

Journées annuelles 2018 du GT Bioss

GULA: Semantics-Free Learning of a Biological Regulatory Networks from a Synchronous, Asynchronous or Generalized State Graph

Maxime FOLSCHETTE

Dyliss team IRISA Univ Rennes, Inria, CNRS, IRISA
Dymec team IRSET Univ Rennes 1, Inserm

`maxime.folschette@irisa.fr`
`http://maxime.folschette.name/`

2018-07-02

Introduction

Learn interaction rules from the dynamical transitions

- LFIT: synchronous, deterministic (Boolean)
[Inoue, Ribeiro, Sakama, *Machine Learning Jour.*, 2014]
- LFKT: synchronous, with memory (Boolean)
[Ribeiro, Magnin, Inoue, Sakama, *Frontiers in Bioeng. and Biotech.*, 2015]
- LUST: synchronous, non-deterministic
[Martinez, Ribeiro, Inoue, Alenya, Torras, *ICLP*, 2015.]
- ACEDIA: synchronous, continuous domains
[Ribeiro, Tourret, Folschette, +5, *ILP*, 2017]

...But all are semantics-specific

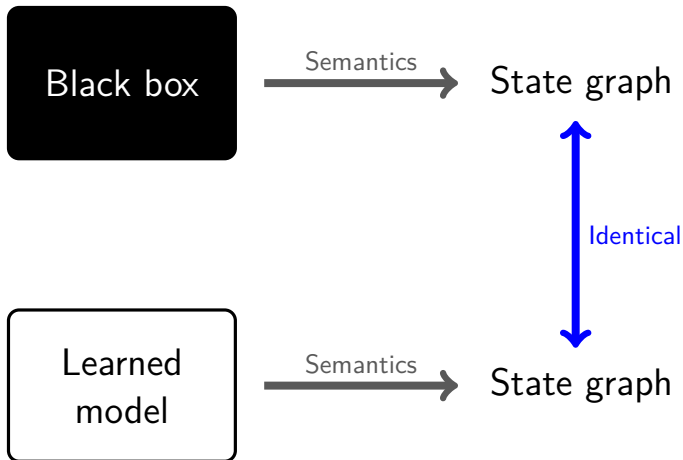
What if the semantics is unknown?

- Learn general rules without foreseeing the semantics
- Properties to characterize the semantics

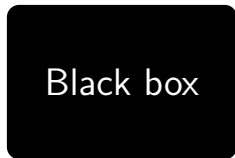
Introduction



Introduction



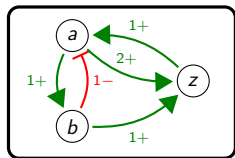
Introduction



State graph



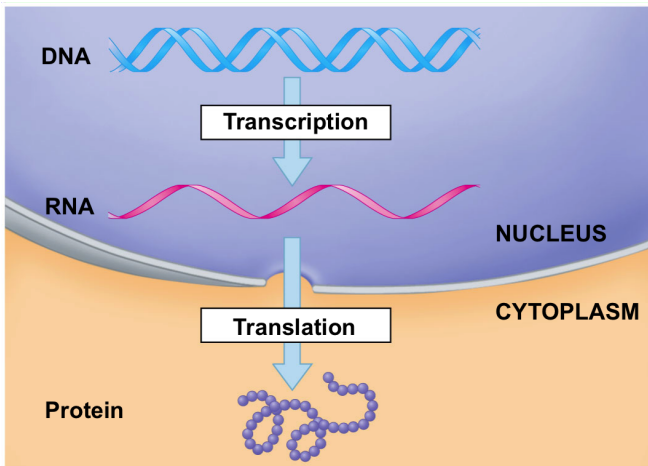
Identical



State graph

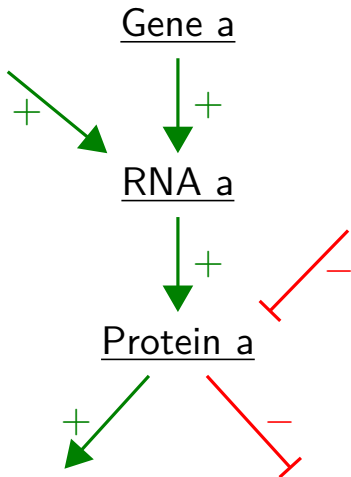
Discrete Networks

Preliminary Abstraction

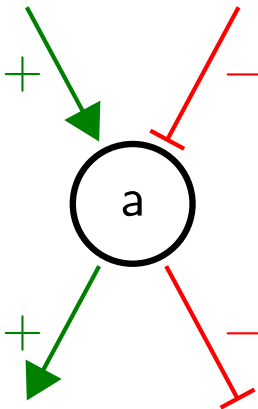


© 2012 Pearson Education, Inc.

