Challenges with the logical modeling of biological systems: learning and analyzing discrete and hybrid models

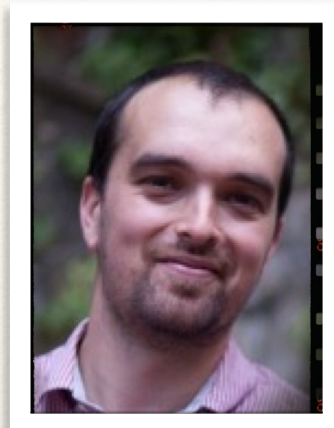
Morgan MAGNIN - http://www.morganmagnin.net - @morgan_it

École centrale de Nantes - LS2N - France
Team MeForBio

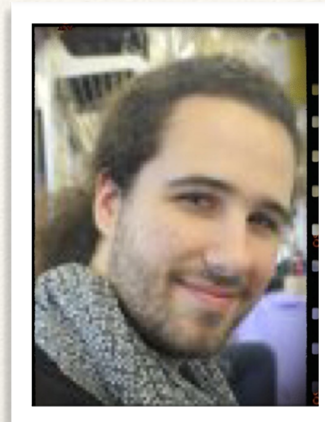
National Institute of Informatics - Japan
Inoue Lab.

Joint work with:

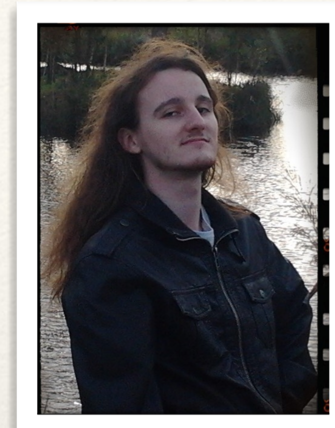
