MDSC team seminar

Qualitative modeling and dynamical analysis of Biological Regulatory Networks using Asynchronous Automata Networks

Maxime FOLSCHETTE

MDSC team / Bioinfo project / I3S laboratory / University of Nice–Sophia Antipolis
maxime.folschette@i3s.unice.fr
http://maxime.folschette.name/

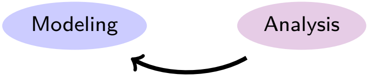
2015/09/28

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension

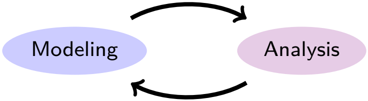


The required analysis has an impact on modeling

- The modeling tools must be adapted to the observed properties

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension



The required analysis has an impact on modeling

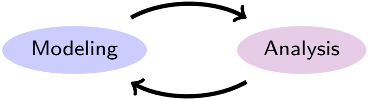
- The modeling tools must be adapted to the observed properties

Modeling choices have an impact on the results of the analysis

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

The Modeling/Analysis duality

Modeling a system is the first step towards its comprehension



The required analysis has an impact on modeling

- The modeling tools must be adapted to the observed properties

Modeling choices have an impact on the results of the analysis

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

The modeling and analysis steps of a system are strongly linked

Overview of This Presentation

Abstracting biological models

- Abstraction of biological components
- Discrete, asynchronous and unitary representations

Overview of This Presentation

Abstracting biological models

- Abstraction of biological components
- Discrete, asynchronous and unitary representations

Examples of **discrete models**

- Discrete Networks (Thomas modeling)
- Asynchronous Automata Networks
- Other extensions of the Process Hitting formalism

Overview of This Presentation

Abstracting biological models

- Abstraction of biological components
- Discrete, asynchronous and unitary representations

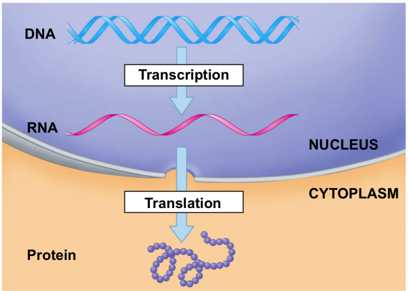
Examples of **discrete models**

- Discrete Networks (Thomas modeling)
- Asynchronous Automata Networks
- Other extensions of the Process Hitting formalism

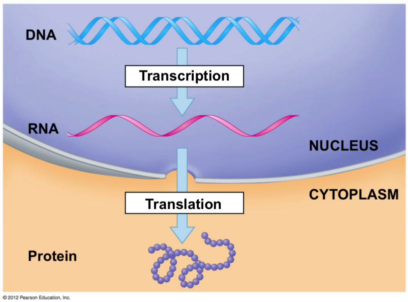
Analysis of the dynamics of discrete models

- Static analysis on the structure
- Abstract interpretation
- A μ-calculus approach

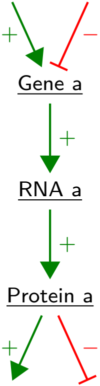
Abstractions of the Representation



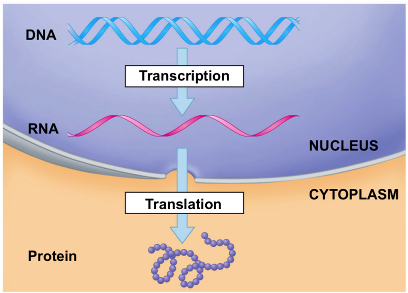
Abstractions of the Representation

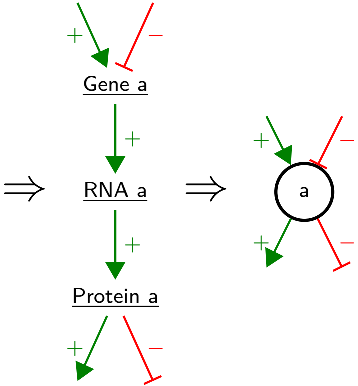


⇒



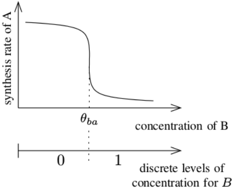
Abstractions of the Representation





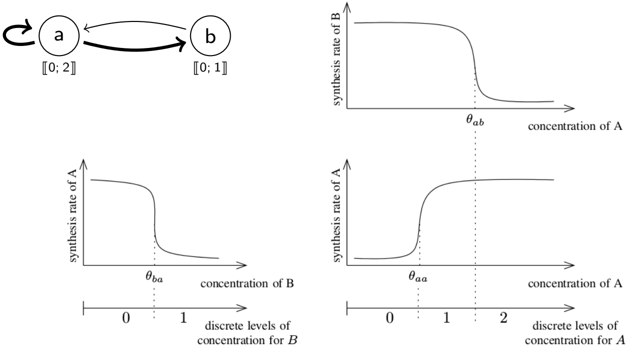
Discretization and Asynchronism

[Richard, *Advances in Applied Mathematics*, 2010]



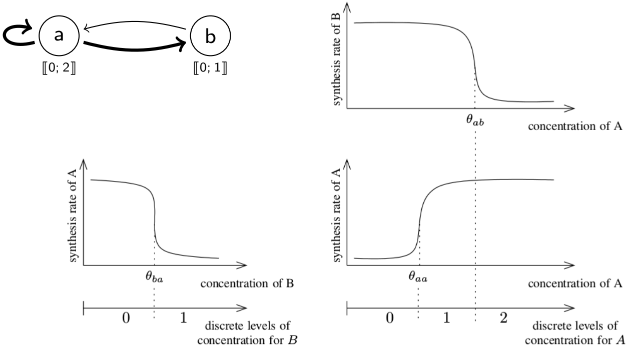
Discretization and Asynchronism

[Richard, *Advances in Applied Mathematics*, 2010]



Discretization and Asynchronism

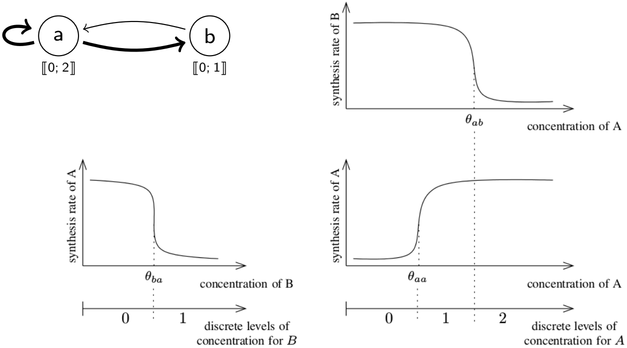
[Richard, *Advances in Applied Mathematics*, 2010]



- Unknown real values of concentrations or continuous activity levels
  $\longrightarrow$ Abstracted as thresholds or **discrete levels**

Discretization and Asynchronism

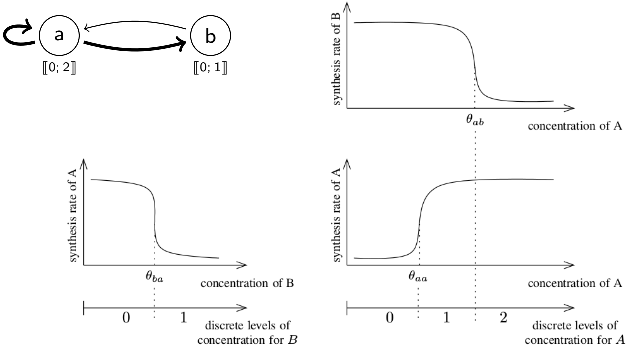
[Richard, *Advances in Applied Mathematics*, 2010]



- Unknown real values of concentrations or continuous activity levels
  $\longrightarrow$ Abstracted as thresholds or **discrete levels**
- Continuous variations of the real values
  $\longrightarrow$ **Unitary** dynamics

Discretization and Asynchronism

[Richard, *Advances in Applied Mathematics*, 2010]

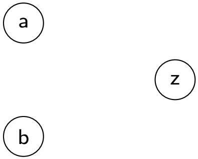


- Unknown real values of concentrations or continuous activity levels
  $\longrightarrow$ Abstracted as thresholds or **discrete levels**
- Continuous variations of the real values
  $\longrightarrow$ **Unitary** dynamics
- Simultaneous crossings of two thresholds never occurs
  $\longrightarrow$ **Asynchronous**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

Discrete Networks / Thomas Modeling

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

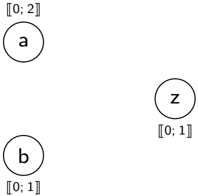
- A set of components $N = \{a, b, z\}$



Discrete Networks / Thomas Modeling

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

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $a \in \mathbb{F}^a = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$



Discrete Networks / Thomas Modeling

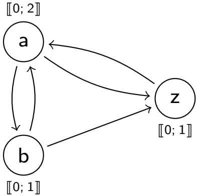
[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $a \in \mathbb{F}^a = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$
- An evolution function for each component $f^z : \mathbb{F} \to \mathbb{F}^z$

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

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

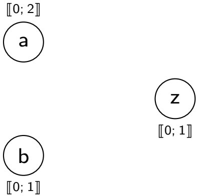
a	b	$f^z(a, b)$
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**0**
2	0	**0**
2	1	**1**



Discrete Networks / Thomas Modeling

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

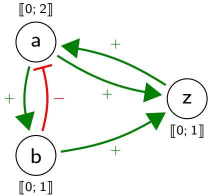
- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $a \in \mathbb{F}^a = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$



Discrete Networks / Thomas Modeling

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

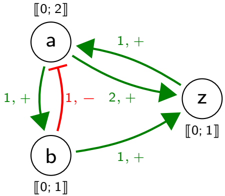
- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $a \in \mathbb{F}^a = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$
- Signs on the edges $a \xrightarrow{+} z$



Discrete Networks / Thomas Modeling

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

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $a \in \mathbb{F}^a = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$
- Signs on the edges $a \xrightarrow{+} z$ or signs + thresholds $a \xrightarrow{2,+} z$



Discrete Networks / Thomas Modeling

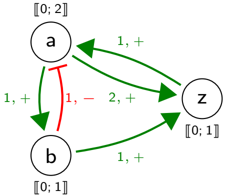
[Kauffman in *Journal of Theoretical Biology*, 1969]
[Thomas in *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A set of discrete expression levels for each component $a \in \mathbb{F}^a = [\![0; 2]\!]$
- The set of global states $\mathbb{F} = \mathbb{F}^a \times \mathbb{F}^b \times \mathbb{F}^z$
- Signs on the edges $a \xrightarrow{+} z$ or signs + thresholds $a \xrightarrow{2,+} z$
- Discrete parameters / evolution functions $f^a : \mathbb{F} \to \mathbb{F}^a$

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

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

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



State-graph of a Discrete Network

Several semantics exist regarding the updates:

- Synchronous (deterministic)
- **Asynchronous** (non-deterministic)
- Generalized (even more non-deterministic)

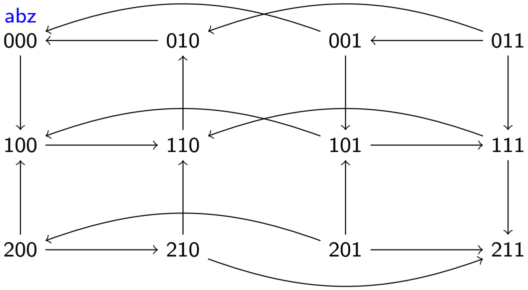
In every case, exponential size in the number of components

State-graph of a Discrete Network

Several semantics exist regarding the updates:

- Synchronous (deterministic)
- **Asynchronous** (non-deterministic)
- Generalized (even more non-deterministic)

In every case, exponential size in the number of components

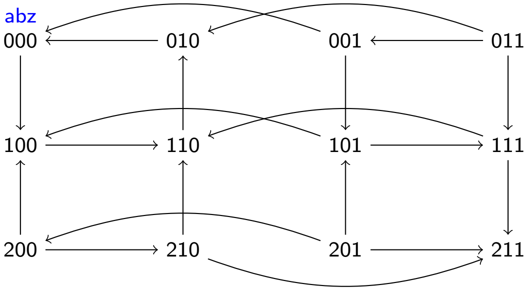


State-graph of a Discrete Network

Several semantics exist regarding the updates:

- Synchronous (deterministic)
- **Asynchronous** (non-deterministic)
- Generalized (even more non-deterministic)

In every case, exponential size in the number of components



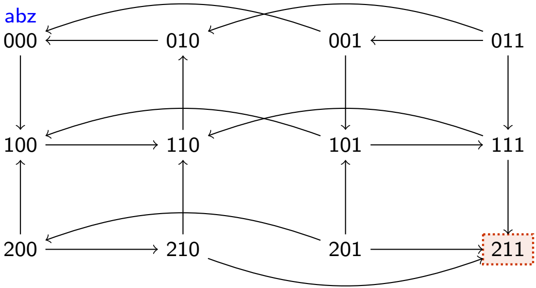
Attractor = minimal set of states from which the dynamics cannot escape
= terminal strongly connected component

State-graph of a Discrete Network

Several semantics exist regarding the updates:

- Synchronous (deterministic)
- **Asynchronous** (non-deterministic)
- Generalized (even more non-deterministic)

In every case, exponential size in the number of components



Attractor = minimal set of states from which the dynamics cannot escape
= terminal strongly connected component

