— MOVEP'2012 —

10th School for young researchers about

Modelling and Verifying Parallel processes

Inferring Biological Regulatory Networks from Process Hitting models

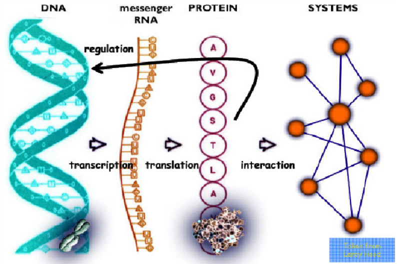
Maxime FOLSCHETTE[1,2]

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, Katsumi INOUE, Morgan MAGNIN, Olivier ROUX

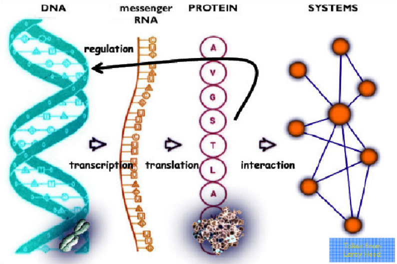
Context and Aims

MeForBio team: Algebraic modeling to study complex dynamical biological systems



Context and Aims

MeForBio team: Algebraic modeling to study complex dynamical biological systems

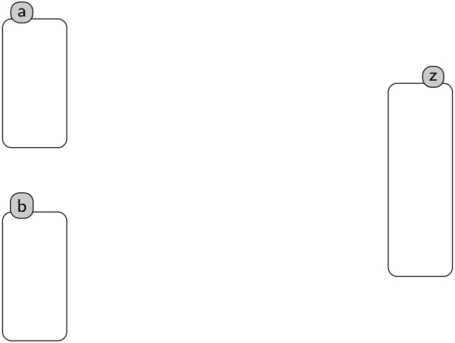


1) Two main models
    - Historical model: **Biological Regulatory Network (René Thomas)**
    - New developed model: **Process Hitting**

2) Allow efficient translation from Process Hitting to BRN

The Process Hitting modeling

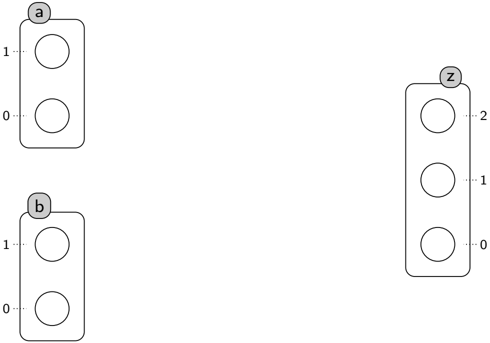
[PMR12-MSCS]



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

The Process Hitting modeling

[PMR12-MSCS]

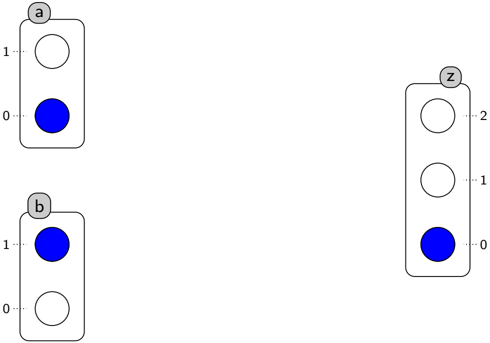


Sorts: components a, b, z

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

The Process Hitting modeling

[PMR12-MSCS]



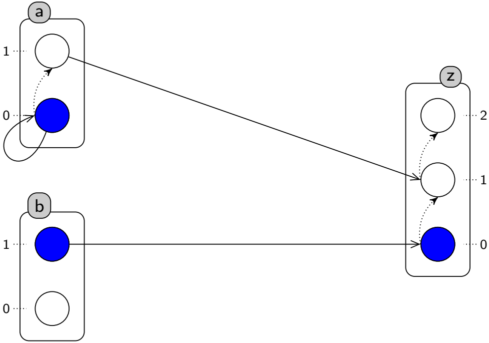
Sorts: components $a,\ b,\ z$

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

[PMR12-MSCS]



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

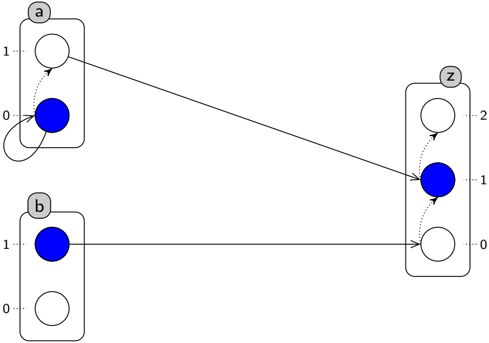
Processes: local states / levels of expression $z_0\,,\ z_1\,,\ z_2$

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

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

The Process Hitting modeling

[PMR12-MSCS]



Sorts: components a, b, z

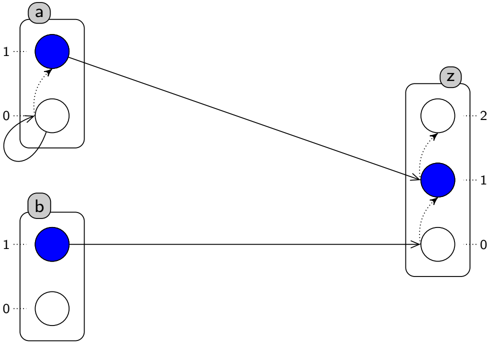
Processes: local states / levels of expression z_0, z_1, z_2

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 modeling

[PMR12-MSCS]