Olivier ROUX



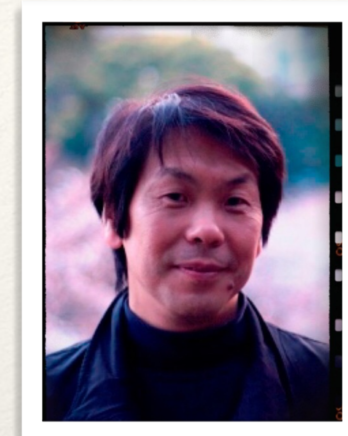
Loïc PAULEVÉ



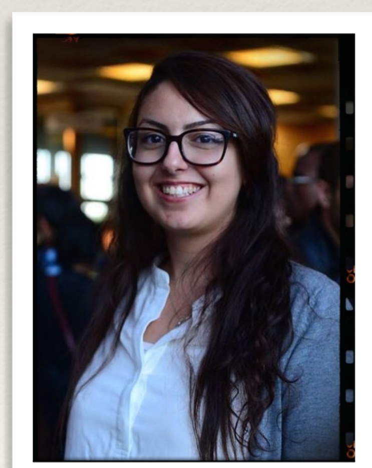
Maxime FOLSCHETTE



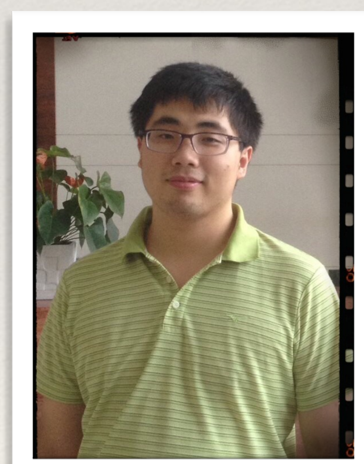
Tony RIBEIRO



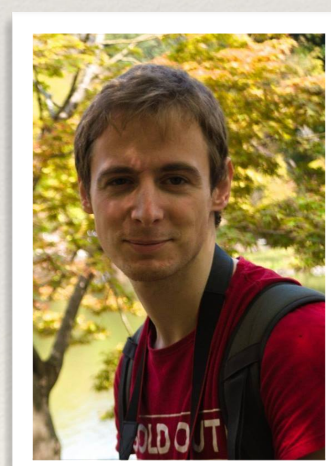
Katsumi INOUE
国立情報学研究所,東京都,日本



Emna BEN ABDALLAH



Xinwei CHAI

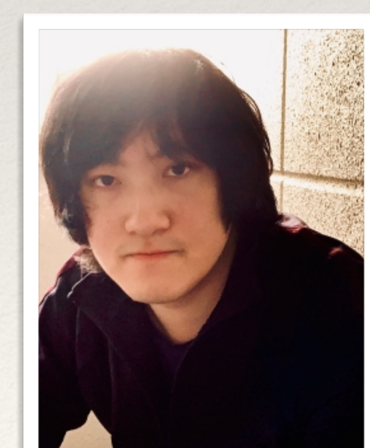


Samuel BUCHET

Honglu SUN

LS2N, Nantes

National Institute of Informatics,