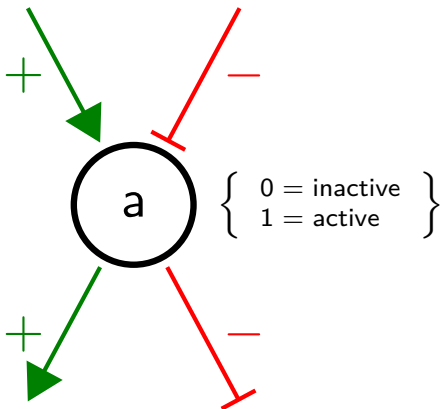
Preliminary Abstraction



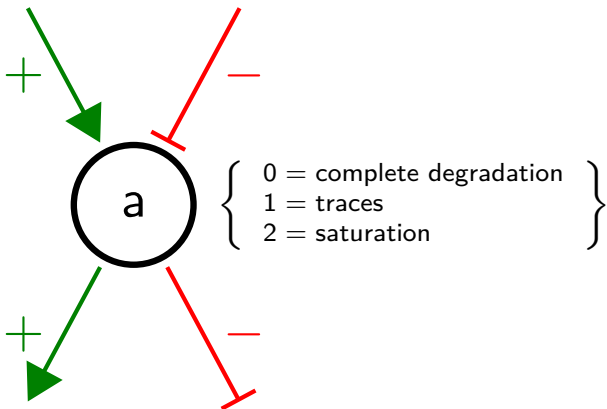
Preliminary Abstraction



Preliminary Abstraction



Preliminary Abstraction



Discrete Networks / Thomas Modeling

[Kauffman, *Journal of Theoretical Biology*, 1969]

[Thomas, *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A discrete domain for each component $\text{dom}(a) = \llbracket 0; 2 \rrbracket$
- Discrete parameters / evolution functions $f^a : \mathcal{S} \rightarrow \text{dom}(a)$
- Signs & thresholds on the edges (redundant) $a \xrightarrow{2+} z$

	a	f^b	z	b	f^a	a	b	f^z
a	0	0	0	0	1	0	0	0
	1	1	0	1	0	0	1	0
	2	1	1	0	1	1	0	0
			1	1	2	1	1	0
						2	0	0
b						2	1	1

Semantics = From this information, what are the next possible state(s)?

Discrete Networks / Thomas Modeling

[Kauffman, *Journal of Theoretical Biology*, 1969]

[Thomas, *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A discrete domain for each component $\text{dom}(a) = \llbracket 0; 2 \rrbracket$
- Discrete parameters / evolution functions $f^a : \mathcal{S} \rightarrow \text{dom}(a)$
- Signs & thresholds on the edges (redundant) $a \xrightarrow{2+} z$

$\llbracket 0; 2 \rrbracket$	a	f^b	z	b	f^a	a	b	f^z
⊙ a	0	0	0	0	1	0	0	0
	1	1	0	1	0	0	1	0
	2	1	1	0	1	1	0	0
			1	1	2	1	1	0
						2	0	0
						2	1	1

⊙
 z

$\llbracket 0; 1 \rrbracket$

⊙
 b

$\llbracket 0; 1 \rrbracket$

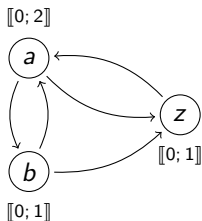
Semantics = From this information, what are the next possible state(s)?

Discrete Networks / Thomas Modeling

[Kauffman, *Journal of Theoretical Biology*, 1969]

[Thomas, *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A discrete domain for each component $\text{dom}(a) = \llbracket 0; 2 \rrbracket$
- Discrete parameters / evolution functions $f^a : \mathcal{S} \rightarrow \text{dom}(a)$
- Signs & thresholds on the edges (redundant) $a \xrightarrow{2+} z$



a	f^b	z	b	f^a	a	b	f^z
0	0	0	0	1	0	0	0
1	1	0	1	0	0	1	0
2	1	1	0	1	1	0	0
		1	1	2	1	1	0
					2	0	0
					2	1	1

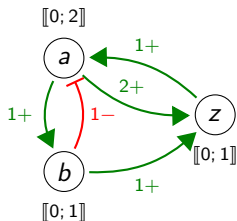
Semantics = From this information, what are the next possible state(s)?