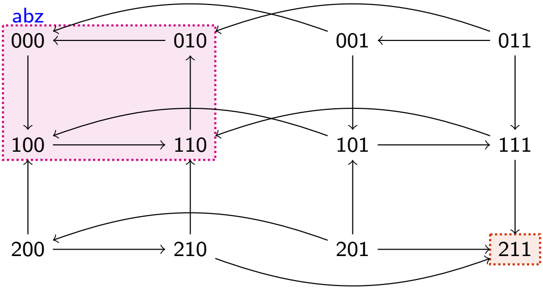
- **Stable state** (state with no successors)

State-graph of a Discrete Network

Several semantics exist regarding the updates:

- Synchronous (deterministic)
- **Asynchronous** (non-deterministic)
- Generalized (even more non-deterministic)

In every case, exponential size in the number of components



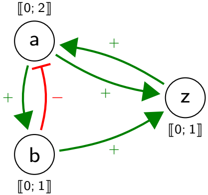
Attractor = minimal set of states from which the dynamics cannot escape
= terminal strongly connected component

- **Stable state** (state with no successors)
- **Complex attractor** (loop or composition of loops)

Static Analysis of Discrete Networks

[Thomas in *Numerical Methods in the Study of Critical Phenomena*, 1981]
[Paulevé & Richard, *Electronic Notes in Theoretical Computer Science* 2012]

Conjectures of René Thomas:

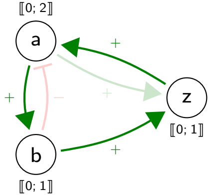


Static Analysis of Discrete Networks

[Thomas in *Numerical Methods in the Study of Critical Phenomena*, 1981]
[Paulevé & Richard, *Electronic Notes in Theoretical Computer Science* 2012]

Conjectures of René Thomas:

- Multiple **stable states** ⇒ positive cycle in the graph



110

111

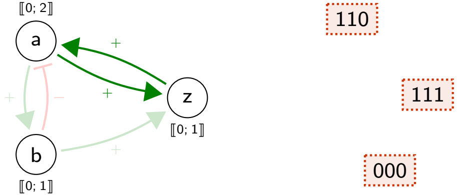
000

Static Analysis of Discrete Networks

[Thomas in *Numerical Methods in the Study of Critical Phenomena*, 1981]
[Paulevé & Richard, *Electronic Notes in Theoretical Computer Science* 2012]

Conjectures of René Thomas:

- Multiple **stable states** ⇒ positive cycle in the graph

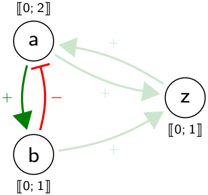


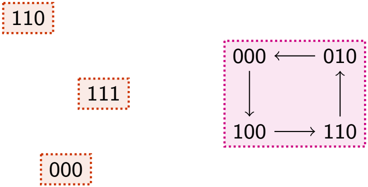
Static Analysis of Discrete Networks

[Thomas in *Numerical Methods in the Study of Critical Phenomena*, 1981]
[Paulevé & Richard, *Electronic Notes in Theoretical Computer Science* 2012]

Conjectures of René Thomas:

- Multiple **stable states** ⇒ positive cycle in the graph
- Sustained oscillations (**complex attractor**) ⇒ negative cycle in the graph



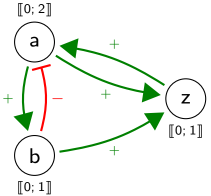


Static Analysis of Discrete Networks

[Thomas in *Numerical Methods in the Study of Critical Phenomena*, 1981]
[Paulevé & Richard, *Electronic Notes in Theoretical Computer Science* 2012]

Conjectures of René Thomas:

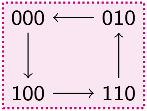
- Multiple **stable states** ⇒ positive cycle in the graph
- Sustained oscillations (**complex attractor**) ⇒ negative cycle in the graph



110

111

000



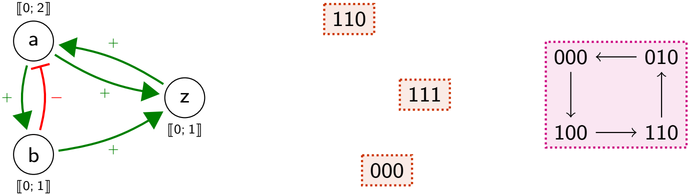
Proofs:
[Remy, Ruet, Thieffry in *Advances in Applied Mathematics*, 2008]
[Richard, *Advances in Applied Mathematics*, 2010]
[Richard, Comet in *Discrete Applied Mathematics*, 2007]

Static Analysis of Discrete Networks

[Thomas in *Numerical Methods in the Study of Critical Phenomena*, 1981]
[Paulevé & Richard, *Electronic Notes in Theoretical Computer Science* 2012]

Conjectures of René Thomas:

- Multiple **stable states** ⇒ positive cycle in the graph
- Sustained oscillations (**complex attractor**) ⇒ negative cycle in the graph



Proofs:
[Remy, Ruet, Thieffry in *Advances in Applied Mathematics*, 2008]
[Richard, *Advances in Applied Mathematics*, 2010]
[Richard, Comet in *Discrete Applied Mathematics*, 2007]

Other results:

- Lower & upper bounds of the number of attractors
- Functionality of the cycles
- Sufficient condition for no stable state / Topological stable states

Dynamic Analysis of Discrete Networks

- These static analysis results are not sufficient to predict the dynamics of independent components.

Examples that cannot be tackled:

1) From the initial state $(a, b, z) = (0, 0, 0)$, is it possible to reach $z = 2$?
2) Does $(0, 0, 0)$ belong to an attractor?
3) What is the set of attractors of the model?

Dynamic Analysis of Discrete Networks

- These static analysis results are not sufficient to predict the dynamics of independent components.

Examples that cannot be tackled:

1) From the initial state $(a, b, z) = (0, 0, 0)$, is it possible to reach $z = 2$?
2) Does $(0, 0, 0)$ belong to an attractor?
3) What is the set of attractors of the model?

- Temporal logics (LTL, CTL, CTL*)

More precise but require to compute the whole state graph

Examples:

1) $(a = 0 \wedge b = 0 \wedge z = 0) \Rightarrow \text{EF}(z = 2)$
2) $(a = 0 \wedge b = 0 \wedge z = 0) \Rightarrow \text{AG}(\text{EF}(a = 0 \wedge b = 0 \wedge z = 0))$
3) ???

Dynamic Analysis of Discrete Networks

- These static analysis results are not sufficient to predict the dynamics of independent components.

Examples that cannot be tackled:
1) From the initial state $(a, b, z) = (0, 0, 0)$, is it possible to reach $z = 2$?
2) Does $(0, 0, 0)$ belong to an attractor?
3) What is the set of attractors of the model?

- Temporal logics (LTL, CTL, CTL*)

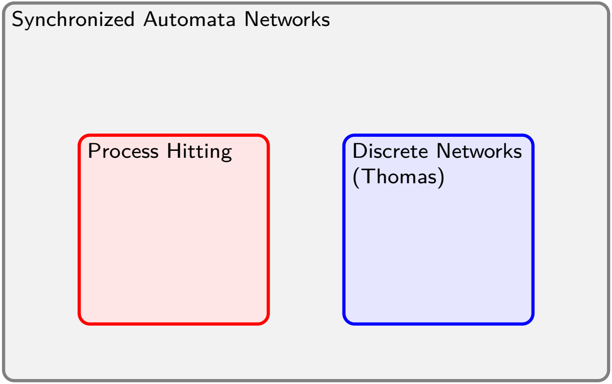
More precise but require to compute the whole state graph

Examples:
1) $(a = 0 \wedge b = 0 \wedge z = 0) \Rightarrow \mathrm{EF}(z = 2)$
2) $(a = 0 \wedge b = 0 \wedge z = 0) \Rightarrow \mathrm{AG}(\mathrm{EF}(a = 0 \wedge b = 0 \wedge z = 0))$
3) ???

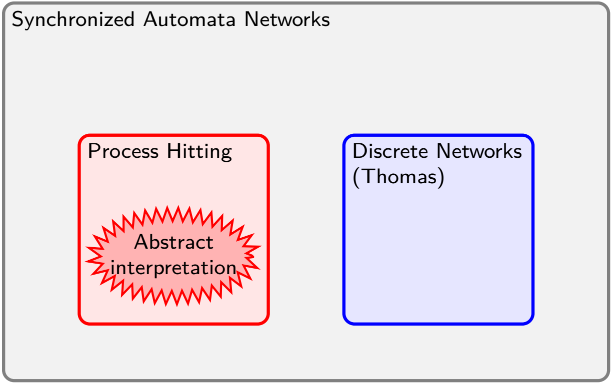
- Applications of CTL and LTL

Check a property on a given model: NuSMV, LibDDD, ...
Create a model for which a property holds: SMBioNet, SPuTNIk, ...
[Bernot, Comet, Richard, Guespin in *Journal of Theoretical Biology*, 2004]

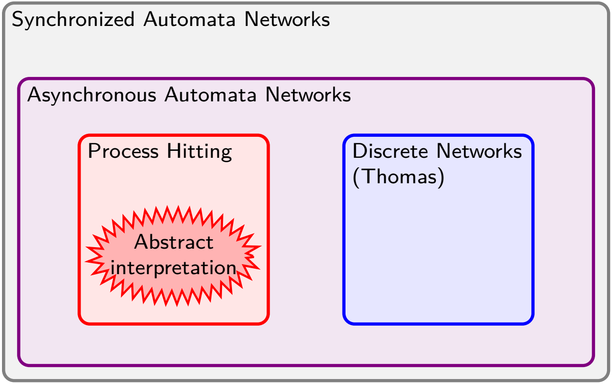
The Enriched Process Hitting



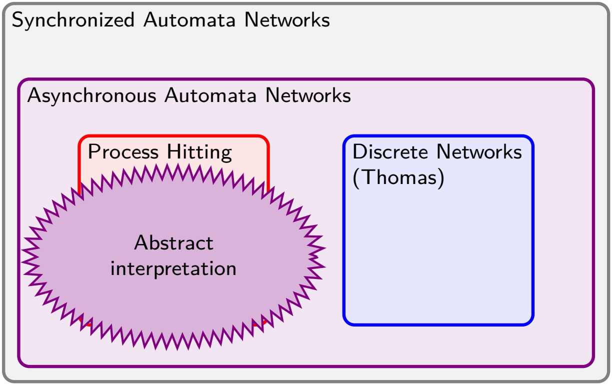
The Enriched Process Hitting



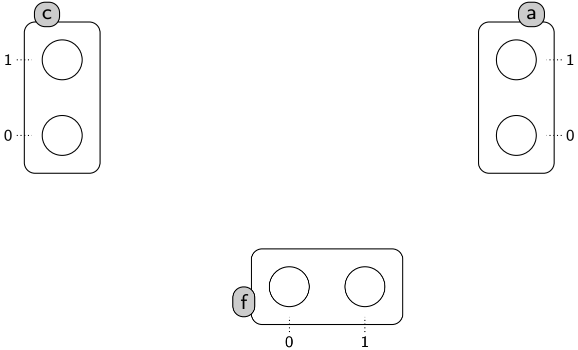
The Enriched Process Hitting



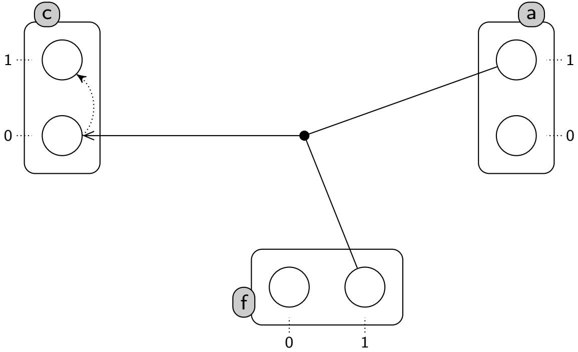
The Enriched Process Hitting



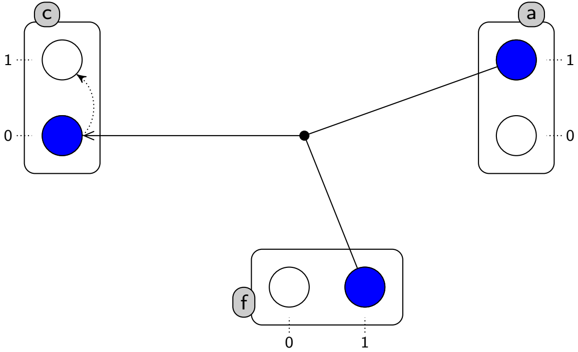
Example of enriched Process Hitting Model



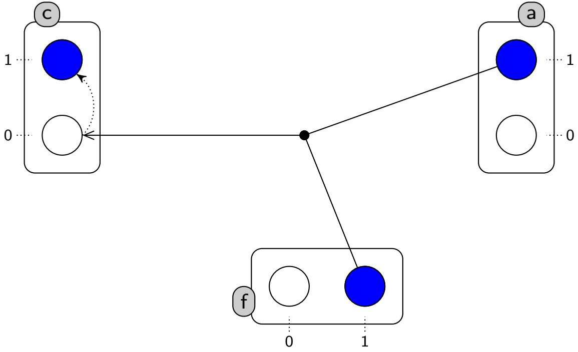
Example of enriched Process Hitting Model