Tôkyô, Japon

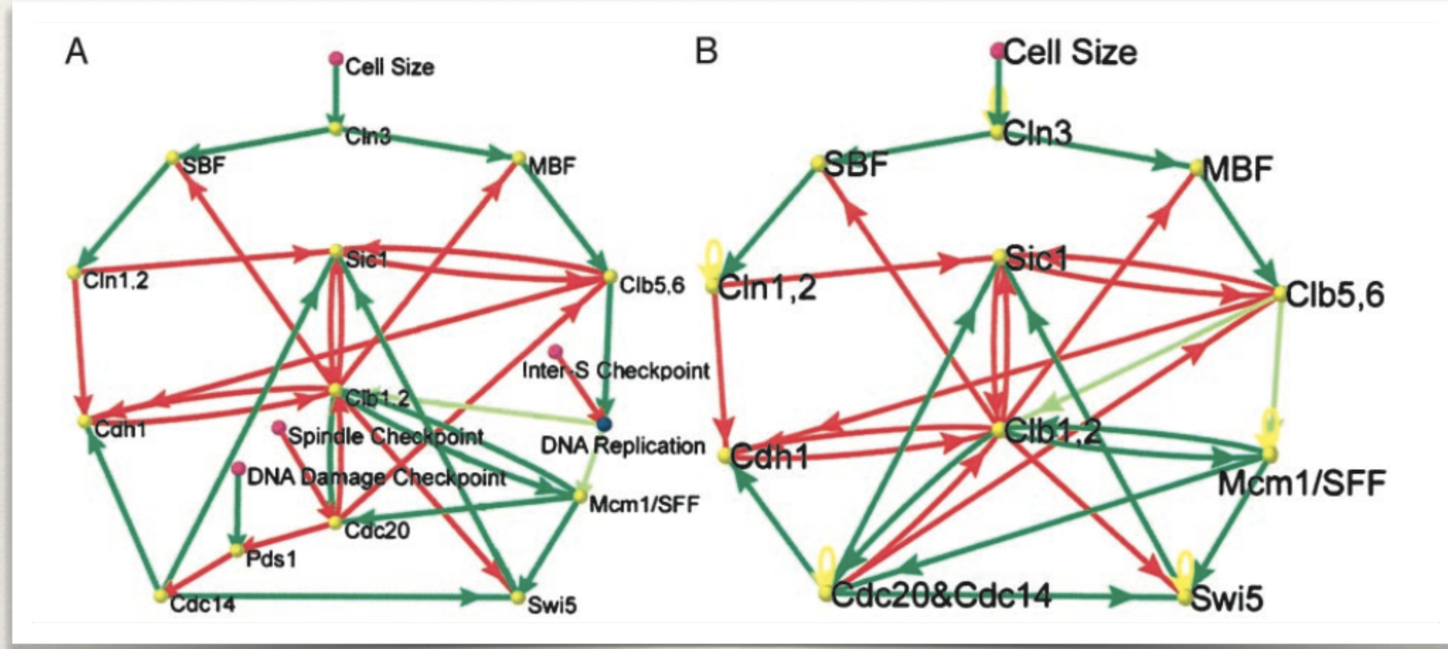


Mitsuhiro ODAKA

Motivations

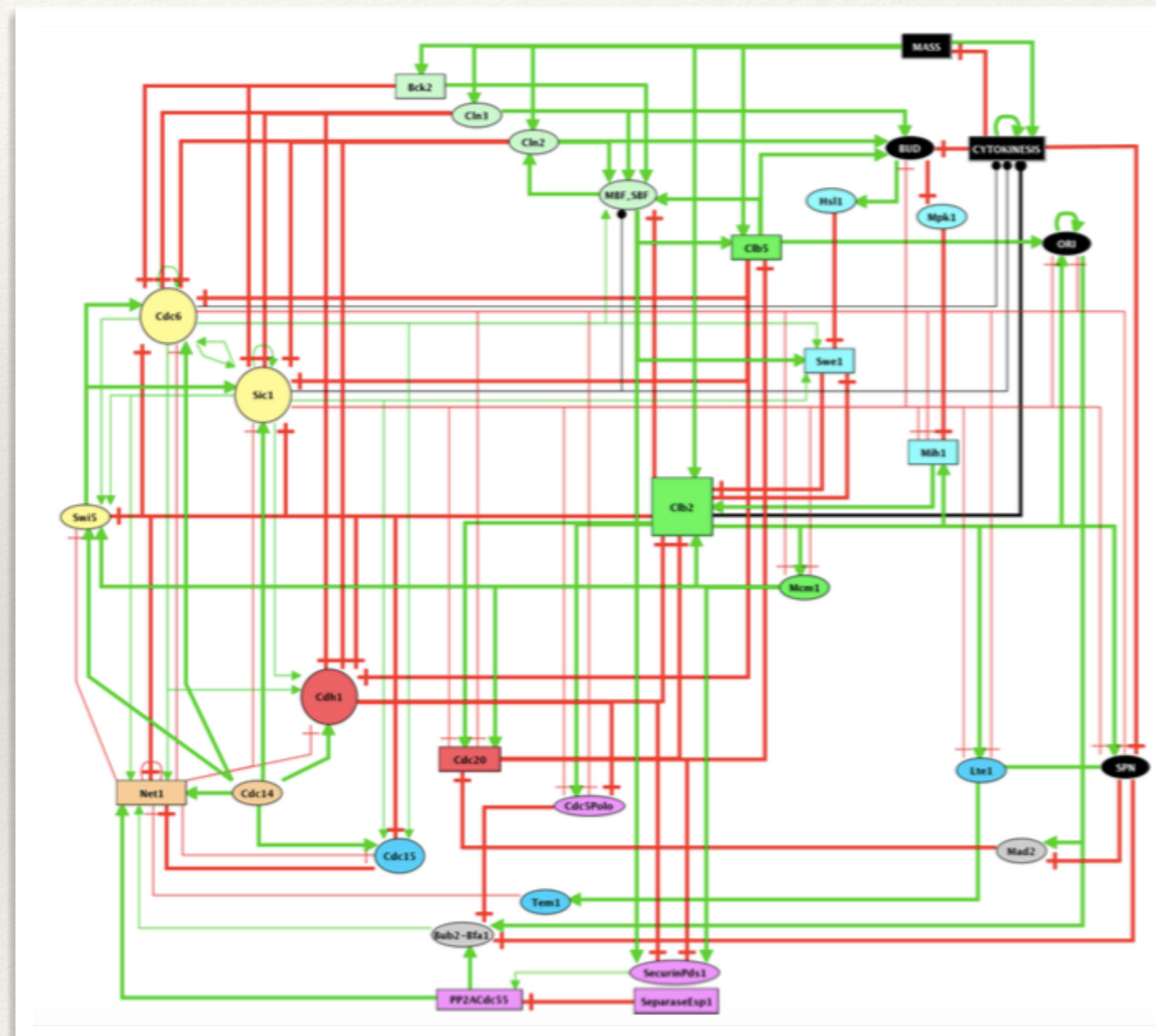
- Goals:
    - Help to the **diagnosis** and **prevention** of diseases
    - Develop new **therapies**
- **Systems biology**
    - Consider a living organism as a system of **interacting networks**
    - Link between physiological functions / **dynamics** of the regulations

Models: why?