Discrete Networks / Thomas Modeling

[Kauffman, *Journal of Theoretical Biology*, 1969]

[Thomas, *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A discrete domain for each component $\text{dom}(a) = \llbracket 0; 2 \rrbracket$
- Discrete parameters / evolution functions $f^a : \mathcal{S} \rightarrow \text{dom}(a)$
- Signs & thresholds on the edges (redundant) $a \xrightarrow{2+} z$



a	f^b	z	b	f^a	a	b	f^z
0	0	0	0	1	0	0	0
1	1	0	1	0	0	1	0
2	1	1	0	1	1	0	0
		1	1	2	1	1	0
					2	0	0
					2	1	1

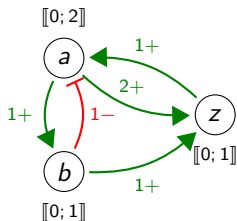
Semantics = From this information, what are the next possible state(s)?

Discrete Networks / Thomas Modeling

[Kauffman, *Journal of Theoretical Biology*, 1969]

[Thomas, *Journal of Theoretical Biology*, 1973]

- A set of components $N = \{a, b, z\}$
- A discrete domain for each component $\text{dom}(a) = \llbracket 0; 2 \rrbracket$
- Discrete parameters / evolution functions $f^a : \mathcal{S} \rightarrow \text{dom}(a)$
- Signs & thresholds on the edges (redundant) $a \xrightarrow{2+} z$



a	f^b	z	b	f^a	a	b	f^z
0	0	0	0	1	0	0	0
1	1	0	1	0	0	1	0
2	1	1	0	1	1	0	0
		1	1	2	1	1	0
					2	0	0
					2	1	1

Semantics = From this information, what are the next possible state(s)?

Semantics

10

11

00

01

Synchronous

Asynchronous

Generalized

Other semantics:

- Round-robin
- With memory
- With priorities
- ...

Semantics

10 focal point
 (11)

00 01

Synchronous

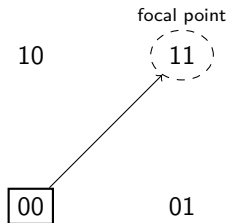
Asynchronous

Generalized

Other semantics:

- Round-robin
- With memory
- With priorities
- ...

Semantics



Synchronous

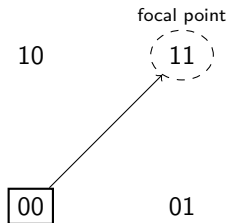
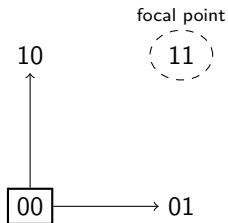
Asynchronous

Generalized

Other semantics:

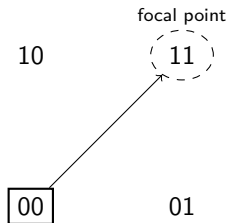
- Round-robin
- With memory
- With priorities
- ...

Semantics

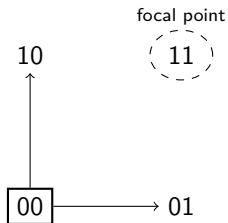
**Synchronous****Asynchronous****Generalized****Other semantics:**

- Round-robin
- With memory
- With priorities
- ...

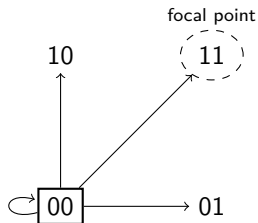
Semantics



Synchronous



Asynchronous

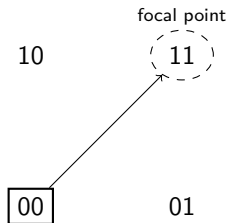
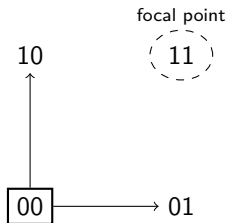
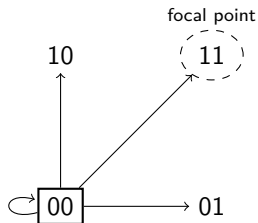


Generalized

Other semantics:

- Round-robin
- With memory
- With priorities
- ...

Semantics

**Synchronous****Asynchronous****Generalized****Other semantics:**

- Round-robin
- With memory
- With priorities
- ...