Example of enriched Process Hitting Model

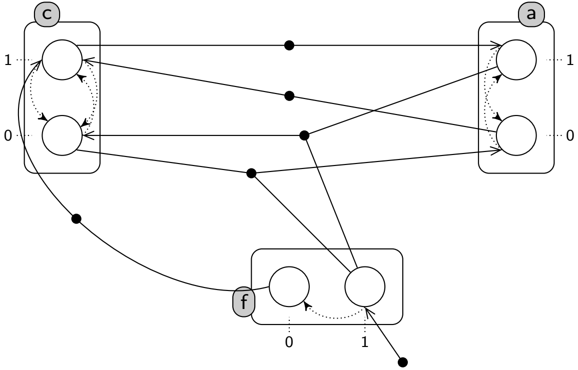


Example of enriched Process Hitting Model

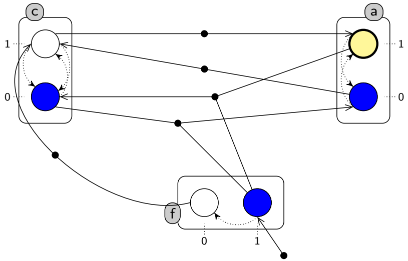


Example of enriched Process Hitting Model

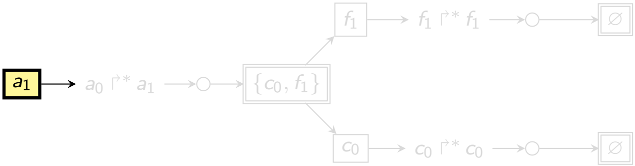
Model from [François *et al.* in Molecular Systems Biology, 2007]



Static analysis

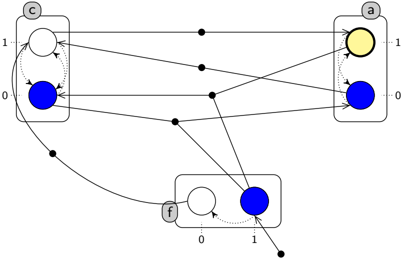


- No conflict
- All leaves are $\emptyset$

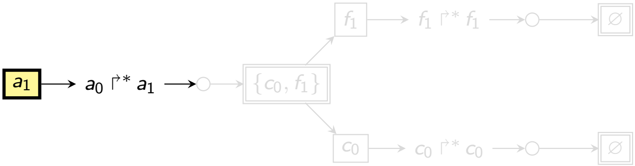


$$\{c_0, f_1\} \to a_0 \upharpoonright a_1$$

Static analysis

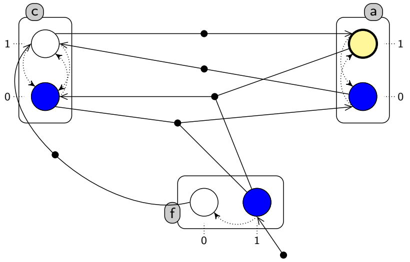


- No conflict
- All leaves are $\boxed{\varnothing}$

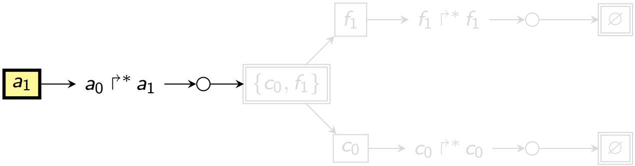


$$\{c_0, f_1\} \to a_0 \upharpoonright a_1$$

Static analysis

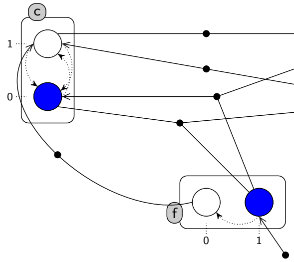


- No conflict
- All leaves are $\boxed{\varnothing}$

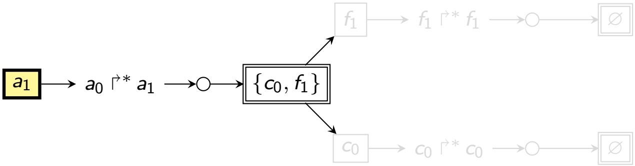


$$\{c_0, f_1\} \to a_0 \upharpoonright a_1$$

Static analysis

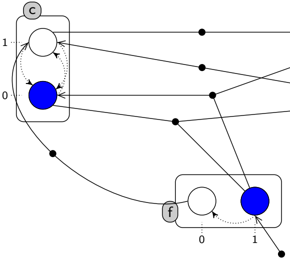


- No conflict
- All leaves are $\boxed{\emptyset}$

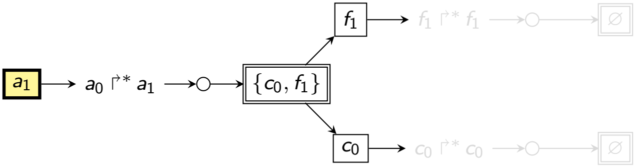


$$\{c_0, f_1\} \to a_0 \upharpoonright a_1$$

Static analysis

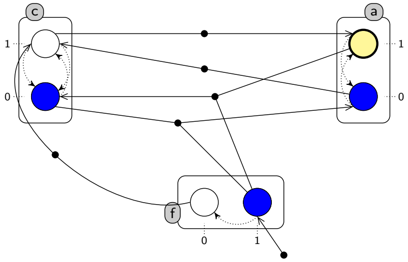


- No conflict
- All leaves are $\emptyset$

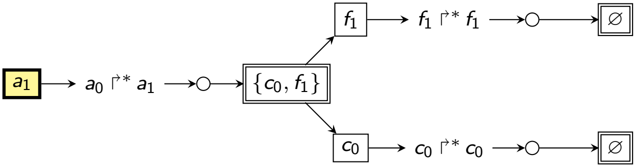


$$\{c_0, f_1\} \rightarrow a_0 \upharpoonright a_1$$

Static analysis

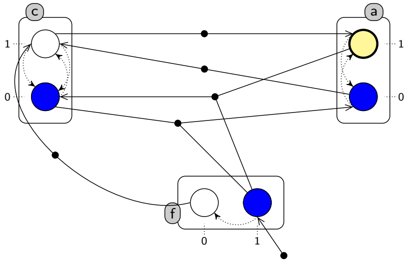


- No conflict
- All leaves are ∅

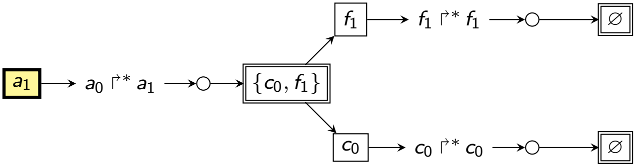


$$\{c_0, f_1\} \rightarrow a_0 \upharpoonright a_1$$

Static analysis

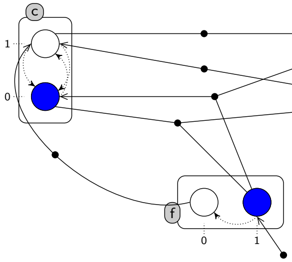


- No conflict
- All leaves are $\boxed{\emptyset}$

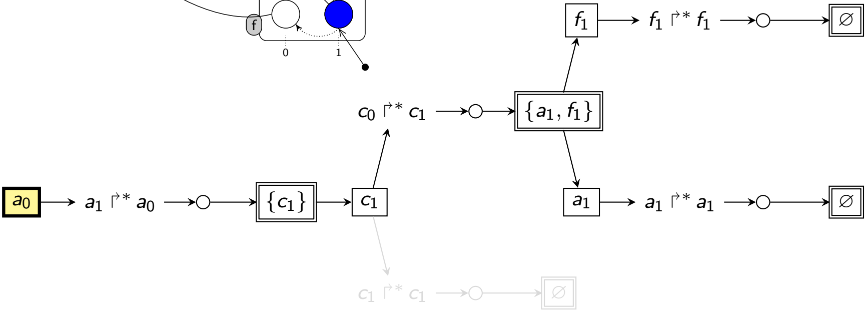


$$\{c_0, f_1\} \rightarrow a_0 \upharpoonright a_1$$

Static analysis

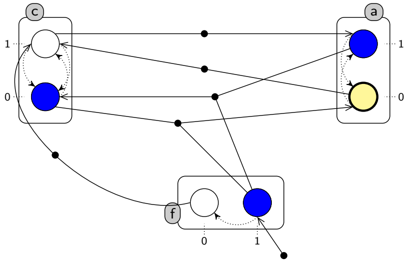


- No conflict
- All leaves are $\emptyset$

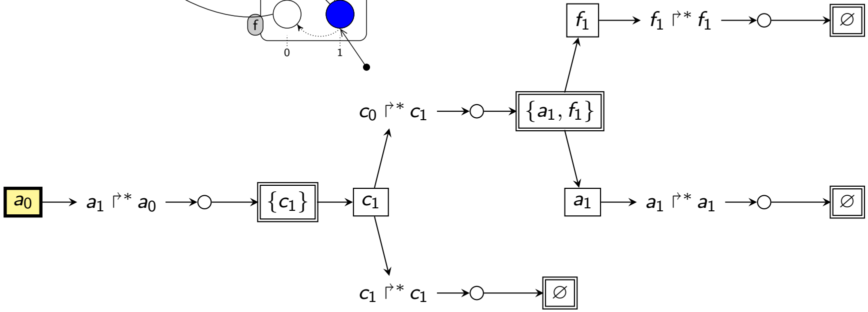


$$\{a_1, f_1\} \rightarrow c_0 \upharpoonright c_1 \;::\; \{c_1\} \rightarrow a_1 \upharpoonright a_0$$

Static analysis

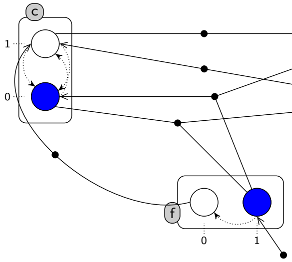


- No conflict
- All leaves are $\boxed{\varnothing}$

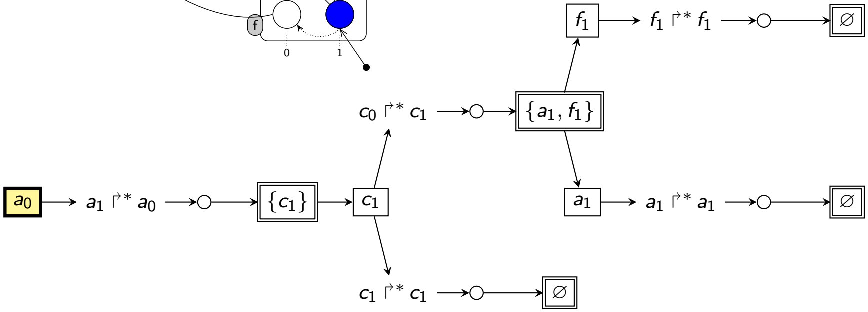


$$\{a_1, f_1\} \rightarrow c_0 \upharpoonright c_1 \;::\; \{c_1\} \rightarrow a_1 \upharpoonright a_0$$

Static analysis

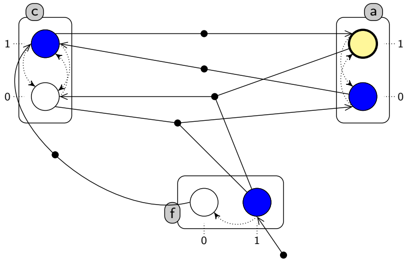


- No conflict
- All leaves are ∅

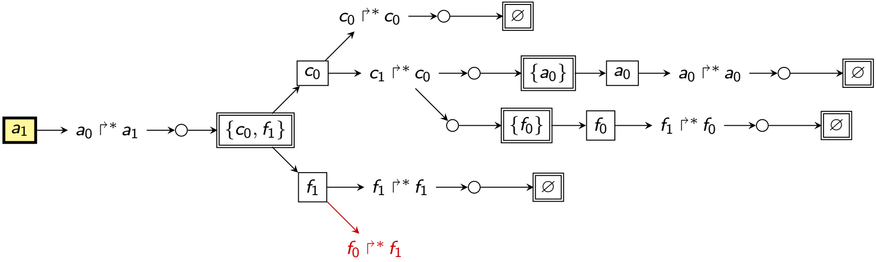


$$\{a_1, f_1\} \rightarrow c_0 \Rsh c_1 \;::\; \{c_1\} \rightarrow a_1 \Rsh a_0$$

Static analysis

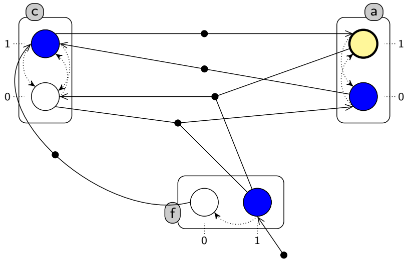


- No conflict
- ~~All leaves are $\emptyset$~~

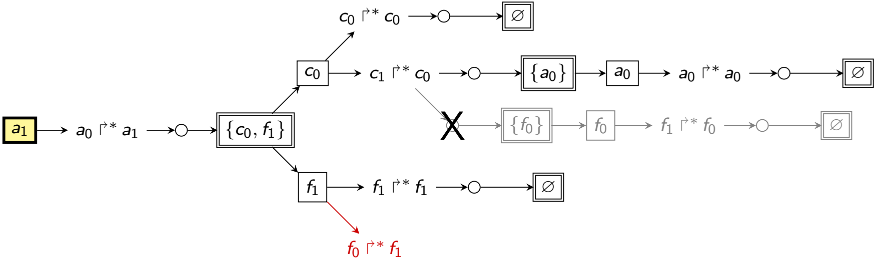


$\{a_0\} \rightarrow c_1 \upharpoonright c_0 \;::\; \{c_0, f_1\} \rightarrow a_0 \upharpoonright a_1$

