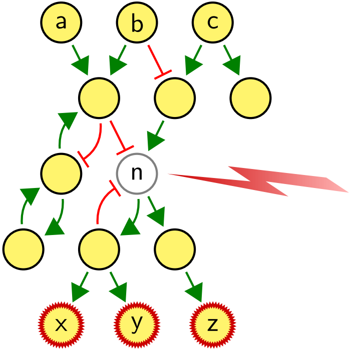
- Replace a mutated gene → remove a **harmful protein**
- Add a new gene → produce a **therapeutic protein**



Gene therapies

Modify DNA to cure a disease

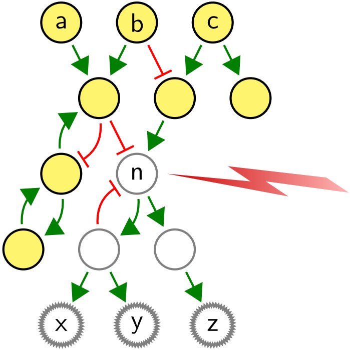
- Replace a mutated gene → remove a **harmful protein**
- Add a new gene → produce a **therapeutic protein**



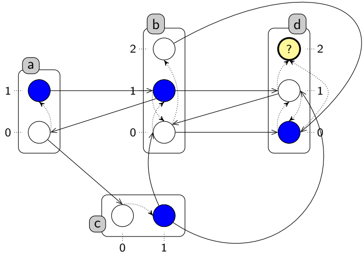
Gene therapies

Modify DNA to cure a disease

- Replace a mutated gene → remove a **harmful protein**
- Add a new gene → produce a **therapeutic protein**



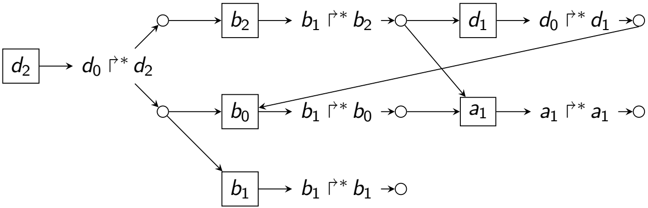
Back to the Over-approximation



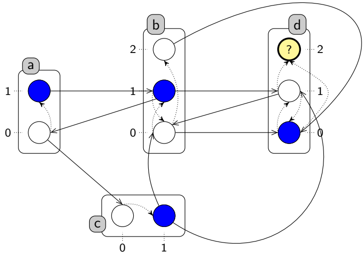
Necessary condition:

There exists a traversal with no cycle

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



Back to the Over-approximation

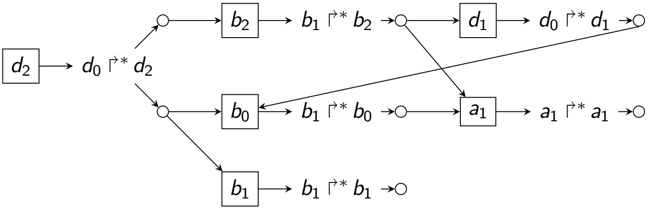


Necessary condition:

There exists a traversal with no cycle

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

***R* is true ⇒ Inconclusive**



$\boxed{d_2}$

$d_0 \Rsh^* d_2$

$\boxed{b_2}$

$b_1 \Rsh^* b_2$

$\boxed{d_1}$

$d_0 \Rsh^* d_1$

$\boxed{b_0}$

$b_1 \Rsh^* b_0$

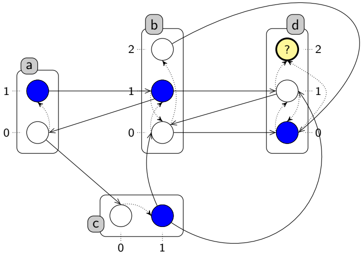
$\boxed{a_1}$

$a_1 \Rsh^* a_1$

$\boxed{b_1}$

$b_1 \Rsh^* b_1$

Back to the Over-approximation

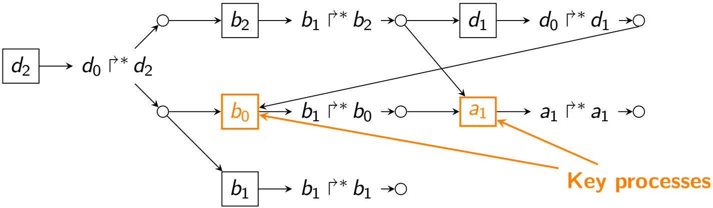


Necessary condition:

There exists a traversal with no cycle

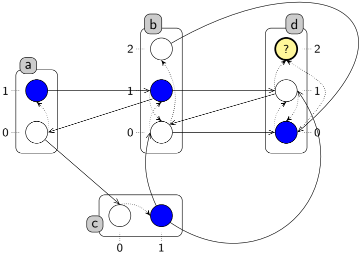
- objective → follow **one** solution
- solution → follow **all** processes
- process → follow **all** objectives

***R* is true ⇒ Inconclusive**



Key processes

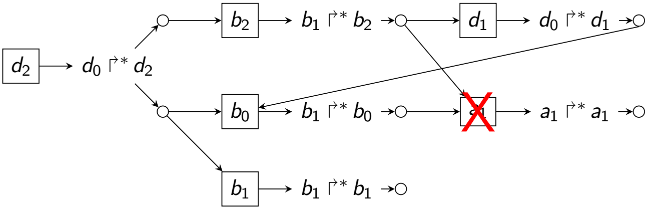
Back to the Over-approximation



Necessary condition:

There exists a traversal with no cycle

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



d_2 → $d_0 \Rsh^* d_2$

b_2 → $b_1 \Rsh^* b_2$

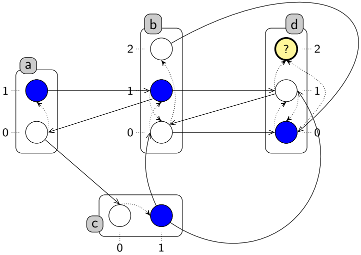
d_1 → $d_0 \Rsh^* d_1$

b_0 → $b_1 \Rsh^* b_0$

a_1 → $a_1 \Rsh^* a_1$

b_1 → $b_1 \Rsh^* b_1$

Back to the Over-approximation

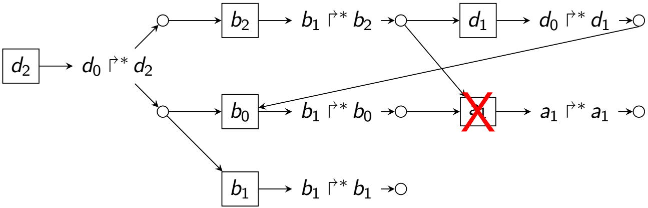


Necessary condition:

There exists a traversal with no cycle

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

Q is **false** ⇒ R is **false**



d_2 → $d_0 \Rsh^* d_2$

b_2 → $b_1 \Rsh^* b_2$ → d_1 → $d_0 \Rsh^* d_1$

b_0 → $b_1 \Rsh^* b_0$ → a_1 → $a_1 \Rsh^* a_1$

b_1 → $b_1 \Rsh^* b_1$

Summary & Conclusion

- What is Bio-informatics?
  → Qualitative modelling of **gene regulations**
  → Large models are hard to study (**exponential**)
- What do I do?
  → The **Process Hitting** modelling
  → Very efficient on large-scale models (**polynomial**)
  → My contribution: reach the expressivity of **boolean networks**
- What for?
  → **Validating** & **utilizing** biological models
  → Gene therapies

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, 171-191. Springer Berlin Heidelberg, 2011.
- Loïc Paulevé, Morgan Magnin, Olivier Roux. Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*. 2012.
- Hidde de Jong. Modeling and simulation of genetic regulatory systems: a literature review, *Journal of Computational biology* 9(1), 67–103. 2002.
- Maxime Folschette, Loïc Paulevé, Katsumi Inoue, Morgan Magnin, Olivier Roux. Concretizing the Process Hitting into Biological Regulatory Networks. In David Gilbert and Monika Heiner, editors, *Computational Methods in Systems Biology X*, Lecture Notes in Computer Science, 166–186. Springer Berlin Heidelberg, 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.

Thank you