Sorts: components a, b, z

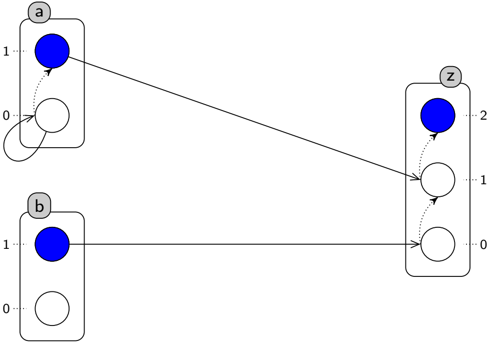
Processes: local states / levels of expression z_0, z_1, z_2

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 modeling

[PMR12-MSCS]



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

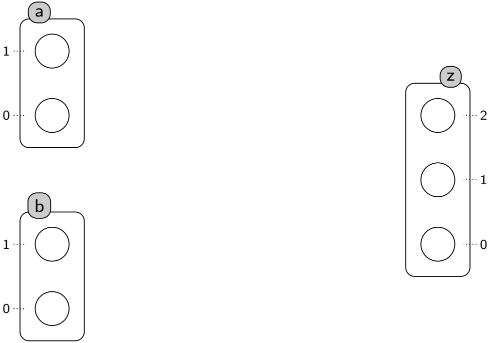
Processes: local states / levels of expression $z_0,\ z_1,\ z_2$

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

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

Adding cooperations

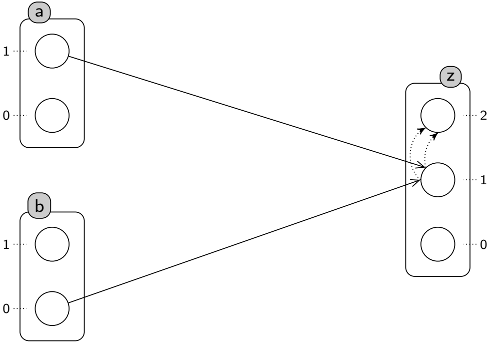
[PMR12-MSCS]



How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Adding cooperations

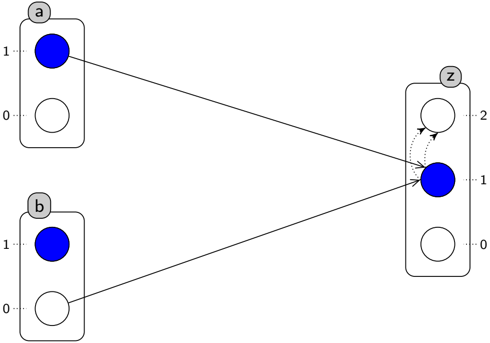
[PMR12-MSCS]



How to introduce some **cooperation** between sorts? $a_1 \land b_0 \rightarrow z_1 \upharpoonright z_2$

Adding cooperations

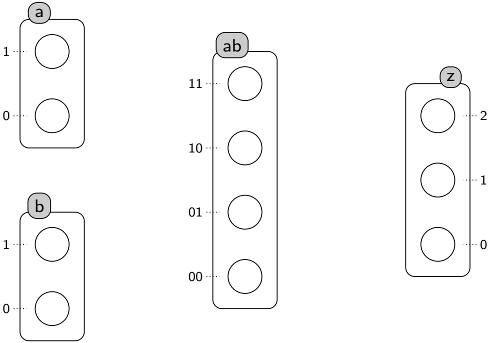
[PMR12-MSCS]



How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Adding cooperations

[PMR12-MSCS]

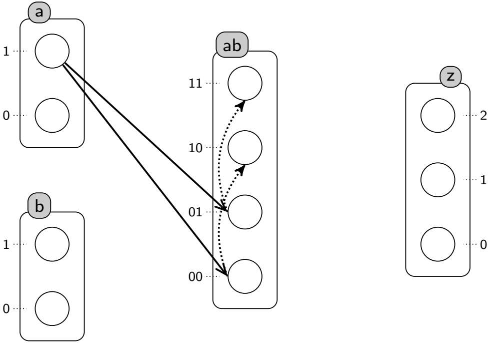


How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Adding cooperations

[PMR12-MSCS]

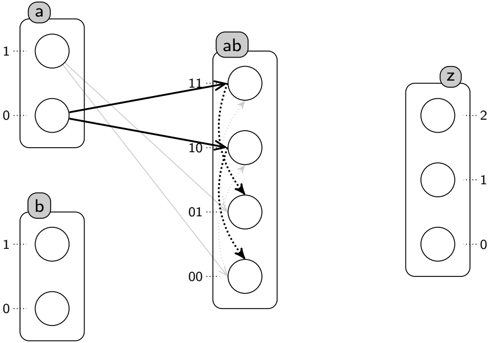


How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Adding cooperations

[PMR12-MSCS]

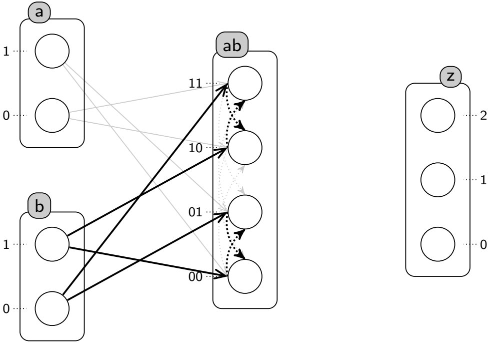


How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Adding cooperations

[PMR12-MSCS]

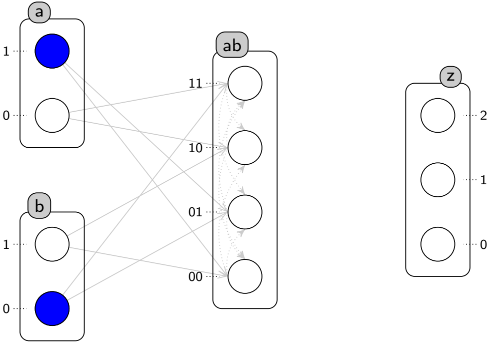


How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Adding cooperations

[PMR12-MSCS]