Static analysis

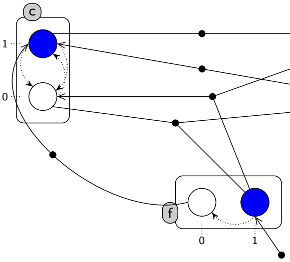


- No conflict
- ~~All leaves are $\boxed{\emptyset}$~~

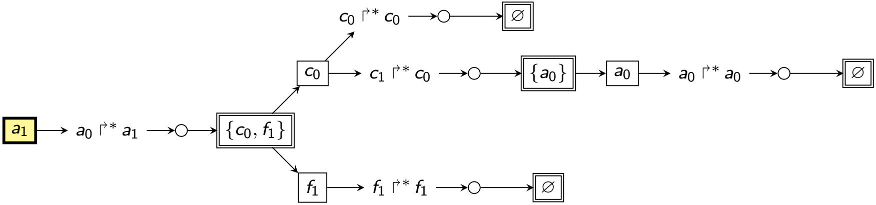


$\{a_0\} \rightarrow c_1 \upharpoonright c_0 \;::\; \{c_0, f_1\} \rightarrow a_0 \upharpoonright a_1$

Static analysis

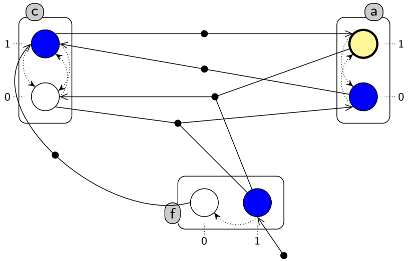


- No conflict
- All leaves are $\boxed{\emptyset}$

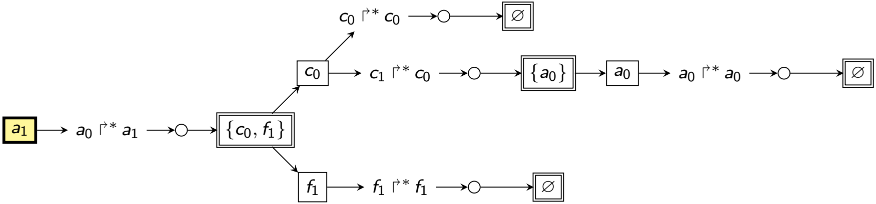


$$\{a_0\} \rightarrow c_1 \upharpoonright c_0 \;::\; \{c_0, f_1\} \rightarrow a_0 \upharpoonright a_1$$

Static analysis



- No conflict
- All leaves are $\boxed{\varnothing}$



$$\{a_0\} \to c_1 \upharpoonright c_0 \;::\; \{c_0, f_1\} \to a_0 \upharpoonright a_1$$

Implementation of the Static Analysis Into PINT

Complexity:

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

Implementation of the Static Analysis Into PINT

Complexity:

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

Makes the study of large networks tractable:

Model	Automata	Actions	States	libddd[1]	GINsim[2]	PINT
egfr20	35	670	2^{64}		$<$1s	**0.02s**
tcrsig40	54	301	2^{73}		∞	**0.02s**
tcrsig94	133	1124	2^{194}	[$>$1min − ∞]		**0.03s**
egfr104	193	2356	2^{320}	[$>$1min − ∞]		**0.16s**

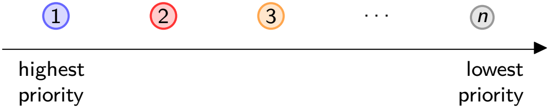
[1] LIP6/Move [Couvreur *et al.*, *Lecture Notes in Computer Science*, 2002]
[2] TAGC/IGC [Chaouiya, Naldi, Thieffry, *Methods in Molecular Biology*, 2012]

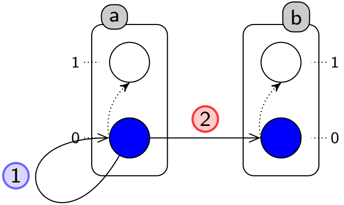
egfr20 : Epithelial Growth Factor Receptor (20 components) [Sahin *et al.*, 2009]
egfr104 : Epithelial Growth Factor Receptor (104 components) [Samaga *et al.*, 2009]
tcrsig40 : T-Cell Receptor (40 composants) [Klamt *et al.*, 2006]
tcrsig94 : T-Cell Receptor (94 composants) [Saez-Rodriguez *et al.*, 2007]

Classes of priorities

[Folschette *et al.* in *Theoretical Computer Science*, 2015b]

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



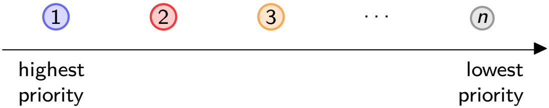


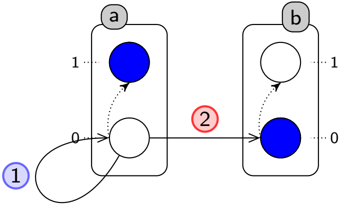
→ b_1 **cannot** be reached

Classes of priorities

[Folschette *et al.* in *Theoretical Computer Science*, 2015b]

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



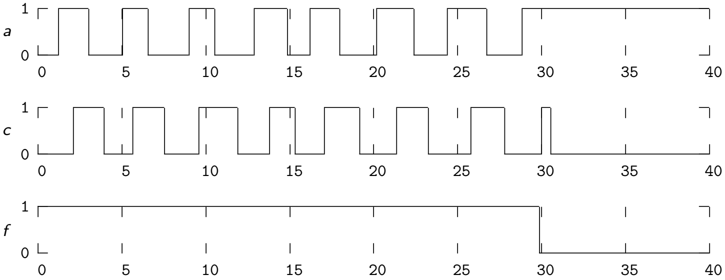


→ b_1 **cannot** be reached

Temporal Simulation

[Paulevé (PhD thesis), 2011]

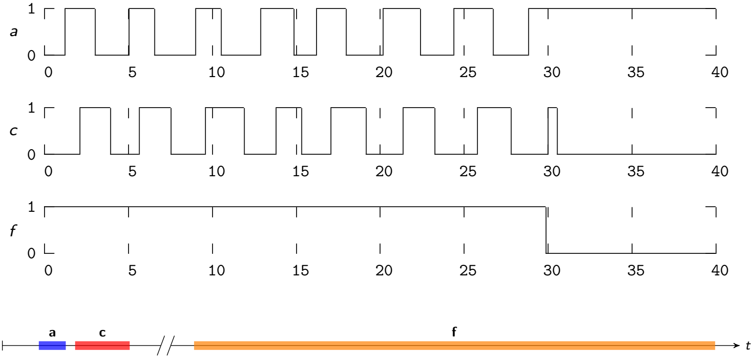
- Simulation with stochastic parameters:



Temporal Simulation

[Paulevé (PhD thesis), 2011]

- Simulation with stochastic parameters:



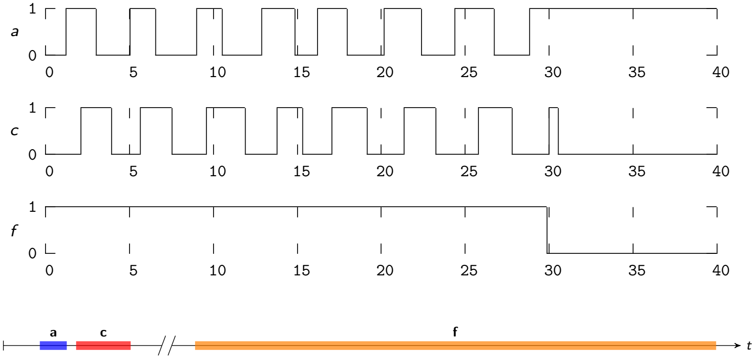
Stochastic parameters:

- **a** $= [0.742; 1.29]$ (mean 1)
- **c** $= [1.48; 2.59]$ (mean 2)
- **f** $= [23.9; 35.4]$ (mean 30)

Temporal Simulation

[Paulevé (PhD thesis), 2011]

- Simulation with stochastic parameters:



Stochastic parameters:

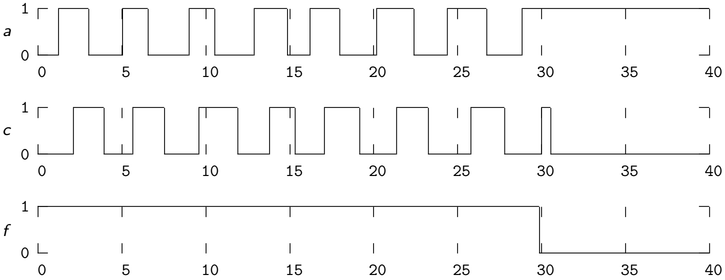
- **a** $= [0.742; 1.29]$ (mean 1)
- **c** $= [1.48; 2.59]$ (mean 2)
- **f** $= [23.9; 35.4]$ (mean 30)

$\Rightarrow \mathbf{a} < \mathbf{c} < \mathbf{f}$

Temporal Simulation

[Paulevé (PhD thesis), 2011]

- Simulation with stochastic parameters:

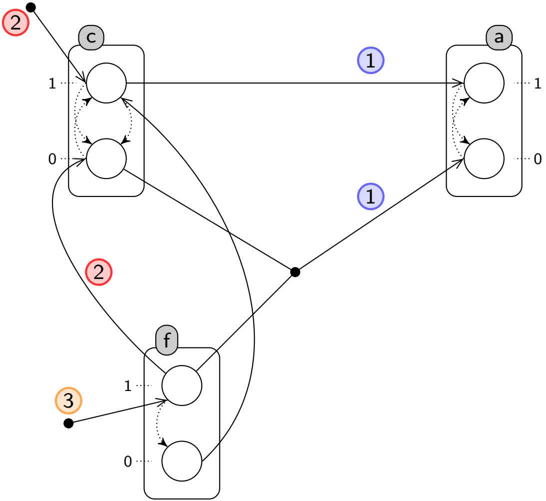


Stochastic parameters:

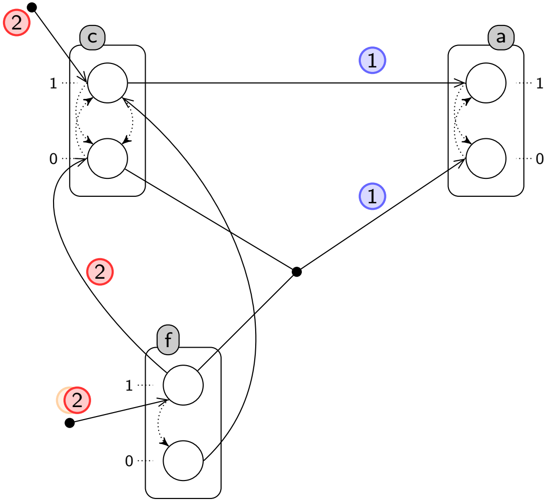
- **a** $= [0.742; 1.29]$ (mean 1) ①
- **c** $= [1.48; 2.59]$ (mean 2) ②
- **f** $= [23.9; 35.4]$ (mean 30) ③

$\Rightarrow \mathbf{a} < \mathbf{c} < \mathbf{f}$

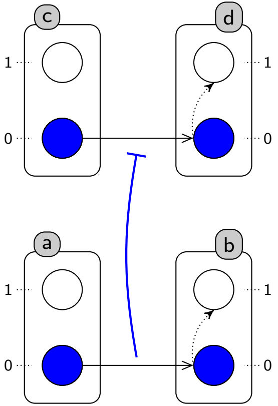
Example with Classes of Priorities



Example with Classes of Priorities



Neutralizing Edges



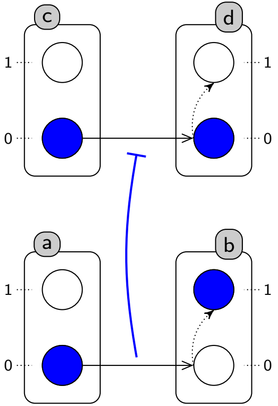
- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to “atomistic priorities”

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

$a_0 \rightarrow b_0 \upharpoonright b_1$ is playable

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

Neutralizing Edges



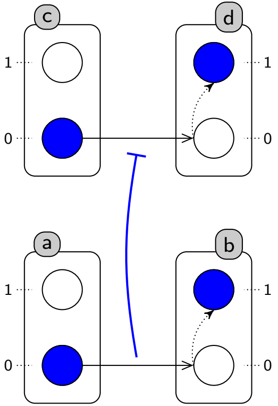
- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to "atomistic priorities"