Logic Programs

Logic Rule

$$\underbrace{x_0^{val_0}(t)}_{\text{head}} \leftarrow \underbrace{x_1^{val_1}(t-1) \wedge x_2^{val_2}(t-1) \wedge \dots \wedge x_n^{val_n}(t-1)}_{\text{body}}.$$

- $x_0, x_1, x_2, \dots, x_n$: variables a, b, z
No duplicates in x_1, x_2, \dots, x_n in the *body*
- $val_0, val_1, val_2, \dots, val_n$: values $0, 1, 2, \dots$
With $val_i \in \text{dom}(x_i)$
- t : time step
Always t in the *head* and $t-1$ in the *body*
- All atoms in the *body* are in conjunction
- \leftarrow is the (reverse) implication

Logic Rule

$$\underbrace{x_0^{val_0}(t)}_{\text{head}} \leftarrow \underbrace{x_1^{val_1}(t-1) \wedge x_2^{val_2}(t-1) \wedge \dots \wedge x_n^{val_n}(t-1)}_{\text{body}}.$$

- $x_0, x_1, x_2, \dots, x_n$: variables a, b, z
No duplicates in x_1, x_2, \dots, x_n in the *body*
- $val_0, val_1, val_2, \dots, val_n$: values $0, 1, 2, \dots$
With $val_i \in \text{dom}(x_i)$
- t : time step
Always t in the *head* and $t - 1$ in the *body*
- All atoms in the *body* are in conjunction
- \leftarrow is the (reverse) implication

Logic Rule

$$\underbrace{x_0^{val_0}(t)}_{\text{head}} \leftarrow \underbrace{x_1^{val_1}(t-1) \wedge x_2^{val_2}(t-1) \wedge \dots \wedge x_n^{val_n}(t-1)}_{\text{body}}.$$

- $x_0, x_1, x_2, \dots, x_n$: variables a, b, z
No duplicates in x_1, x_2, \dots, x_n in the *body*
- $val_0, val_1, val_2, \dots, val_n$: values $0, 1, 2, \dots$
With $val_i \in \text{dom}(x_i)$
- t : time step
Always t in the *head* and $t - 1$ in the *body*
- All atoms in the *body* are in conjunction
- \leftarrow is the (reverse) implication

Logic Rule

$$\underbrace{x_0^{val_0}(t)}_{\text{head}} \leftarrow \underbrace{x_1^{val_1}(t-1) \wedge x_2^{val_2}(t-1) \wedge \dots \wedge x_n^{val_n}(t-1)}_{\text{body}}.$$

- $x_0, x_1, x_2, \dots, x_n$: variables a, b, z
No duplicates in x_1, x_2, \dots, x_n in the *body*
- $val_0, val_1, val_2, \dots, val_n$: values $0, 1, 2, \dots$
With $val_i \in \text{dom}(x_i)$
- t : time step
Always t in the *head* and $t-1$ in the *body*
- All atoms in the *body* are in conjunction
- \leftarrow is the (reverse) implication

Logic Rule

$$\underbrace{x_0^{val_0}(t)}_{head} \leftarrow \underbrace{x_1^{val_1}(t-1) \wedge x_2^{val_2}(t-1) \wedge \dots \wedge x_n^{val_n}(t-1)}_{body}.$$

- $x_0, x_1, x_2, \dots, x_n$: variables a, b, z
No duplicates in x_1, x_2, \dots, x_n in the *body*
- $val_0, val_1, val_2, \dots, val_n$: values $0, 1, 2, \dots$
With $val_i \in \text{dom}(x_i)$
- t : time step
Always t in the *head* and $t-1$ in the *body*
- All atoms in the *body* are in conjunction
- \leftarrow is the (reverse) implication

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}(t)}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}(t-1), x_2^{val_2}(t-1), \dots, x_n^{val_n}(t-1)\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{all match } \{a^2, b^0, c^1\}$$

A rule *R* **matches** a state *s* iff *body* $\subseteq s$

→ For all state *s*, if the rule **matches** *s* then there exists a state *s'* so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{all match } (a^2, b^0, c^1)$$

A rule *R* **matches** a state *s* iff *body* $\subseteq s$

→ For all state *s*, if the rule **matches** *s* then there exists a state *s'* so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{all match } (a^2, b^0, c^1)$$

A rule R **matches** a state s iff $\text{body} \subseteq s$ → For all state s , if the rule **matches** s then there exists a state s' so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$ **Semantics:** same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{all match } \langle a^2, b^0, c^1 \rangle$$

A rule R **matches** a state s iff $\text{body} \subseteq s$

→ For all state s , if the rule **matches** s then there exists a state s' so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{ all match } \langle a^2, b^0, c^1 \rangle$$

A rule *R* **matches** a state *s* iff *body* $\subseteq s$

→ For all state *s*, if the rule **matches** *s* then there exists a state *s'* so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{ all match } \langle a^2, b^0, c^1 \rangle$$

A rule *R* **matches** a state *s* iff *body* $\subseteq s$

→ For all state *s*, if the rule **matches** *s* then there exists a state *s'* so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* is a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{ all match } \langle a^2, b^0, c^1 \rangle$$