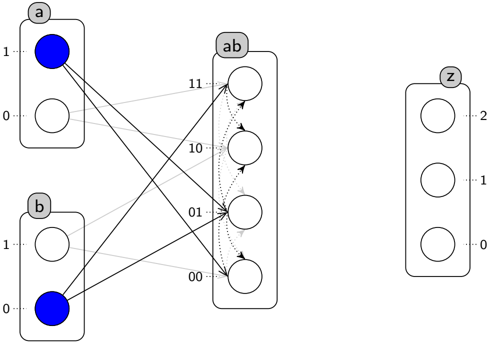
How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \Rsh z_2$

Solution: a **cooperative sort** ab

Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle$

Adding cooperations

[PMR12-MSCS]



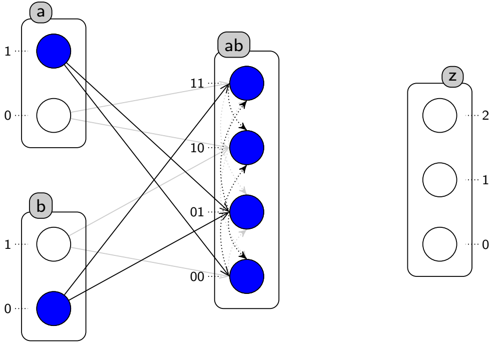
How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle$

Adding cooperations

[PMR12-MSCS]



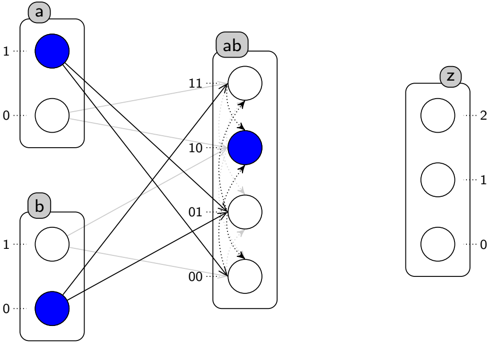
How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle$

Adding cooperations

[PMR12-MSCS]



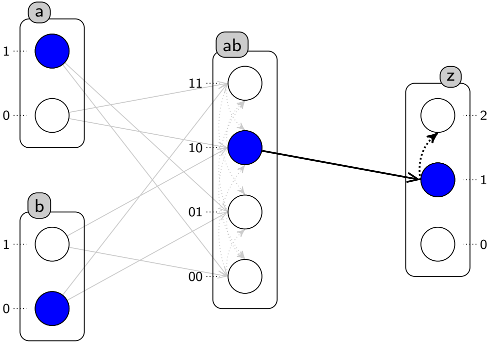
How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

Solution: a **cooperative sort** ab

Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle \Rightarrow ab_{10}$

Adding cooperations

[PMR12-MSCS]



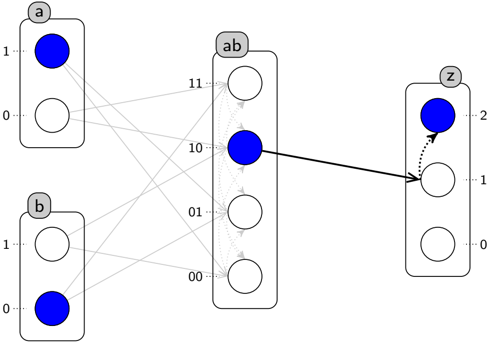
How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

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

Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle \Rightarrow ab_{10}$

Adding cooperations

[PMR12-MSCS]



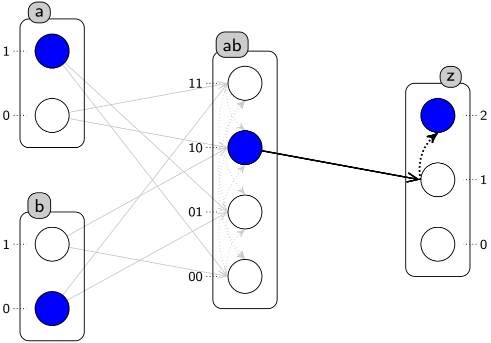
How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

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

Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle \Rightarrow ab_{10}$

Adding cooperations

[PMR12-MSCS]



How to introduce some **cooperation** between sorts? $a_1 \wedge b_0 \rightarrow z_1 \upharpoonright z_2$

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

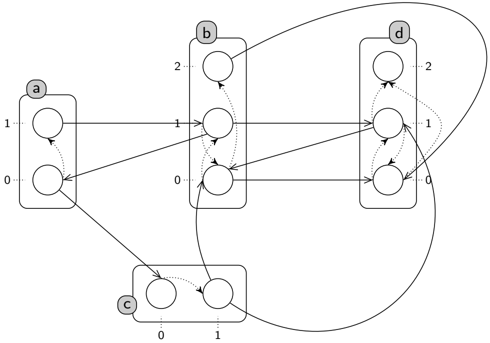
Constraint: each configuration is represented by one process $\langle a_1, b_0 \rangle \Rightarrow ab_{10}$

Advantage: regular sort; drawbacks: complexity, temporal shift

Static analysis: successive reachability

[PMR12-MSCS]

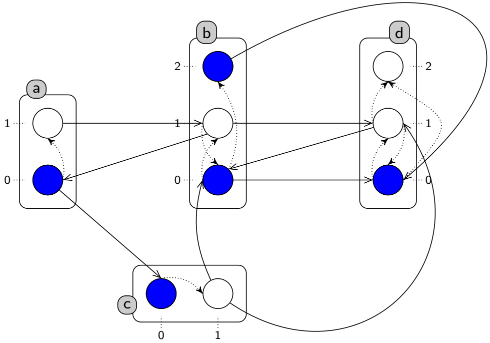
Successive reachability of processes:



Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



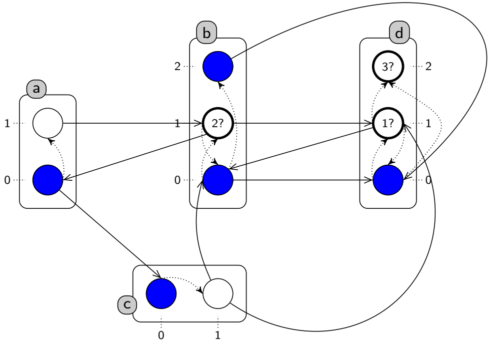
- Initial context

$\langle a_1, \{b_0, b_1\}, c_0, z_0\rangle$

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:

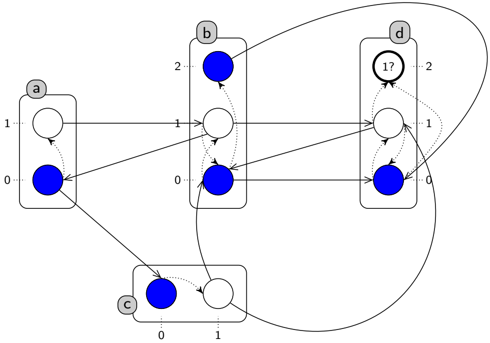


- Initial context
  $\langle a_1, \{b_0, b_1\}, c_0, z_0\rangle$
- Objectives
  $[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



- Initial context

$\langle a_1, \{b_0, b_1\}, c_0, z_0\rangle$

- Objectives

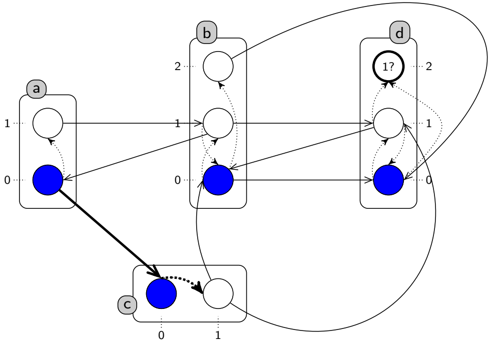
$[\ \Rsh d_1 :: \Rsh b_1 :: \Rsh d_2\ ]$

$[\ \Rsh d_2\ ]$

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



- Initial context

  $\langle a_1, \{b_0, b_1\}, c_0, z_0\rangle$

- Objectives

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

  $[\ \Rsh d_2\ ]$

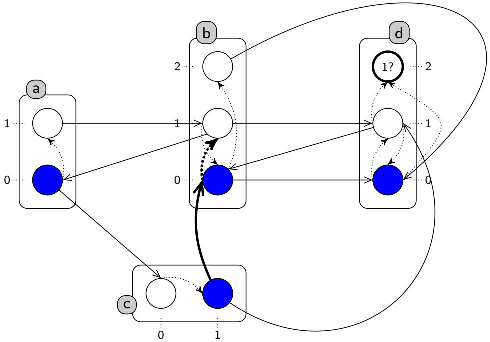
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



- Initial context

$\langle a_1, \{b_0, b_1\}, c_0, z_0 \rangle$

- Objectives

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

$[\ \Rsh d_2\ ]$

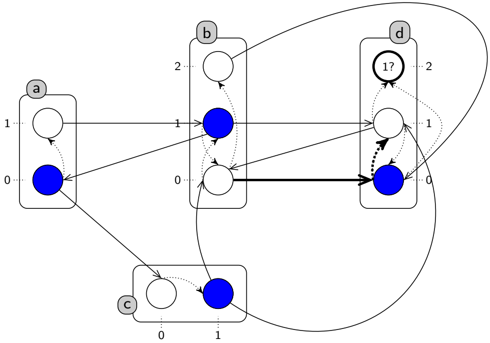
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



- Initial context

$\langle a_1, \{b_0, b_1\}, c_0, z_0\rangle$

- Objectives

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

$[\; \Rsh d_2 \;]$

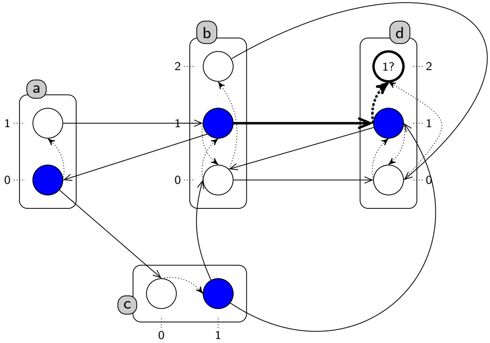
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



- Initial context

$\langle a_1, \{b_0, b_1\}, c_0, z_0 \rangle$

- Objectives

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

$[\ \Rsh d_2\ ]$

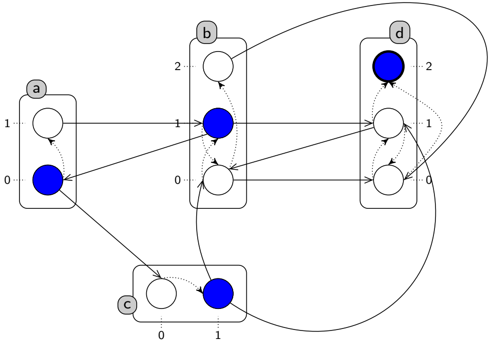
$\rightarrow$ Concretization of the objective = scenario

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

Static analysis: successive reachability

[PMR12-MSCS]

Successive reachability of processes:



- Initial context

  $\langle a_1, \{b_0, b_1\}, c_0, z_0 \rangle$

- Objectives

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

  $[\ \Rsh d_2\ ]$

$\rightarrow$ Concretization of the objective $=$ scenario

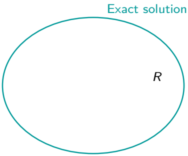
$a_0 \rightarrow c_0 \Rsh c_1 \ :: \ \ b_0 \rightarrow d_0 \Rsh d_1 \ :: \ \ c_1 \rightarrow b_0 \Rsh b_1 \ :: \ \ b_1 \rightarrow d_1 \Rsh d_2$

Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

- → Directly checking an objective sequence R is hard
- → Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



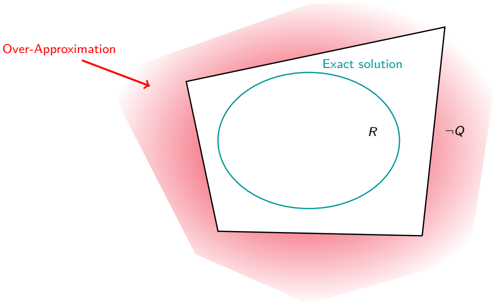
Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

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



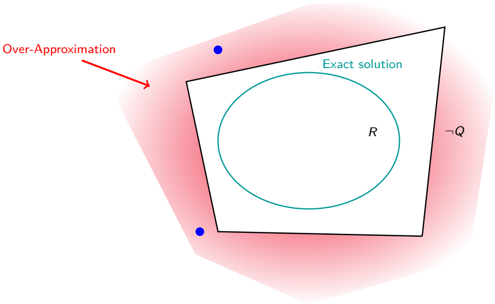
Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

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

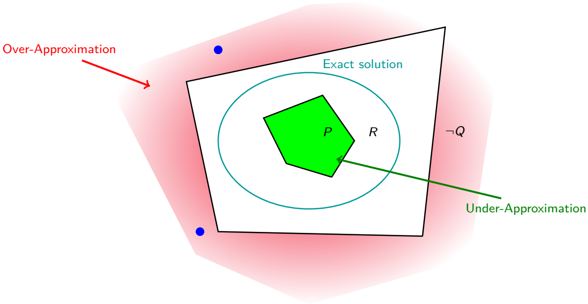


Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

- → Directly checking an objective sequence R is hard
- → Rather check the approximations P and Q, where $P \Rightarrow R \Rightarrow Q$:



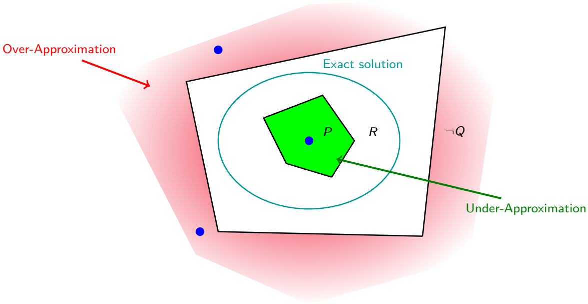
Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

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



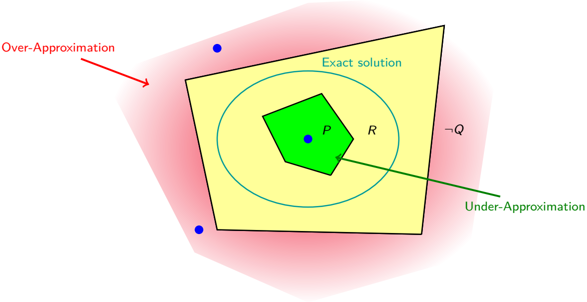
Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

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



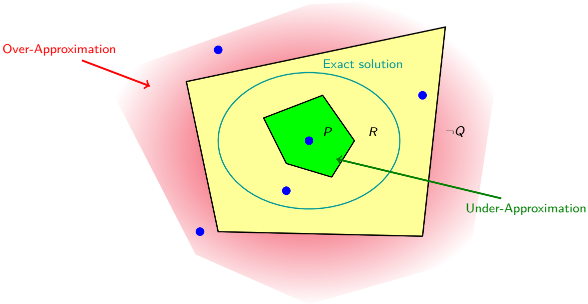
Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

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



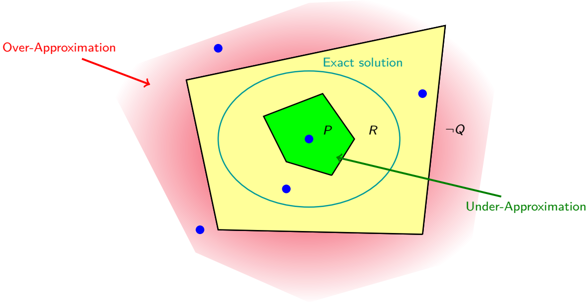
Over- and Under-approximations

[PMR12-MSCS]

Static analysis by abstractions:

→ Directly checking an objective sequence R is hard

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



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

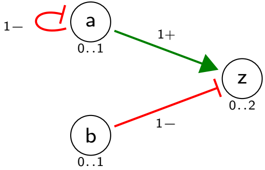
→ Efficient for big models with few levels of expression

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 **linear time** in the number of sorts
- Useful for the study of **large biological models**
  - → Up to hundreds of sorts
- (Future) extensions
  - → Actions with stochasticity
  - → Actions with priorities
  - → Continuous time with clocks?