$c_0 \to d_0 \upharpoonright d_1$ cannot be plays **while**

$a_0 \to b_0 \upharpoonright b_1$ is playable

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

Neutralizing Edges



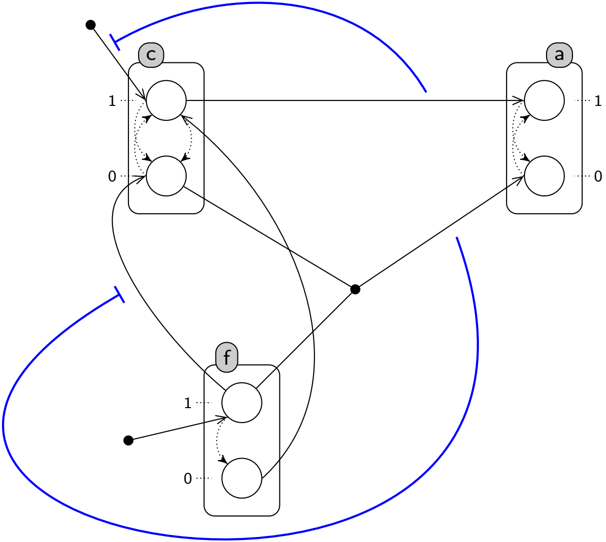
- Integration of temporal data about relative reaction speeds
- Atomistic preemptions between actions similar to "atomistic priorities"

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

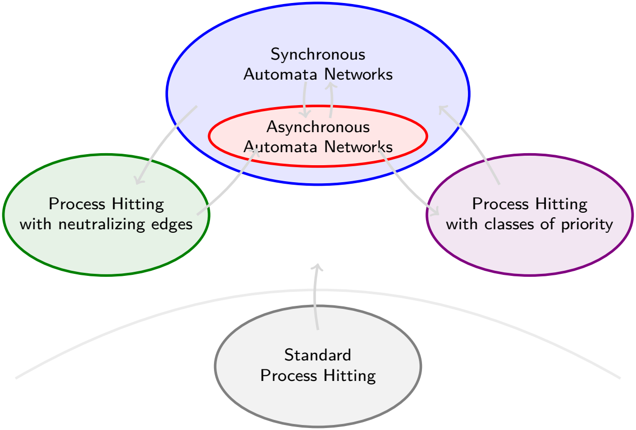
$a_0 \rightarrow b_0 \upharpoonright b_1$ is playable

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

Example with Neutralizing Edges



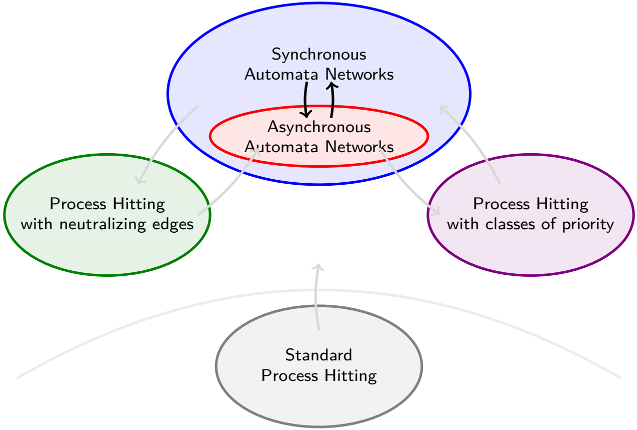
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

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

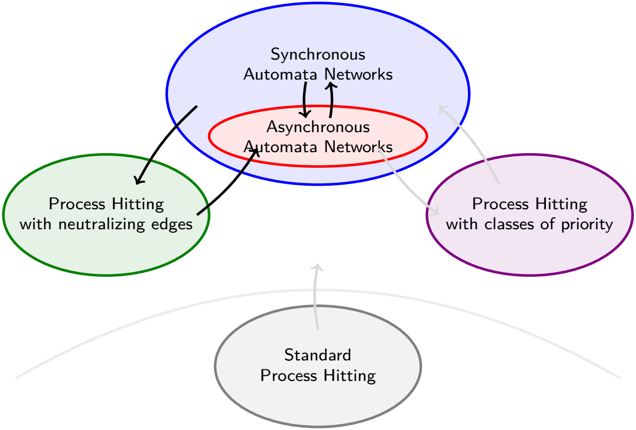
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

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

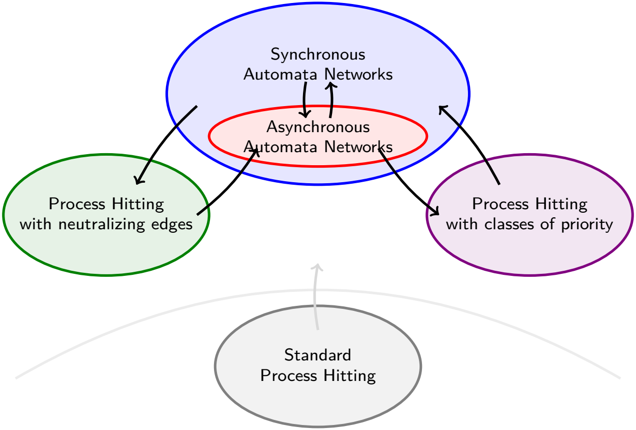
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

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

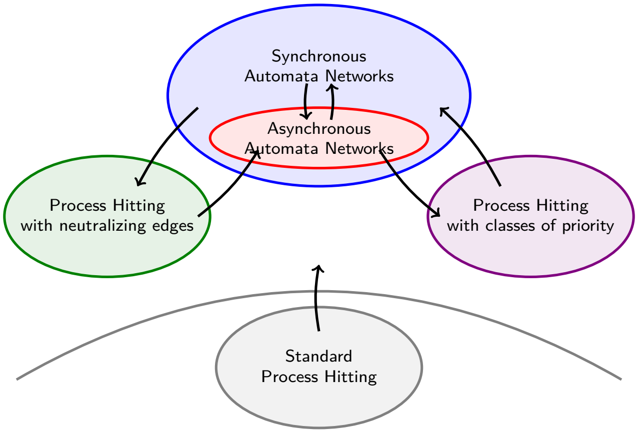
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

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

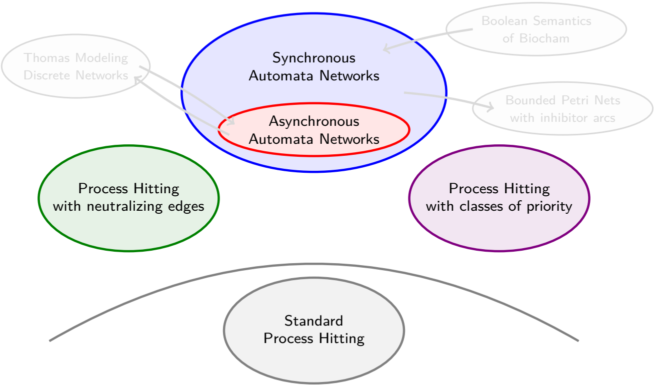
Equivalence Between Process Hitting Extensions



All developed enrichments have the same expressivity

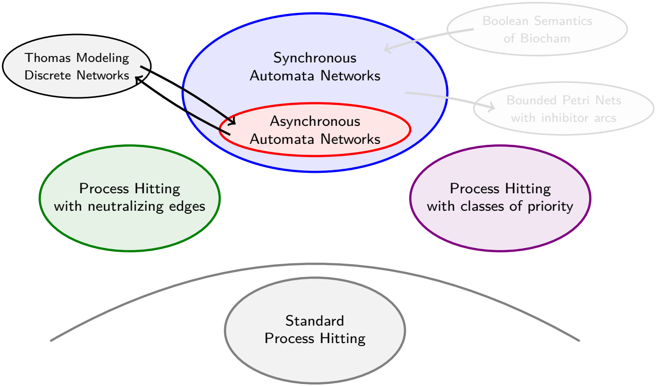
- Expressive power improved
- Can always be translated to the canonical form
- But sometimes at the cost of an exponential translation

Translation From and To Other Discrete Models



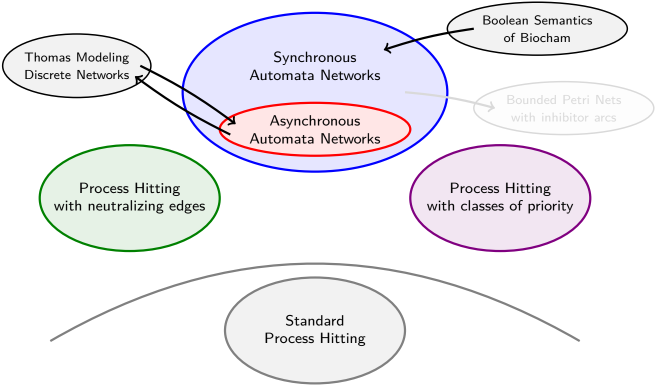
- Equivalence with Discrete Networks / Thomas modeling
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



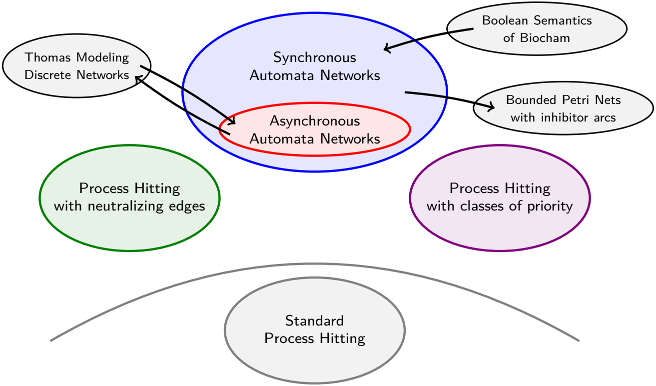
- Equivalence with Discrete Networks / Thomas modeling
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



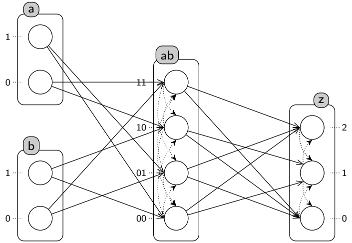
- Equivalence with Discrete Networks / Thomas modeling
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

Translation From and To Other Discrete Models



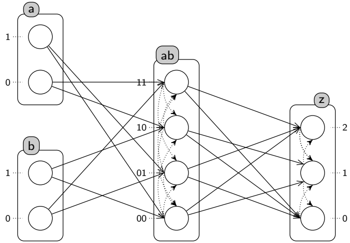
- Equivalence with Discrete Networks / Thomas modeling
- Translation towards (bounded) Petri nets with inhibitor arcs
- Translation from the Boolean semantics of Biocham

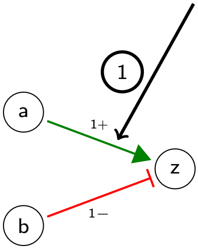
Inferring a BRN with Thomas' parameters



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

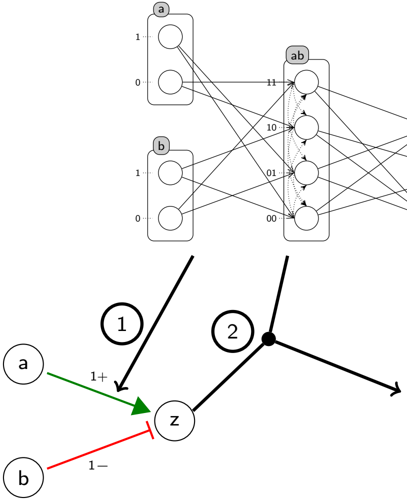
Inferring a BRN with Thomas' parameters





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

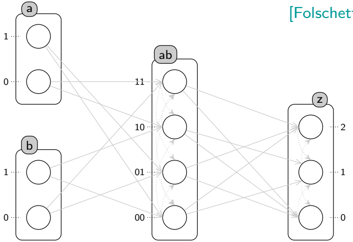
Inferring a BRN with Thomas' parameters

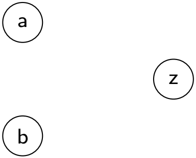


ω		$k_{z,\omega}$
a	b	
$-$	$+$	1
$-$	$-$	0
$+$	$+$	2
$+$	$-$	1

Inferring the Interaction Graph

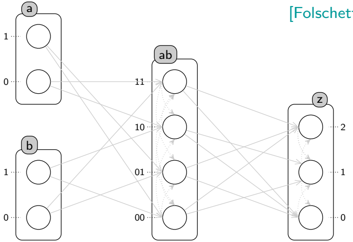
[Folschette *et al.* in *Theoretical Computer Science*, 2015a]

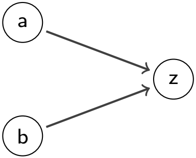




Inferring the Interaction Graph

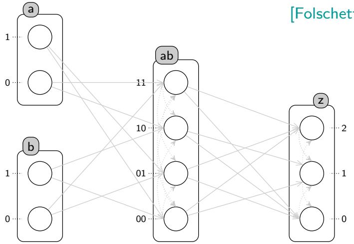
[Folschette *et al.* in *Theoretical Computer Science*, 2015a]

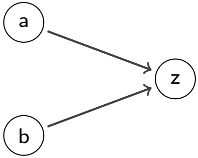




Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]

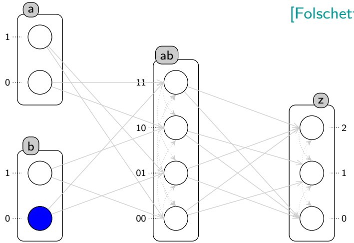


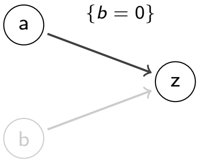


→ **Exhaustive search in all possible configurations**

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



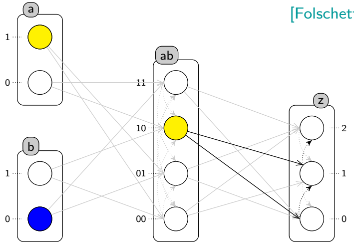


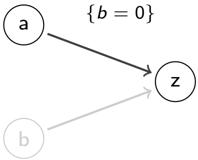
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