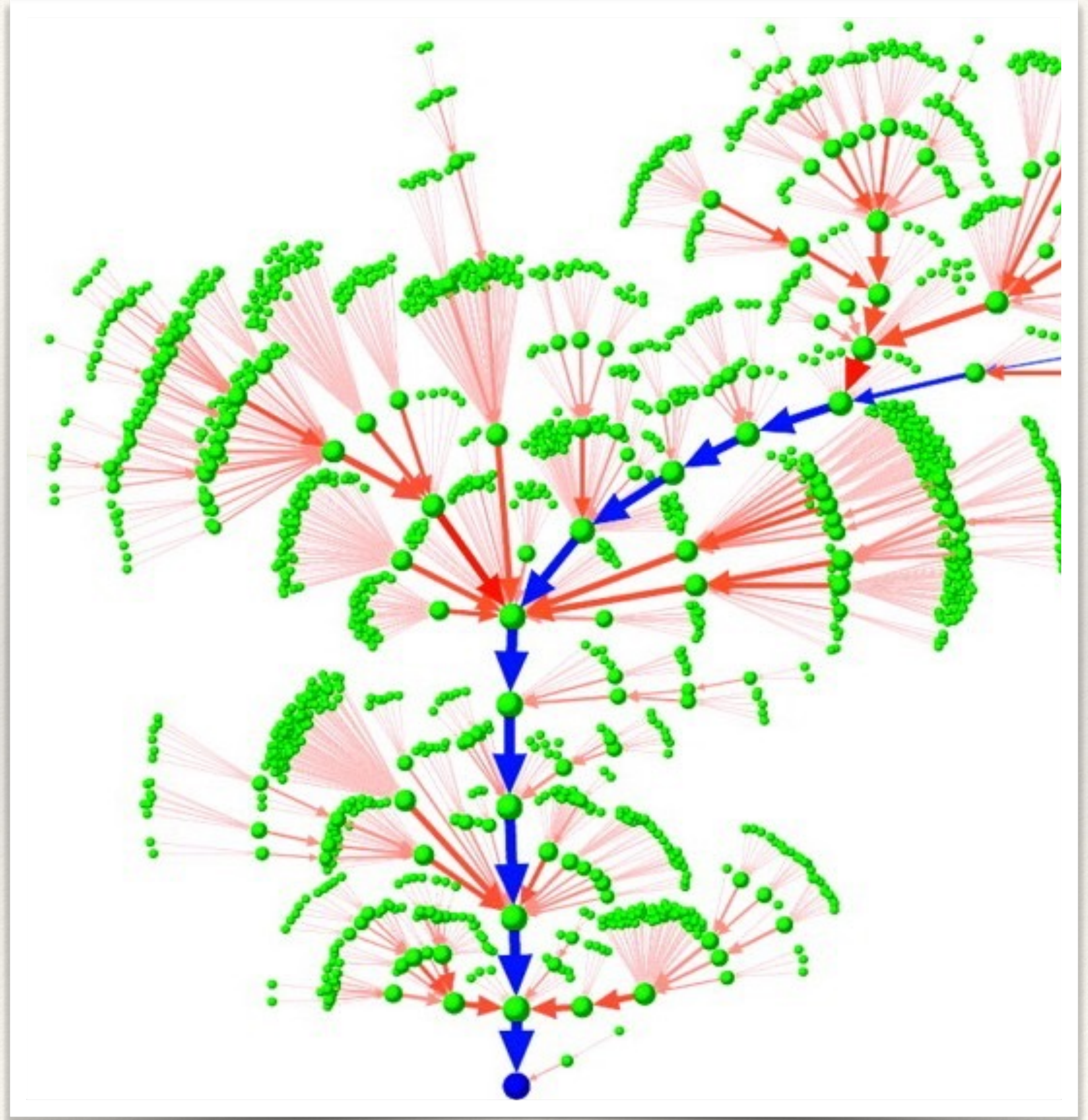
(A) A first model, and (B) a simplified model of the cellular cycle of the budding yeast [LLL+04]

Models: why?



Logical regulatory graph of the budding yeast [FNL+09]