A rule *R* **matches** a state *s* iff *body* $\subseteq s$

→ For all state *s*, if the rule **matches** *s* then there exists a state *s'* so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Logic Rule

Set interpretation:

$$\underbrace{x_0^{val_0}}_{\text{head}} \leftarrow \underbrace{\{x_1^{val_1}, x_2^{val_2}, \dots, x_n^{val_n}\}}_{\text{body}}.$$

→ When *body* is true, *head* **may be** a potential outcome

$$\text{Examples: } \left. \begin{array}{l} a^1 \leftarrow \{a^2, b^0, c^1\}. \\ b^1 \leftarrow \{c^1\}. \\ c^0 \leftarrow \emptyset. \end{array} \right\} \text{ all match } \langle a^2, b^0, c^1 \rangle$$

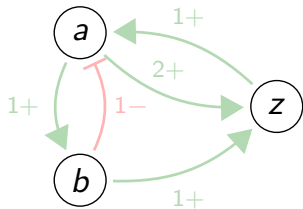
A rule *R* **matches** a state *s* iff *body* $\subseteq s$

→ For all state *s*, if the rule **matches** *s* then there **should exist** a state *s'* so that $s \rightarrow s'$ and $x_0^{val_0} \in s'$

Semantics: same as for discrete networks

Model as a Logic Program

Discrete model:



+ Table of parameters
or logic gates

Logic program:

$$b(1, t) \leftarrow a(1, t - 1).$$

$$b(1, t) \leftarrow a(2, t - 1).$$

$$b(0, t) \leftarrow a(0, t - 1).$$

$$z(1, t) \leftarrow a(2, t - 1) \wedge b(1, t - 1).$$

$$z(0, t) \leftarrow a(0, t - 1).$$

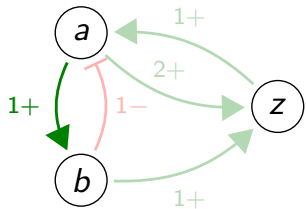
$$z(0, t) \leftarrow a(1, t - 1).$$

$$z(0, t) \leftarrow b(0, t - 1).$$

etc...

Model as a Logic Program

Discrete model:



+ Table of parameters
or logic gates

Logic program:

$$b(1, t) \leftarrow a(1, t - 1).$$

$$b(1, t) \leftarrow a(2, t - 1).$$

$$b(0, t) \leftarrow a(0, t - 1).$$

$$z(1, t) \leftarrow a(2, t - 1) \wedge b(1, t - 1).$$

$$z(0, t) \leftarrow a(0, t - 1).$$

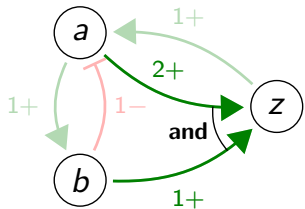
$$z(0, t) \leftarrow a(1, t - 1).$$

$$z(0, t) \leftarrow b(0, t - 1).$$

etc...

Model as a Logic Program

Discrete model:



+ Table of parameters
or logic gates

Logic program:

$$b(1, t) \leftarrow a(1, t - 1).$$

$$b(1, t) \leftarrow a(2, t - 1).$$

$$b(0, t) \leftarrow a(0, t - 1).$$

$$z(1, t) \leftarrow a(2, t - 1) \wedge b(1, t - 1).$$

$$z(0, t) \leftarrow a(0, t - 1).$$

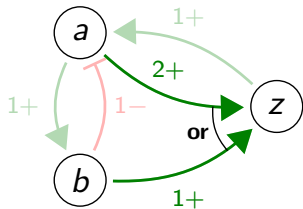
$$z(0, t) \leftarrow a(1, t - 1).$$

$$z(0, t) \leftarrow b(0, t - 1).$$

etc...

Model as a Logic Program

Discrete model:



+ Table of parameters
or logic gates

Logic program:

$$b(1, t) \leftarrow a(1, t - 1).$$

$$b(1, t) \leftarrow a(2, t - 1).$$

$$b(0, t) \leftarrow a(0, t - 1).$$

$$z(1, t) \leftarrow a(2, t - 1).$$

$$z(1, t) \leftarrow b(1, t - 1).$$

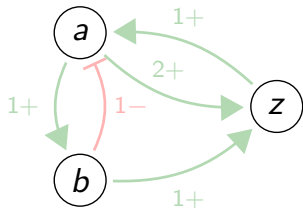
$$z(0, t) \leftarrow a(1, t - 1) \wedge b(0, t - 1).$$

$$z(0, t) \leftarrow a(0, t - 1) \wedge b(0, t - 1).$$

etc...