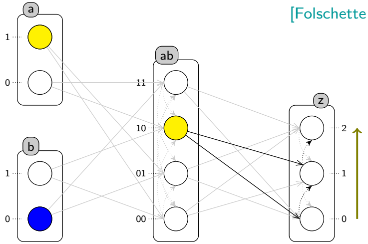


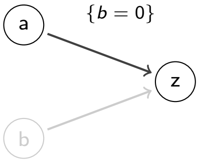
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



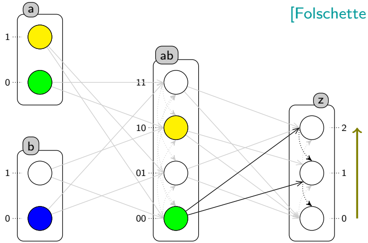


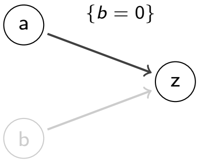
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



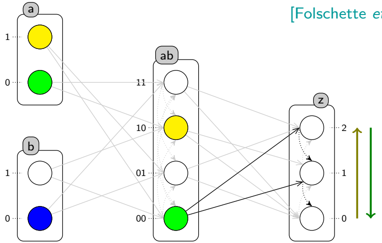


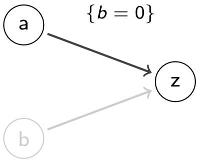
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



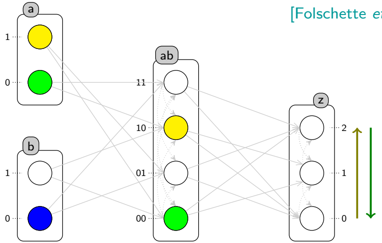


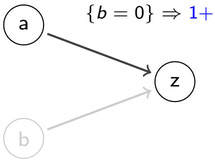
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



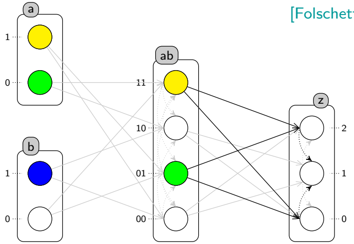


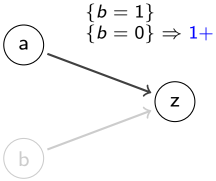
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



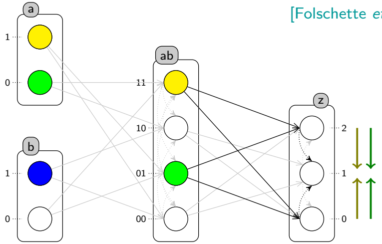


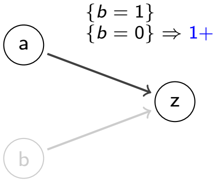
$\rightarrow$ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



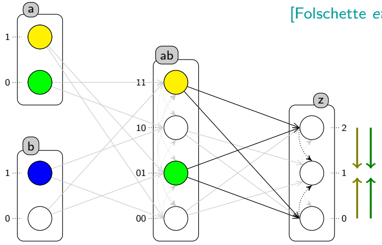


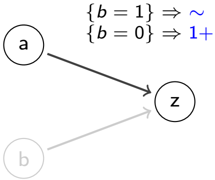
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



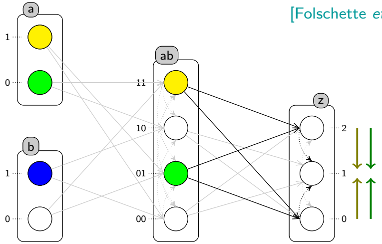


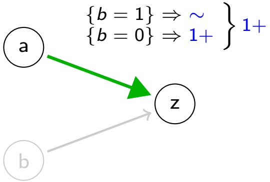
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



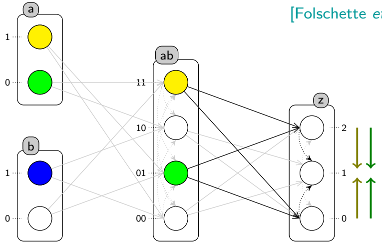


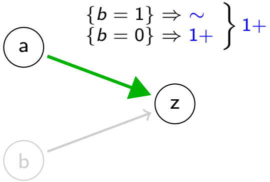
→ **Exhaustive search in all possible configurations**

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

Inferring the Interaction Graph

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]





→ **Exhaustive search in all possible configurations**

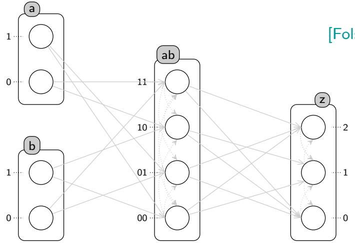
1. Pick one regulator [a], and choose an active process for all the others [b_0].
2. Change the active process of the regulator [a_0, a_1] and watch the **evolution**.
3. Conclude locally: $(a_0 \upharpoonright a_1 \Rightarrow z_0 \upharpoonright z_2) \Rightarrow$ activation $(+)$ & threshold $= 1$.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

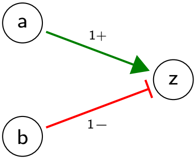
Problematic cases:

→ No focal processes (cycle)
→ Opposite influences $(+ \ \& \ -)$
} ⇒ Unsigned edge

Inferring Parameters

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]

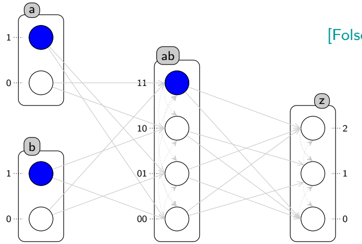


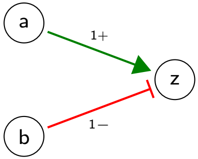


ω		$k_{z,\omega}$
a	b	
$-$	$+$	
$-$	$-$	
$+$	$+$	
$+$	$-$	

Inferring Parameters

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



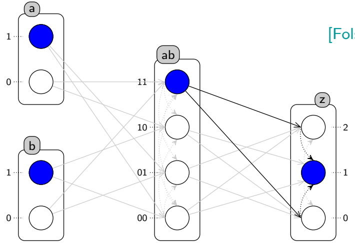


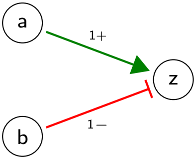
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	

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

Inferring Parameters

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



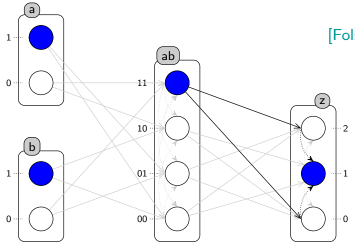


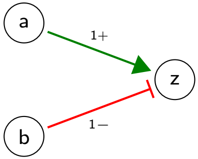
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	

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

Inferring Parameters

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]



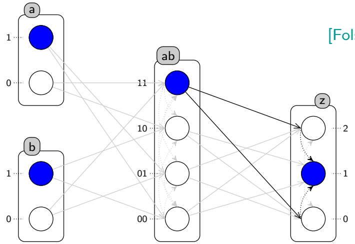


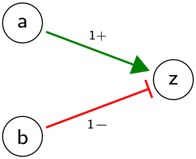
ω		$k_{z,\omega}$
a	b	
−	+	
−	−	
+	+	
+	−	1

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

Inferring Parameters

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]





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

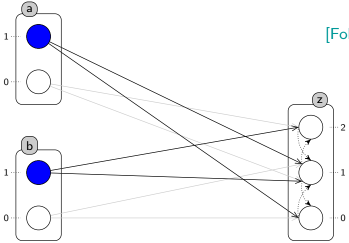
1. For each configuration of resources $[\omega = \{a^+, b^-\}]$
   find the **focal processes**. If possible, conclude. $[k_{z,\{a^+,b^-\}} = 1]$

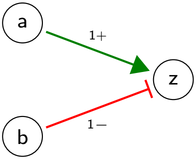
   Inconclusive cases:

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

Inferring Parameters

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]





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

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

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

2. If some parameters could not be inferred, enumerate all admissible parametrizations, regarding:
   - Biological constraints [Bernot *et al.* in *Concurrent Models in Molecular Biology*, 2007]
   - The dynamics of the Process Hitting

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

Translation to Thomas Modeling

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]

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

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

Translation to Thomas Modeling

[Folschette *et al.* in *Theoretical Computer Science*, 2015a]

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

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

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

egfr20 : Epithelial Growth Factor Receptor (20 components) [Sahin *et al.*, 2009]
egfr104 : Epithelial Growth Factor Receptor (104 components) [Samaga *et al.*, 2009]
tcrsig40 : T-Cell Receptor (40 composants) [Klamt *et al.*, 2006]
tcrsig94 : T-Cell Receptor (94 composants) [Saez-Rodriguez *et al.*, 2007]

The Modal μ-calculus

LTL: Example of the "Until" operator

$p\ U\ q \equiv$ "Either q, or p and the next state also verifies $p\ U\ q$"

$\Rightarrow$ Implicit fixed point

(Modal) μ-calculus makes such fixed points explicit

$$\varphi = p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

- Basic property: p ("p is verified in this node")
- Modal operators: $\Box$ ("for all successors"), $\Diamond$ ("there exists a successor")
- Fixed points: μ (least fixed point), ν (greatest fixed point)

Polyadic (modal) μ-calculus allows to manipulate several tokens in parallel

$$\varphi = p_i \mid i \leftarrow j \mid i = j \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond_i\varphi \mid \Box_i\varphi \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

- Token manipulation:
  $i = j$ ("tokens i and j point to the same node")
  $i \leftarrow j$ ("move token i to the position of token j")

The Modal μ-calculus

LTL: Example of the "Until" operator

$p\ U\ q \equiv$ "Either q, or p and the next state also verifies $p\ U\ q$"

$\Rightarrow$ Implicit fixed point

(Modal) μ-calculus makes such fixed points explicit

$$\varphi = p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

- Basic property: p ("p is verified in this node")
- Modal operators: $\Box$ ("for all successors"), $\Diamond$ ("there exists a successor")
- Fixed points: μ (least fixed point), ν (greatest fixed point)

Polyadic (modal) μ-calculus allows to manipulate several tokens in parallel

$$\varphi = p_i \mid i \leftarrow j \mid i = j \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond_i\varphi \mid \Box_i\varphi \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

- Token manipulation:
  $i = j$ ("tokens i and j point to the same node")
  $i \leftarrow j$ ("move token i to the position of token j")

The Modal μ-calculus

LTL: Example of the "Until" operator
$p\ U\ q \equiv$ "Either q, or p and the next state also verifies $p\ U\ q$"
$\Rightarrow$ Implicit fixed point

(Modal) μ-calculus makes such fixed points explicit

$$\varphi = p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

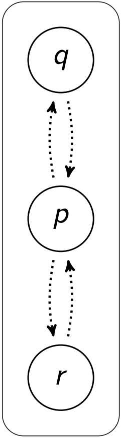
- Basic property: p ("p is verified in this node")
- Modal operators: $\Box$ ("for all successors"), $\Diamond$ ("there exists a successor")
- Fixed points: μ (least fixed point), ν (greatest fixed point)

Polyadic (modal) μ-calculus allows to manipulate several tokens in parallel

$$\varphi = p_i \mid i \leftarrow j \mid i = j \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond_i\varphi \mid \Box_i\varphi \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

- Token manipulation:
  $i = j$ ("tokens i and j point to the same node")
  $i \leftarrow j$ ("move token i to the position of token j")

Examples with Modal μ-calculus



No tokens: only one evolution is studied

Atomic property (p, q, r)

$[\![p]\!] = \{p\}$

$[\![q \vee r]\!] = \{q; r\}$

Possible future ("may")

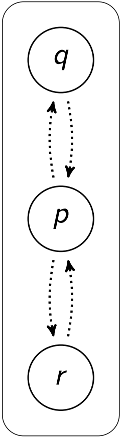
$[\![\Diamond\ q]\!] = \{p\}$

Necessary future ("must")

$[\![\Box\ q]\!] = \varnothing$

$[\![\Box\ p]\!] = \{q; r\}$

Examples with Modal µ-calculus



No tokens: only one evolution is studied

Atomic property (p, q, r)

$[\![p]\!] = \{p\}$

$[\![q \vee r]\!] = \{q; r\}$

Possible future ("may")

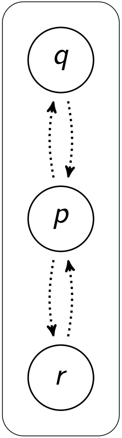
$[\![\Diamond\ q]\!] = \{p\}$

Necessary future ("must")

$[\![\Box\ q]\!] = \varnothing$

$[\![\Box\ p]\!] = \{q; r\}$

Examples with Modal μ-calculus



No tokens: only one evolution is studied

Atomic property (p, q, r)

$[\![p]\!] = \{p\}$
$[\![q \vee r]\!] = \{q; r\}$

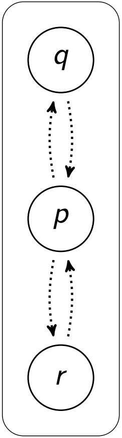
Possible future ("may")

$[\![\Diamond\ q]\!] = \{p\}$

Necessary future ("must")

$[\![\Box\ q]\!] = \varnothing$
$[\![\Box\ p]\!] = \{q; r\}$

Examples with Modal μ-calculus



No tokens: only one evolution is studied

Atomic property (p, q, r)