Biological Regulatory Network (Thomas' modeling)

[RCB08]



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

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

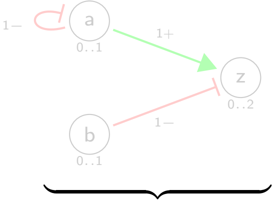
Proposed by René Thomas in 1973, several extensions since then

Historical bio-informatics model for studying genes interactions

Widely used and well-adapted to represent dynamic gene systems

Biological Regulatory Network (Thomas' modeling)

[RCB08]



Interaction Graph

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

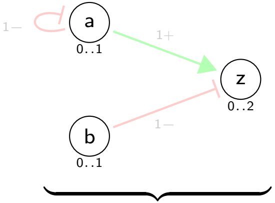
ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

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

Biological Regulatory Network (Thomas' modeling)

[RCB08]



Interaction Graph

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

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

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

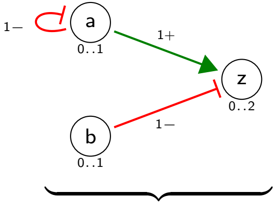
Nodes: genes

→ Name *a*, *b*, *z*

→ Possible values (levels of expression) 0..1, 0..2

Biological Regulatory Network (Thomas' modeling)

[RCB08]



Interaction Graph

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

ω	
a	$k_{a,\omega}$
$+$	1
$-$	0

$k_{b,\omega}$
1

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

Nodes: genes

→ Name a, b, z

→ Possible values (levels of expression) 0..1, 0..2

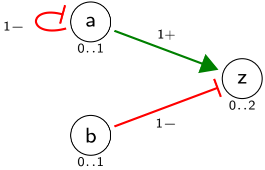
Edges: interactions

→ Threshold 1

→ Type (activation or inhibition) $+$ / $-$

Biological Regulatory Network (Thomas' modeling)

[RCB08]



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

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Parametrization

Parametrization: strength of the influences (cooperations)

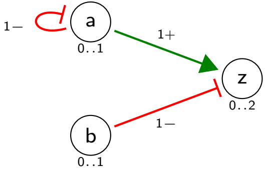
Maps of tendencies for each gene

→ To any **influences of predecessors** ω

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

Biological Regulatory Network (Thomas' modeling)

[RCB08]



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

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Parametrization

Parametrization: strength of the influences (cooperations)

Maps of tendencies for each gene

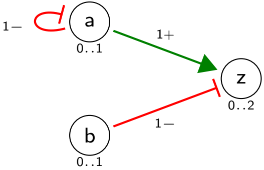
→ To any **influences of predecessors** ω

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

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

Biological Regulatory Network (Thomas' modeling)

[RCB08]



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

ω	$k_{a,\omega}$
a	
$+$	1
$-$	0

$k_{b,\omega}$
1

Biological Regulatory Network

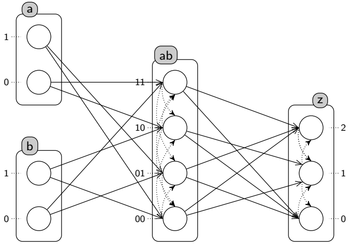
→ All needed information to run the model or study its dynamics:

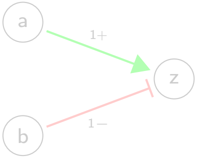
- Build the State Graph
- Find reachability properties, fixed points, attractors
- Other properties...

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

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

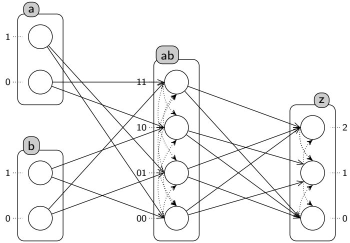
Inferring a BRN with Thomas' parameters

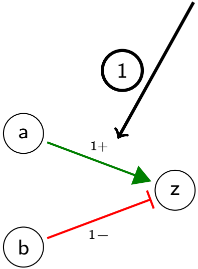




ω	$k_{z,\omega}$
$\varnothing$	1
$\{b\}$	0
$\{a\}$	2
$\{a; b\}$	1

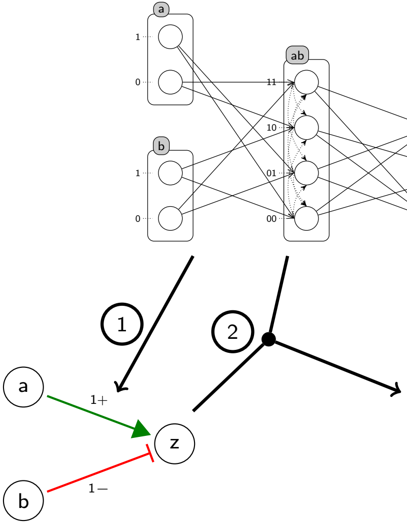
Inferring a BRN with Thomas' parameters





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

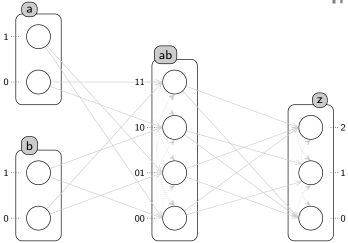
Inferring a BRN with Thomas' parameters

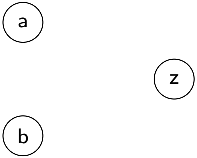


ω	$k_{z,\omega}$
$\varnothing$	1
$\{b\}$	0
$\{a\}$	2
$\{a; b\}$	1

Inferring the Interaction Graph

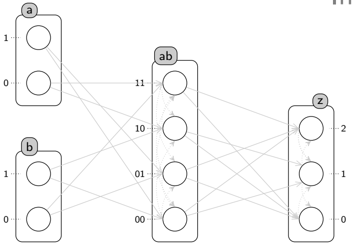
[CMSB12]

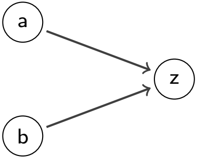




Inferring the Interaction Graph

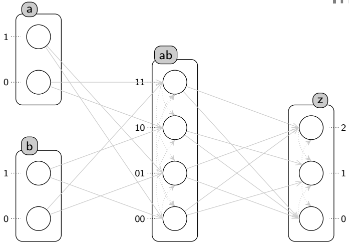
[CMSB12]

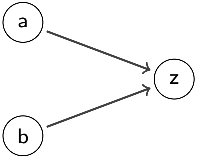




Inferring the Interaction Graph

[CMSB12]