Model as a Logic Program

Discrete model:



+ Table of parameters
or logic gates

Logic program:

$$b(1, t) \leftarrow a(1, t - 1).$$

$$b(1, t) \leftarrow a(2, t - 1).$$

$$b(0, t) \leftarrow a(0, t - 1).$$

$$z(1, t) \leftarrow a(2, t - 1).$$

$$z(1, t) \leftarrow b(1, t - 1).$$

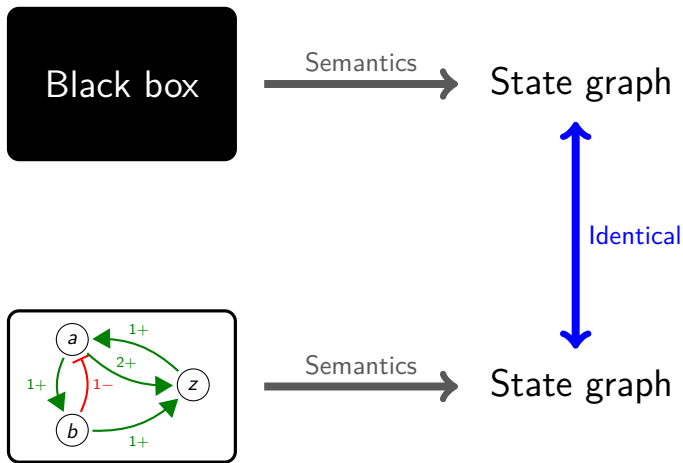
$$z(0, t) \leftarrow a(1, t - 1) \wedge b(0, t - 1).$$

$$z(0, t) \leftarrow a(0, t - 1) \wedge b(0, t - 1).$$

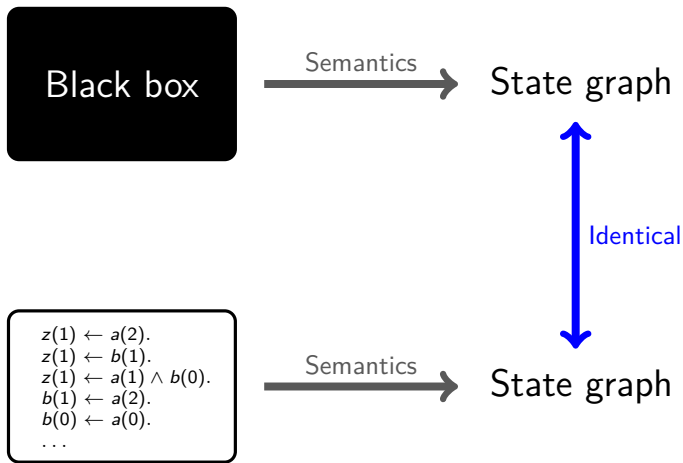
etc...

Learning

Principle of the Learning



Principle of the Learning



GULA = General Usage LFIT Algorithm

Input: a set of transitions (elements of $\mathcal{S} \times \mathcal{S}$)

Output: a program that respects:

- **Consistency:** The program allows no negative examples
- **Realization:** The program covers all positive examples
- **Completeness:** The program covers all the state space
- **Minimality** of the rules (most general bodies)

Algorithm

- Start from the most general program:

$$P := \{x^{val} \leftarrow \emptyset. \mid x \in V \wedge val \in \text{dom}(x)\}$$

- For each state s :
 - consider all transitions that start from s
 - make minimal revisions on the program with these transitions
- Remove all dominated rules (that are not the most general)

where R dominates R' iff $\text{head}(R) = \text{head}(R')$ and $\text{body}(R) \subseteq \text{body}(R')$

This learning is independent from the semantics! (but...)

Formally proved: Compatible with transitions generated in **synchronous**, **asynchronous** and **generalized** semantics

Expectation: Compatible with a wider class of “learnable” semantics

Algorithm

- Start from the most general program:

$$P := \{x^{val} \leftarrow \emptyset. \mid x \in V \wedge val \in \text{dom}(x)\}$$

- For each state s :
 - consider all transitions that start from s
 - make minimal revisions on the program with these transitions
- Remove all dominated rules (that are not the most general)

where R dominates R' iff $\text{head}(R) = \text{head}(R')$ and $\text{body}(R) \subseteq \text{body}(R')$

This learning is independent from the semantics! (but...)