$[\![p]\!] = \{p\}$

$[\![q \vee r]\!] = \{q; r\}$

Possible future ("may")

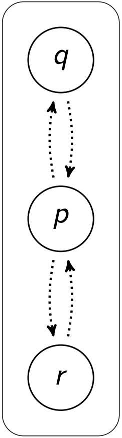
$[\![\Diamond\ q]\!] = \{p\}$

Necessary future ("must")

$[\![\Box\ q]\!] = \varnothing$

$[\![\Box\ p]\!] = \{q; r\}$

Examples with Modal μ-calculus



No tokens: only one evolution is studied

Atomic property (p, q, r)

$[\![p]\!] = \{p\}$

$[\![q \vee r]\!] = \{q; r\}$

Possible future ("may")

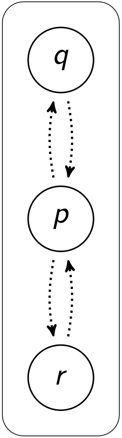
$[\![\Diamond\ q]\!] = \{p\}$

Necessary future ("must")

$[\![\Box\ q]\!] = \varnothing$

$[\![\Box\ p]\!] = \{q; r\}$

Examples with Polyadic μ-calculus



Atomic property (p, q, r)

$[\![p_1 \wedge r_2]\!] = \{(p, r)\}$

$[\![p_1]\!] = \{(p, p); (p, q); (p, r)\}$

Token affectation ($i \leftarrow j$)

$[\![\{2 \leftarrow 1\}\ p_1 \wedge p_2]\!] = \{(p, p); (p, q); (p, r)\}$

Token comparison ($i = j$)

$[\![1 = 2]\!] = \{(p, p); (q, q); (r, r)\}$

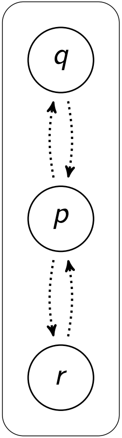
Possible future ("may")

$[\![\Diamond_1\ q]\!] = \{(p, p); (p, q); (p, r)\}$

Necessary future ("must")

$[\![\Box_1\ q]\!] = \varnothing$

Examples with Polyadic μ-calculus



Atomic property (p, q, r)

$[\![p_1 \wedge r_2]\!] = \{(p, r)\}$

$[\![p_1]\!] = \{(p, p); (p, q); (p, r)\}$

Token affectation ($i \leftarrow j$)

$[\![\{2 \leftarrow 1\}\ p_1 \wedge p_2]\!] = \{(p, p); (p, q); (p, r)\}$

Token comparison ($i = j$)

$[\![1 = 2]\!] = \{(p, p); (q, q); (r, r)\}$

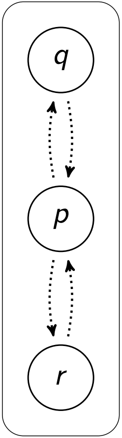
Possible future ("may")

$[\![\Diamond_1\ q]\!] = \{(p, p); (p, q); (p, r)\}$

Necessary future ("must")

$[\![\Box_1\ q]\!] = \emptyset$

Examples with Polyadic μ-calculus



Atomic property (p, q, r)

$[\![p_1 \wedge r_2]\!] = \{(p, r)\}$

$[\![p_1]\!] = \{(p, p); (p, q); (p, r)\}$

Token affectation ($i \leftarrow j$)

$[\![\{2 \leftarrow 1\}\ p_1 \wedge p_2]\!] = \{(p, p); (p, q); (p, r)\}$

Token comparison ($i = j$)

$[\![1 = 2]\!] = \{(p, p); (q, q); (r, r)\}$

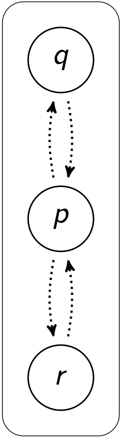
Possible future (“may”)

$[\![\Diamond_1\ q]\!] = \{(p, p); (p, q); (p, r)\}$

Necessary future (“must”)

$[\![\Box_1\ q]\!] = \varnothing$

Examples with Polyadic μ-calculus



Atomic property (p, q, r)

$[\![p_1 \wedge r_2]\!] = \{(p, r)\}$

$[\![p_1]\!] = \{(p, p); (p, q); (p, r)\}$

Token affectation ($i \leftarrow j$)

$[\![\{2 \leftarrow 1\}\ p_1 \wedge p_2]\!] = \{(p, p); (p, q); (p, r)\}$

Token comparison ($i = j$)

$[\![1 = 2]\!] = \{(p, p); (q, q); (r, r)\}$

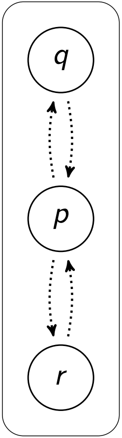
Possible future ("may")

$[\![\Diamond_1\ q]\!] = \{(p, p); (p, q); (p, r)\}$

Necessary future ("must")

$[\![\Box_1\ q]\!] = \varnothing$

Examples with Polyadic μ-calculus



Atomic property (p, q, r)

$[\![p_1 \wedge r_2]\!] = \{(p, r)\}$

$[\![p_1]\!] = \{(p, p); (p, q); (p, r)\}$

Token affectation ($i \leftarrow j$)

$[\![\{2 \leftarrow 1\}\ p_1 \wedge p_2]\!] = \{(p, p); (p, q); (p, r)\}$

Token comparison ($i = j$)

$[\![1 = 2]\!] = \{(p, p); (q, q); (r, r)\}$

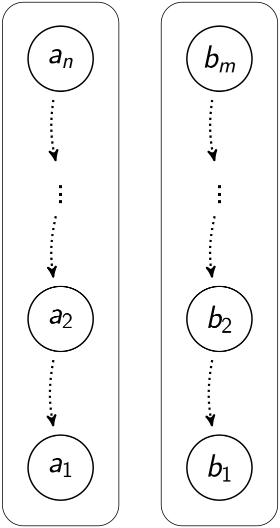
Possible future (“may”)

$[\![\Diamond_1\ q]\!] = \{(p, p); (p, q); (p, r)\}$

Necessary future (“must”)

$[\![\Box_1\ q]\!] = \varnothing$

Examples with Polyadic μ-calculus



Least fixed point (μ)

$\phi = \mu X.(\Box_1 \bot \wedge \Box_2 \bot) \vee \Diamond_1 \Diamond_2 X$

Iterations:

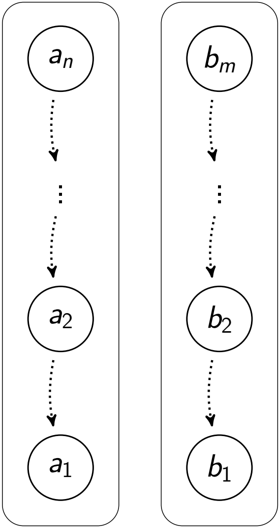
$[\![\phi]\!]_0 = \varnothing$
$[\![\phi]\!]_1 = \{(a_1, b_1)\}$
$[\![\phi]\!]_2 = \{(a_1, b_1); (a_2, b_2)\}$
$[\![\phi]\!]_3 = \{(a_1, b_1); (a_2, b_2); (a_3, b_3)\}$
⋮

Generalization:

$[\![\phi]\!] = \{(a_i, b_i) \mid i \in [1; \min(m, n)]\}$

Idea: use one (or n) token per automata

Examples with Polyadic μ-calculus



Least fixed point (μ)

$\phi = \mu X.(\Box_1 \bot \wedge \Box_2 \bot) \vee \Diamond_1 \Diamond_2 X$

Iterations:

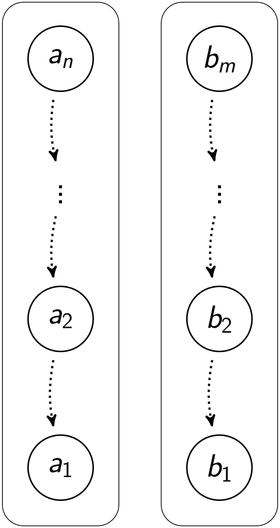
$[\![\phi]\!]_0 = \varnothing$
$[\![\phi]\!]_1 = \{(a_1, b_1)\}$
$[\![\phi]\!]_2 = \{(a_1, b_1); (a_2, b_2)\}$
$[\![\phi]\!]_3 = \{(a_1, b_1); (a_2, b_2); (a_3, b_3)\}$
⋮

Generalization:

$[\![\phi]\!] = \{(a_i, b_i) \mid i \in [1; \min(m, n)]\}$

Idea: use one (or n) token per automata

Examples with Polyadic μ-calculus



Least fixed point (μ)

$$\phi = \mu X.(\Box_1 \bot \wedge \Box_2 \bot) \vee \Diamond_1 \Diamond_2 X$$

Iterations:

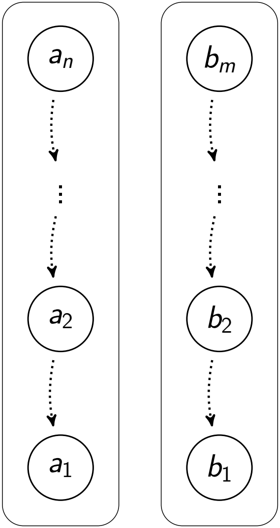
$$[\![\phi]\!]_0 = \varnothing$$
$$[\![\phi]\!]_1 = \{(a_1, b_1)\}$$
$$[\![\phi]\!]_2 = \{(a_1, b_1); (a_2, b_2)\}$$
$$[\![\phi]\!]_3 = \{(a_1, b_1); (a_2, b_2); (a_3, b_3)\}$$
$$\vdots$$

Generalization:

$$[\![\phi]\!] = \{(a_i, b_i) \mid i \in [1; \min(m, n)]\}$$

Idea: use one (or n) token per automata

Examples with Polyadic μ-calculus



Least fixed point (μ)

$\phi = \mu X.(\Box_1 \bot \wedge \Box_2 \bot) \vee \Diamond_1 \Diamond_2 X$

Iterations:

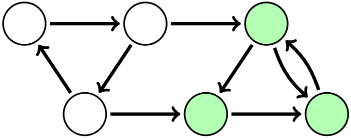
$[\![\phi]\!]_0 = \varnothing$
$[\![\phi]\!]_1 = \{(a_1, b_1)\}$
$[\![\phi]\!]_2 = \{(a_1, b_1); (a_2, b_2)\}$
$[\![\phi]\!]_3 = \{(a_1, b_1); (a_2, b_2); (a_3, b_3)\}$
$\vdots$

Generalization:

$[\![\phi]\!] = \{(a_i, b_i) \mid i \in [1; \min(m, n)]\}$

Idea: use one (or n) token per automata

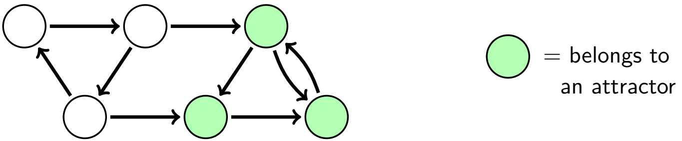
Search for Attractors with Polyadic μ-calculus



$$\varphi_{\text{att}} = \{\boldsymbol{y} \leftarrow \boldsymbol{x}\}\nu W.\underbrace{\underbrace{(\mu Z.\,(\boldsymbol{x} = \boldsymbol{y}) \vee (\Diamond_{\boldsymbol{x}} Z))}_{\varphi_{\text{reach}}} \wedge (\Box_{\boldsymbol{x}} W)}_{\varphi_{\text{explore}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s; t) \mid s \rightarrow^* t\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{y}$"
- $[\![\varphi_{\text{explore}}]\!] = \{(s; t) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* t\}$
  $\varphi_{\text{explore}} \equiv$ "All successors of $\boldsymbol{x}$ can reach $\boldsymbol{y}$"
- $[\![\varphi_{\text{att}}]\!] = \{(s; s) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* s\}$
  $\varphi_{\text{att}} \equiv$ "$\boldsymbol{x}$ belongs to an attractor"

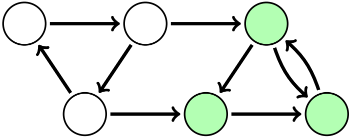
Search for Attractors with Polyadic μ-calculus



$$\varphi_{\text{att}} = \{\boldsymbol{y} \leftarrow \boldsymbol{x}\} \nu W . \underbrace{\underbrace{(\mu Z . (\boldsymbol{x} = \boldsymbol{y}) \vee (\Diamond_{\boldsymbol{x}} Z))}_{\varphi_{\text{reach}}} \wedge (\Box_{\boldsymbol{x}} W)}_{\varphi_{\text{explore}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s; t) \mid s \rightarrow^* t\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{y}$"
- $[\![\varphi_{\text{explore}}]\!] = \{(s; t) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* t\}$
  $\varphi_{\text{explore}} \equiv$ "All successors of $\boldsymbol{x}$ can reach $\boldsymbol{y}$"
- $[\![\varphi_{\text{att}}]\!] = \{(s; s) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* s\}$
  $\varphi_{\text{att}} \equiv$ "$\boldsymbol{x}$ belongs to an attractor"