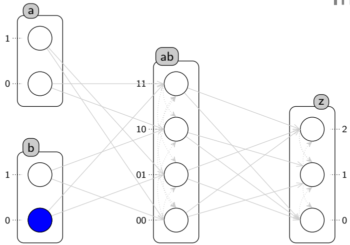


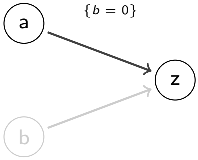


→ **Exhaustive search in all possible configurations**

Inferring the Interaction Graph

[CMSB12]



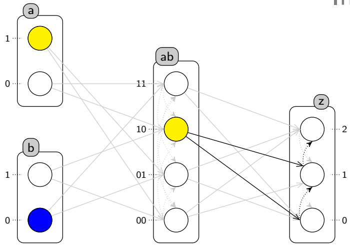


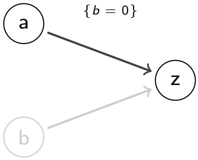
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



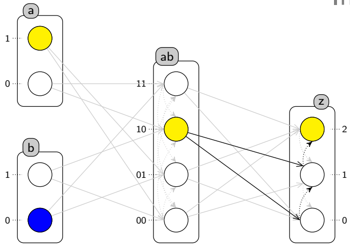


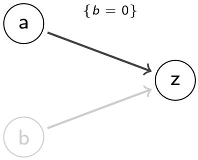
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



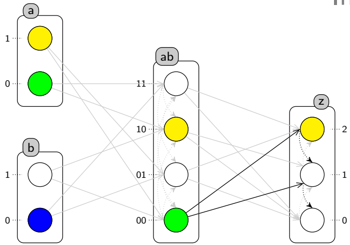


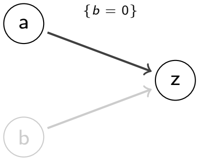
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



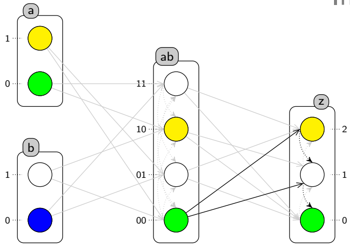


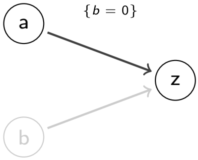
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



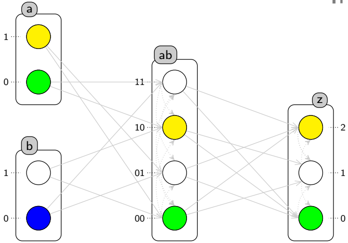


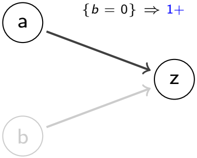
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



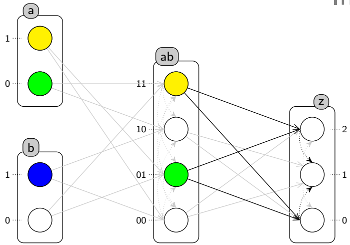


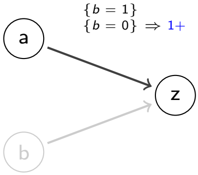
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



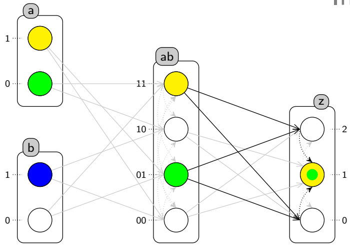


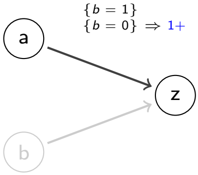
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



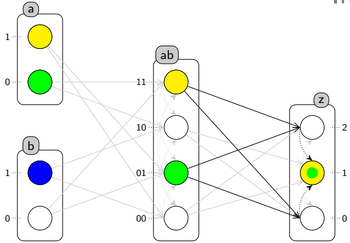


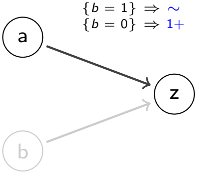
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



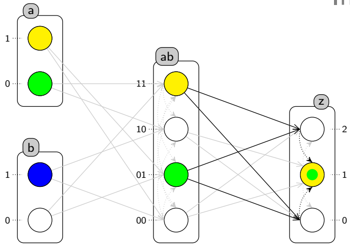


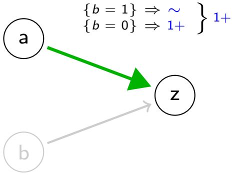
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]



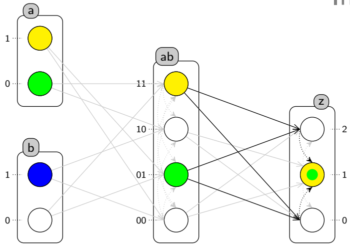


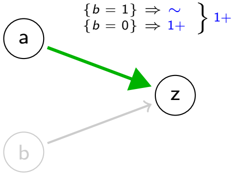
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[CMSB12]





$\{b = 1\} \Rightarrow \sim$
$\{b = 0\} \Rightarrow 1+$ $\Big\}$ $1+$

→ **Exhaustive search in all possible configurations**

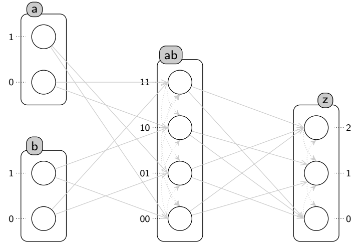
1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of this regulator [a_0, a_1] and watch the **focal processes**.
3. Conclude locally: $(a_0 \Rsh a_1 \Rightarrow z_0 \Rsh z_2) \Rightarrow$ activation $(+)$ & threshold $= 1$.
4. Iterate and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