Formally proved: Compatible with transitions generated in **synchronous**, **asynchronous** and **generalized** semantics

Expectation: Compatible with a wider class of “learnable” semantics

Algorithm

- Start from the most general program:

$$P := \{x^{val} \leftarrow \emptyset. \mid x \in V \wedge val \in \text{dom}(x)\}$$

- For each state s :
 - consider all transitions that start from s
 - make minimal revisions on the program with these transitions
- Remove all dominated rules (that are not the most general)

where R dominates R' iff $\text{head}(R) = \text{head}(R')$ and $\text{body}(R) \subseteq \text{body}(R')$

This learning is independent from the semantics! (**but...**)

Formally proved: Compatible with transitions generated in **synchronous**, **asynchronous** and **generalized** semantics

Expectation: Compatible with a wider class of “learnable” semantics

Semantics

Definition of a Semantics

Formally, a semantics is an element of:

$$\{\text{Set of complete programs}\} \rightarrow (\mathcal{S} \rightarrow \wp(\mathcal{S}) \setminus \emptyset)$$

Complete program = for all state s and all variable x , there exists a rule that matches s and whose head is about x

That is, a function that, to each complete program, associates a set of transitions so that each state has at least one successor.

- Non-determinism is the general case
Because each state can have several successors
- Stable rules (not changing the state) can be included or not
Depends on the chosen semantics
- No dead-ends, only fixed points as self-loops
Because the state graph must also be complete

Examples of Semantics

Synchronous semantics

$$P \mapsto (s \mapsto \{\{ \mathit{head}(R) \mid R \in P \wedge R \text{ matches } s \} \subseteq \mathcal{S}\})$$

Update all variables at each step using *heads* of on matching rules

Asynchronous semantics

$$P \mapsto (s \mapsto \{s \parallel \mathit{head}(R) \mid R \in P \wedge R \text{ matches } s \wedge \mathit{head}(R) \notin s\} \\ \cup \{s \mid \text{if } s \rightarrow s \text{ is the only outcome}\})$$

where: $s \parallel x^{val} := s$ in which the value of x is replaced by val

Update one variable at each step, avoid self-loops if possible

- + **Generalized semantics**
- + **Characterizations** of each semantics

Examples of Semantics

Synchronous semantics

$$P \mapsto (s \mapsto \{\{ \mathit{head}(R) \mid R \in P \wedge R \text{ matches } s \} \subseteq \mathcal{S}\})$$

Update all variables at each step using *heads* of on matching rules

Asynchronous semantics

$$P \mapsto (s \mapsto \{s \parallel \mathit{head}(R) \mid R \in P \wedge R \text{ matches } s \wedge \mathit{head}(R) \notin s\} \\ \cup \{s \mid \text{if } s \rightarrow s \text{ is the only outcome}\})$$

where: $s \parallel x^{val} := s$ in which the value of x is replaced by val

Update one variable at each step, avoid self-loops if possible

- + Generalized semantics
- + Characterizations of each semantics

Examples of Semantics

Synchronous semantics

$$P \mapsto (s \mapsto \{\{ \mathit{head}(R) \mid R \in P \wedge R \text{ matches } s \} \subseteq \mathcal{S}\})$$

Update all variables at each step using *heads* of on matching rules

Asynchronous semantics

$$P \mapsto (s \mapsto \{s \parallel \mathit{head}(R) \mid R \in P \wedge R \text{ matches } s \wedge \mathit{head}(R) \notin s\} \\ \cup \{s \mid \text{if } s \rightarrow s \text{ is the only outcome}\})$$

where: $s \parallel x^{val} := s$ in which the value of x is replaced by val

Update one variable at each step, avoid self-loops if possible

- + **Generalized semantics**
- + **Characterizations** of each semantics

Limits of the Semantics

Limit of this definition:

$$P \mapsto (s \mapsto \{\langle 00\dots 0 \rangle\})$$

is a valid semantics, that doesn't allow to learn the interactions

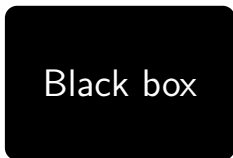
The learned program will always be: $\{x^0 \leftarrow \emptyset. \mid x \in V\}$ whatever the original program P used to generate the dynamics

⇒ An “interesting” (learnable) semantics takes into account the *heads* of the rules in P

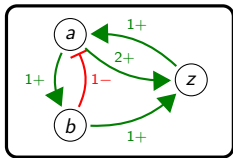
Other limitations?