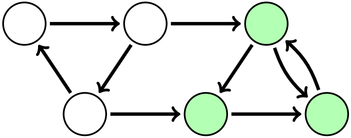
Search for Attractors with Polyadic μ-calculus



$$\varphi_{\text{att}} = \{\boldsymbol{y} \leftarrow \boldsymbol{x}\}\nu W.\underbrace{\underbrace{(\mu Z.\,(\boldsymbol{x} = \boldsymbol{y}) \vee (\Diamond_{\boldsymbol{x}} Z))}_{\varphi_{\text{reach}}} \wedge (\Box_{\boldsymbol{x}} W)}_{\varphi_{\text{explore}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s; t) \mid s \rightarrow^* t\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{y}$"
- $[\![\varphi_{\text{explore}}]\!] = \{(s; t) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* t\}$
  $\varphi_{\text{explore}} \equiv$ "All successors of $\boldsymbol{x}$ can reach $\boldsymbol{y}$"
- $[\![\varphi_{\text{att}}]\!] = \{(s; s) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* s\}$
  $\varphi_{\text{att}} \equiv$ "$\boldsymbol{x}$ belongs to an attractor"

Search for Attractors with Polyadic μ-calculus

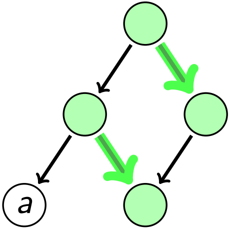


= belongs to an attractor

$$\varphi_{\text{att}} = \{\boldsymbol{y} \leftarrow \boldsymbol{x}\}\nu W.\underbrace{\underbrace{(\mu Z.\,(\boldsymbol{x} = \boldsymbol{y}) \vee (\Diamond_{\boldsymbol{x}} Z))}_{\varphi_{\text{reach}}} \wedge (\Box_{\boldsymbol{x}} W)}_{\varphi_{\text{explore}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s; t) \mid s \rightarrow^* t\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{y}$"
- $[\![\varphi_{\text{explore}}]\!] = \{(s; t) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* t\}$
  $\varphi_{\text{explore}} \equiv$ "All successors of $\boldsymbol{x}$ can reach $\boldsymbol{y}$"
- $[\![\varphi_{\text{att}}]\!] = \{(s; s) \mid \forall s', s \rightarrow^* s' \Rightarrow s' \rightarrow^* s\}$
  $\varphi_{\text{att}} \equiv$ "$\boldsymbol{x}$ belongs to an attractor"

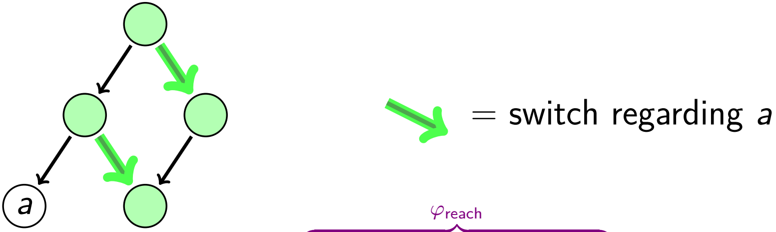
Search for Switches with Polyadic μ-calculus



$$\varphi_{\text{switch}}(\boldsymbol{a}) = \overbrace{\left(\mu W.(\boldsymbol{x} = \boldsymbol{a}) \vee (\Diamond_{\boldsymbol{x}} W)\right)}^{\varphi_{\text{reach}}} \wedge$$
$$\Diamond_{\boldsymbol{x}}\{\boldsymbol{x} \leftarrow \boldsymbol{y}\}\underbrace{\left(\nu Z.\neg(\boldsymbol{y} = \boldsymbol{a}) \wedge (\Box_{\boldsymbol{y}} Z)\right)}_{\varphi_{\text{noreach}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s; t) \mid s \to^* a\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{noreach}}]\!] = \{(s; t) \mid \neg(t \to^* a)\}$
  $\varphi_{\text{noreach}} \equiv$ "There exists no path from $\boldsymbol{y}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{switch}}]\!] = \{(s; t) \mid s \to t \wedge s \to^* a \wedge \neg(t \to^* a)\}$
  $\varphi_{\text{switch}} \equiv$ "There is a switch between $\boldsymbol{x}$ and $\boldsymbol{y}$"

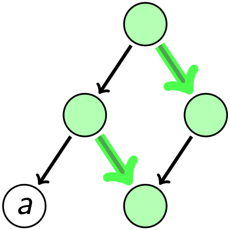
Search for Switches with Polyadic μ-calculus



$$\varphi_{\text{switch}}(\boldsymbol{a}) = \overbrace{\left(\mu W.(\boldsymbol{x} = \boldsymbol{a}) \vee (\Diamond_{\boldsymbol{x}} W)\right)}^{\varphi_{\text{reach}}} \wedge$$
$$\Diamond_{\boldsymbol{x}}\{\boldsymbol{x} \leftarrow \boldsymbol{y}\}\underbrace{\left(\nu Z.\neg(\boldsymbol{y} = \boldsymbol{a}) \wedge (\Box_{\boldsymbol{y}} Z)\right)}_{\varphi_{\text{noreach}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s;t) \mid s \to^* a\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{noreach}}]\!] = \{(s;t) \mid \neg(t \to^* a)\}$
  $\varphi_{\text{noreach}} \equiv$ "There exists no path from $\boldsymbol{y}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{switch}}]\!] = \{(s;t) \mid s \to t \wedge s \to^* a \wedge \neg(t \to^* a)\}$
  $\varphi_{\text{switch}} \equiv$ "There is a switch between $\boldsymbol{x}$ and $\boldsymbol{y}$"

Search for Switches with Polyadic μ-calculus



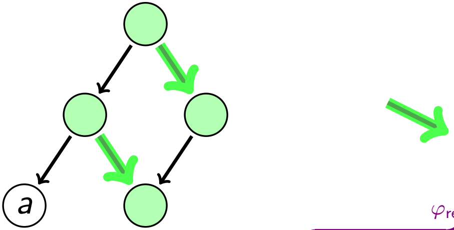
= switch regarding a

$$\varphi_{\text{switch}}(\boldsymbol{a}) = \overbrace{\Big(\mu W.(\boldsymbol{x} = \boldsymbol{a}) \vee (\Diamond_{\boldsymbol{x}} W)\Big)}^{\varphi_{\text{reach}}} \wedge$$

$$\Diamond_{\boldsymbol{x}}\{\boldsymbol{x} \leftarrow \boldsymbol{y}\}\underbrace{\Big(\nu Z.\neg(\boldsymbol{y} = \boldsymbol{a}) \wedge (\Box_{\boldsymbol{y}} Z)\Big)}_{\varphi_{\text{noreach}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s;t) \mid s \rightarrow^* a\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{noreach}}]\!] = \{(s;t) \mid \neg(t \rightarrow^* a)\}$
  $\varphi_{\text{noreach}} \equiv$ "There exists no path from $\boldsymbol{y}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{switch}}]\!] = \{(s;t) \mid s \rightarrow t \wedge s \rightarrow^* a \wedge \neg(t \rightarrow^* a)\}$
  $\varphi_{\text{switch}} \equiv$ "There is a switch between $\boldsymbol{x}$ and $\boldsymbol{y}$"

Search for Switches with Polyadic μ-calculus



= switch regarding a

$$\varphi_{\text{switch}}(\boldsymbol{a}) = \overbrace{\left(\mu W.(\boldsymbol{x} = \boldsymbol{a}) \vee (\Diamond_{\boldsymbol{x}} W)\right)}^{\varphi_{\text{reach}}} \wedge$$
$$\Diamond_{\boldsymbol{x}}\{\boldsymbol{x} \leftarrow \boldsymbol{y}\} \underbrace{\left(\nu Z.\neg(\boldsymbol{y} = \boldsymbol{a}) \wedge (\Box_{\boldsymbol{y}} Z)\right)}_{\varphi_{\text{noreach}}}$$

- $[\![\varphi_{\text{reach}}]\!] = \{(s; t) \mid s \rightarrow^* a\}$
  $\varphi_{\text{reach}} \equiv$ "There exists a path from $\boldsymbol{x}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{noreach}}]\!] = \{(s; t) \mid \neg(t \rightarrow^* a)\}$
  $\varphi_{\text{noreach}} \equiv$ "There exists no path from $\boldsymbol{y}$ to $\boldsymbol{a}$"
- $[\![\varphi_{\text{switch}}]\!] = \{(s; t) \mid s \rightarrow t \wedge s \rightarrow^* a \wedge \neg(t \rightarrow^* a)\}$
  $\varphi_{\text{switch}} \equiv$ "There is a switch between $\boldsymbol{x}$ and $\boldsymbol{y}$"

Bisimulation with Polyadic µ-calculus

Generic **bisimulation** between two models:

$$\varphi_{\text{bisim}} = \nu X.(\bigwedge_{p \in P} p_1 \Leftrightarrow p_2) \wedge (\Box_1 \Diamond_2 X \wedge \Box_2 \Diamond_1 X)$$

Bisimulation only on two sets of **observable components** O and O':

$$\varphi_{\text{bisim-obs}} = \nu X.(\bigwedge_{p \in P} \bigwedge_{(i;j) \in C} p_i \Leftrightarrow p_j) \wedge (\Box^*_{\overline{O}} \Box_O \Diamond^*_{\overline{O'}} \Diamond_{O'} X)$$

Summary & Conclusion

- Discrete modeling = coherent abstraction of real biochemical phenomena
  - → Discrete Networks / Thomas modeling
  - → Asynchronous Automata Networks
  - → Other extensions of the Process Hitting
- Static analysis based on the structure
  - → Results on attractors (multiple stable states / complex attractors)
  - → But results are not always fine enough
- Static analysis by abstract interpretation
  - → Reachability properties
  - → Very efficient (polynomial complexity)
  - → Broad rand of models (+ translations)
  - → But only one kind of property (CTL operator *EF*)
- μ-calculus
  - → More generic than CTL*
  - → Example: enumeration of attractors
  - → More ongoing work: cycles, switches...
  - → Ongoing implementation...

Summary & Conclusion

- Discrete modeling = coherent abstraction of real biochemical phenomena
  - → Discrete Networks / Thomas modeling
  - → Asynchronous Automata Networks
  - → Other extensions of the Process Hitting
- Static analysis based on the structure
  - → Results on attractors (multiple stable states / complex attractors)
  - → But results are not always fine enough
- Static analysis by abstract interpretation
  - → Reachability properties
  - → Very efficient (polynomial complexity)
  - → Broad rand of models (+ translations)
  - → But only one kind of property (CTL operator *EF*)
- µ-calculus
  - → More generic than CTL*
  - → Example: enumeration of attractors
  - → More ongoing work: cycles, switches...
  - → Ongoing implementation...

Summary & Conclusion

- Discrete modeling = coherent abstraction of real biochemical phenomena
  - → Discrete Networks / Thomas modeling
  - → Asynchronous Automata Networks
  - → Other extensions of the Process Hitting
- Static analysis based on the structure
  - → Results on attractors (multiple stable states / complex attractors)
  - → But results are not always fine enough
- Static analysis by abstract interpretation
  - → Reachability properties
  - → Very efficient (polynomial complexity)
  - → Broad rand of models (+ translations)
  - → But only one kind of property (CTL operator *EF*)
- μ-calculus
  - → More generic than CTL*
  - → Example: enumeration of attractors
  - → More ongoing work: cycles, switches...
  - → Ongoing implementation...

Summary & Conclusion

- Discrete modeling = coherent abstraction of real biochemical phenomena
  - → Discrete Networks / Thomas modeling
  - → Asynchronous Automata Networks
  - → Other extensions of the Process Hitting
- Static analysis based on the structure
  - → Results on attractors (multiple stable states / complex attractors)
  - → But results are not always fine enough
- Static analysis by abstract interpretation
  - → Reachability properties
  - → Very efficient (polynomial complexity)
  - → Broad rand of models (+ translations)
  - → But only one kind of property (CTL operator *EF*)
- μ-calculus
  - → More generic than CTL*
  - → Example: enumeration of attractors
  - → More ongoing work: cycles, switches...
  - → Ongoing implementation...

Bibliography

- René Thomas. On the Relation Between the Logical Structure of Systems and Their Ability to Generate Multiple Steady States or Sustained Oscillations. In Jean Della Dora, Jacques Demongeot and Bernard Lacolle, editors: *Numerical Methods in the Study of Critical Phenomena*, Synergies 9, 180–193. Springer Berlin Heidelberg, 1981.
- 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.
- Loïc Paulevé, Adrien Richard. Static Analysis of Boolean Networks Based on Interaction Graphs: A Survey, *Electronic Notes in Theoretical Computer Science* 284, 93–104. Elsevier, 2012.
- Adrien Richard and Jean-Paul Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics* 155(18), 2403–2413. 2007.
- Adrien Richard. Negative circuits and sustained oscillations in asynchronous automata networks, *Advances in Applied Mathematics* 44(4), 378–392. Elsevier, 2010.
- Élisabeth Remy, Paul Ruet and Denis Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework, *Advances in Applied Mathematics* 41(3), 335–350. Elsevier, 2008.
- Mmaxime Folschette, Loïc Paulevé, Kastumi Inoue, Morgan Magnin and Olivier Roux. Identification of Biological Regulatory Networks from Process Hitting models, *Theoretical Computer Science* 568, 49–71. Elsevier, 2015a.
- Maxime Folschette, Loïc Paulevé, Morgan Magnin and Olivier Roux. Sufficient Conditions for Reachability in Automata Networks with Priorities, *Theoretical Computer Science*. Elsevier, 2015b, in press.

Thank you