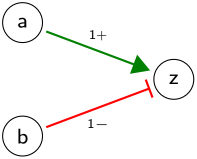
Problematic cases:

$$\left.\begin{array}{l}\rightarrow \text{No focal processes (cycle)}\\ \rightarrow \text{Opposite influences } (+\ \&\ -)\end{array}\right\} \Rightarrow \text{ Unsigned edge}$$

Inferring Parameters

[PMR10-TCSB]

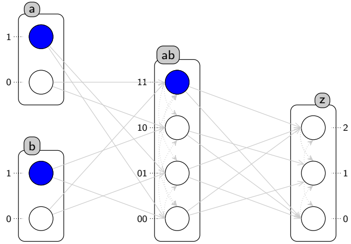


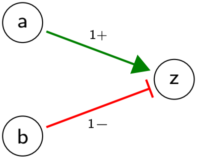


ω		$k_{z,\omega}$
a	b	
$-$	$+$	
$-$	$-$	
$+$	$+$	
$+$	$-$	

Inferring Parameters

[PMR10-TCSB]



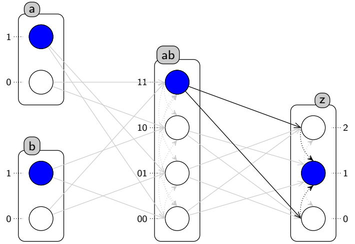


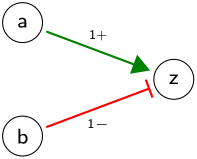
ω		$k_{z,\omega}$
a	b	
$-$	$+$	
$-$	$-$	
$+$	$+$	
$+$	$-$	

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

Inferring Parameters

[PMR10-TCSB]



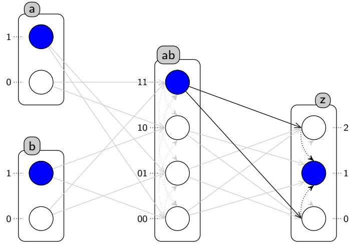


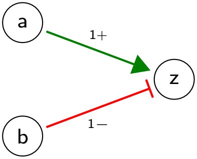
ω		$k_{z,\omega}$
a	b	
$-$	$+$	
$-$	$-$	
$+$	$+$	
$+$	$-$	

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

Inferring Parameters

[PMR10-TCSB]



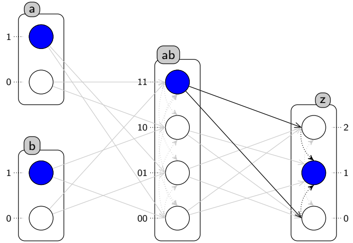


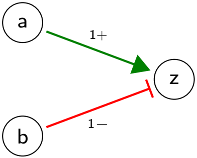
ω		$k_{z,\omega}$
a	b	
$-$	$+$	
$-$	$-$	
$+$	$+$	
$+$	$-$	1

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

Inferring Parameters

[PMR10-TCSB]





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

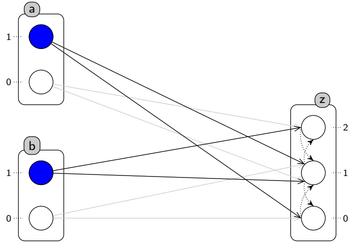
1. For each configuration of resources [$\omega = \{a^+, b^-\}$]
find the **focal processes**. If possible, conclude. [$k_{z,\{a^+,b^-\}} = 1$]

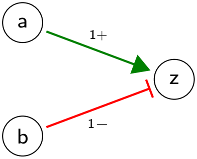
   Inconclusive cases:

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

Inferring Parameters

[PMR10-TCSB]





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

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

   Inconclusive cases:

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

2. If some parameters could not be inferred, enumerate all admissible parametrizations, regarding:

   - Biological constraints
   - The dynamics of the Process Hitting

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

Implementation

Workflow:

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

Implementation

Workflow:

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

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

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

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

Summary

1. Inference of the **complete Interaction Graph**
2. Inference of the **possibly partial Parametrization**
3. Enumerate all full & **admissible Parametrizations**

→ Exhaustive approaches

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

Summary

1. Inference of the **complete Interaction Graph**
2. Inference of the **possibly partial Parametrization**
3. Enumerate all full & **admissible Parametrizations**

→ Exhaustive approaches

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

Conclusion

Existing translation: René Thomas $\rightsquigarrow$ Process Hitting

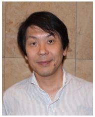
New translation: Process Hitting $\rightsquigarrow$ René Thomas

→ New **formal link** between the two models

→ More **visibility** to the Process Hitting

A multi-team topic

Inoue Laboratory (NII, Sokendai): Constraint Programming, Systems Biology
MeForBio (IRCCyN, ÉCN): Formal Methods for Bioinformatics
AMIB (LIX, Polytechnique): Algorithms and Models for Integrative Biology



Katsumi INOUE
Professor & team leader

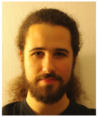
Inoue Laboratory

Loïc PAULEVÉ
Post-doc

AMIB

Olivier ROUX
Professor & team leader

Morgan MAGNIN
Associate professor



Maxime FOLSCHETTE
2nd year PhD student

MeForBio

Bibliography

[PMR10-TCSB] 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, 171-191. Springer Berlin/Heidelberg, 2011.

[PMR12-MSCS] Loïc Paulevé, Morgan Magnin, Olivier Roux. Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*, 2012.

[RCB08] Adrien Richard, Jean-Paul Comet, Gilles Bernot. R. Thomas' logical method, 2008. Invited at *Tutorials on modelling methods and tools: Modelling a genetic switch and Metabolic Networks*, Spring School on Modelling Complex Biological Systems in the Context of Genomics.

[CMSB12] 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, pages 166–186. Springer Berlin Heidelberg, 2012.

Thank you