Learning Time Series

Potential Usage

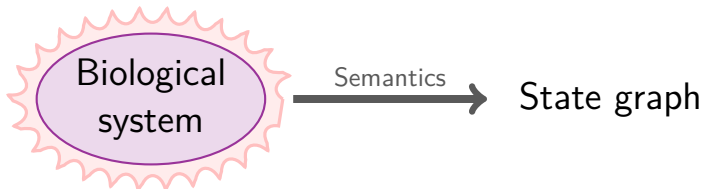


State graph

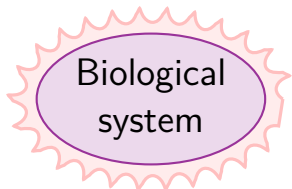


State graph

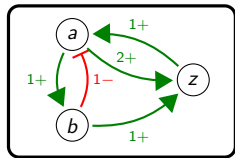
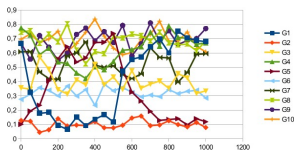
Potential Usage



Potential Usage



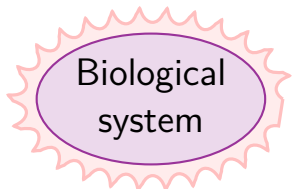
Behavior →



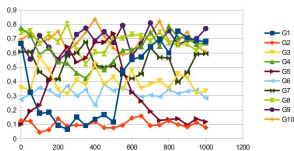
Semantics →

State graph

Potential Usage

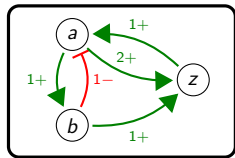


Behavior →



Discretization

State graph



Semantics →

Potential Usage

Towards the automation of learning time series

Challenges:

- Discretization → ACEDIA
A different discretization gives a different result
- Partial data → LUST
Predict parts of the system
- Noise
Try not to over-learn
- Changing or heterogeneous “semantics”
Over time or between components

Potential Usage

Towards the automation of learning time series

Challenges:

- Discretization → ACEDIA
A different discretization gives a different result
- Partial data → LUST
Predict parts of the system
- Noise
Try not to over-learn
- Changing or heterogeneous “semantics”
Over time or between components

Conclusion

Logic rules \Leftrightarrow networks interactions \Leftrightarrow automata transitions

Learning of the structure of a model 1-step learning algorithm by successive refinements

Independent of the semantics

Proved for synchronous, asynchronous, generalized semantics

Outlooks

- Definition of what can be learned / what is interesting in a semantics
- Automatic learning of time series data (noise, discretization, ...)
- Learning the semantics alongside the structure
- Optimizations (parallelization, approximations)

References

- Stuart A. Kauffman. [Metabolic stability and epigenesis in randomly constructed genetic nets](#). *Journal of Theoretical Biology*, volume 22, n. 3, pages 437–467, 1969.
- René Thomas. [Boolean formalization of genetic control circuits](#). *Journal of Theoretical Biology*, volume 42, n. 3, pages 563–85, 1973.
- Katsumi Inoue, Tony Ribeiro, and Chiaki Sakama. [Learning from interpretation transition](#). *Machine Learning Journal*, volume 94, issue 1, pages 51–79, 2014.
- Tony Ribeiro, Morgan Magnin, Katsumi Inoue, Chiaki Sakama. [Learning delayed influences of biological systems](#). *Frontiers in Bioengineering and Biotechnology*, volume 2, issue 81, 2015.
- David Martinez, Tony Ribeiro, Katsumi Inoue, Guillem Alenya, Carme Torras. [Learning probabilistic action models from interpretation transitions](#). *The 31st International Conference on Logic Programming (ICLP)*, Cork, Ireland, 2015.
- Tony Ribeiro, Sophie Touret, Maxime Folschette, Morgan Magnin, Domenico Borzacchiello, Francisco Chinesta, Olivier Roux, Katsumi Inoue. [Inductive Learning from State Transitions over Continuous Domains](#). *The 27th International Conference on Inductive Logic Programming (ILP)*, Orléans, France, 2017.

Learning Process

The algorithm successively takes into account groups of transitions and performs **minimal modifications** on the program learned so far

Let R a rule in conflict with the current transitions, that is: there exists a state s so that R **matches** s , but $\forall s' \in \mathcal{S}$ so that $s \rightarrow s'$, $head(R) \notin s'$
That is: R expresses a potential outcome for a variable which never happens in s

Least specialization of R by s :

$$R := head \leftarrow body$$

$$L_{spe}(R, s) := \{ head \leftarrow body \cup \{x^{val}\} \mid x^{val} \notin s \wedge \forall val' \in \mathbb{N}, x^{val'} \notin body \}$$

Least revision of P by a set of transitions T :

$$L_{rev}(P, T) := (P \setminus R_P) \cup \bigcup_{R \in R_P} L_{spe}(R, s)$$