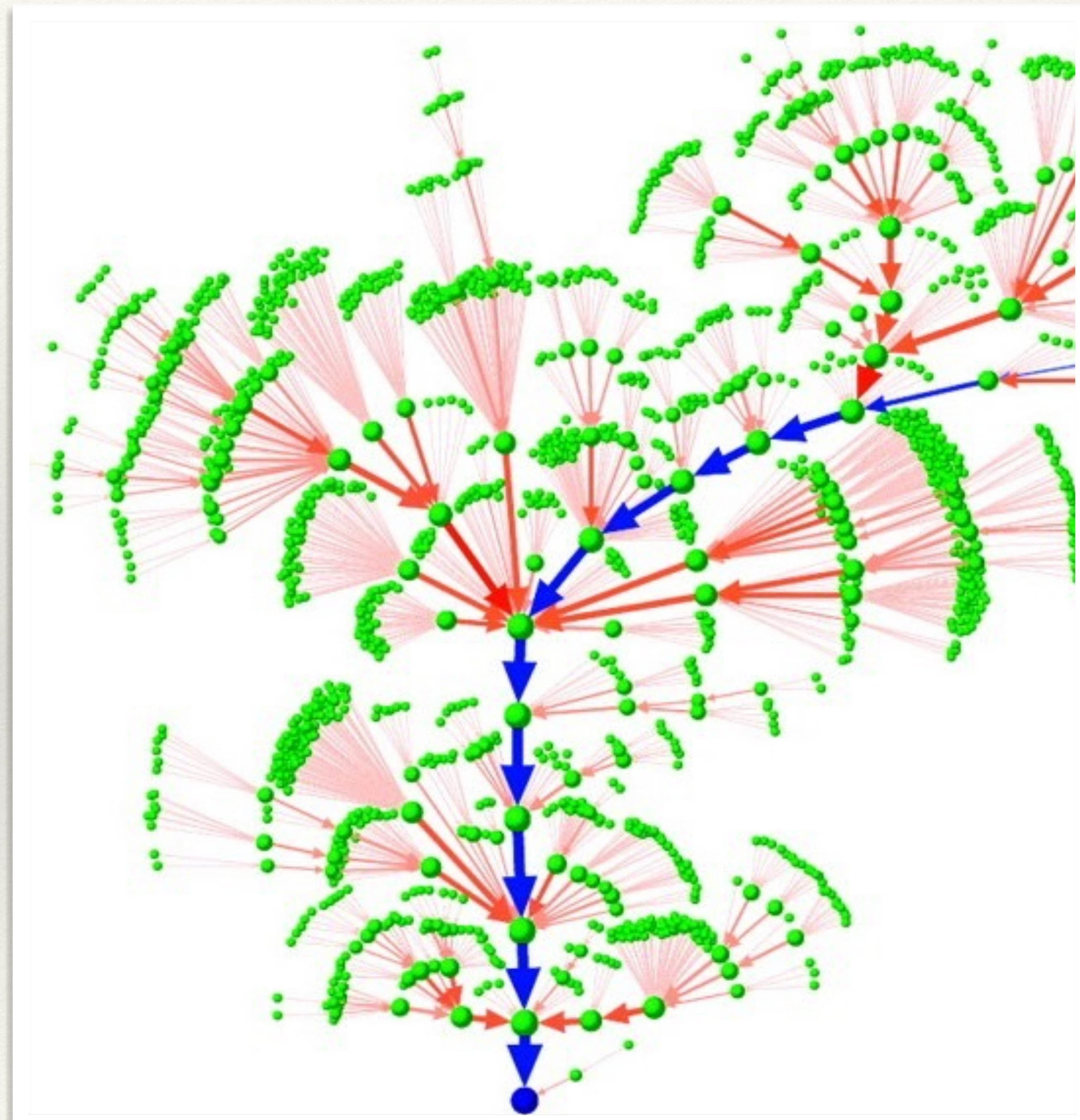
Models: why?



State space of the cellular cycle of the budding yeast [LLL+04]

Models: why?

- Objectives:
  - **Check** whether some expected / unexpected **behaviors** are **possible**
  - Identify **attractors, oscillations,** etc.
  - Understand which are the components **at the origin** of some behaviors
- Problem: how to **circumvent the combinatorial explosion** of the number of behaviors?



State space of the cellular cycle of the budding yeast [LLL+04]

Modeling the dynamics

$t_1\ t_2\ t_3\ t_1\ t_1\ t_5$ ...

Discrete event systems:

- Modeled through transition systems
- Capture the **chronology** (the order) of the events

$(t_1,d_1)\ (t_2,d_2)\ (t_3,d_3)\ (t_1,d_4)\ (t_1,d_5)\ (t_5,d_6)$ ...

Timed systems:

- Modeled through timed transition systems
- Capture the **chronometry** of the events (with their dates)

Scientific challenge

Goals: build **predictive** dynamic **models** from **time series data**

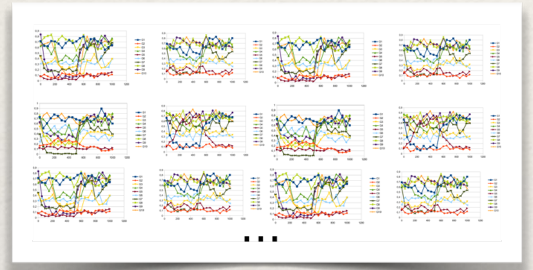
Our motto: **logical modeling** has its own merits w.r.t. continuous approaches (e.g., ODE)

Underlying questions:

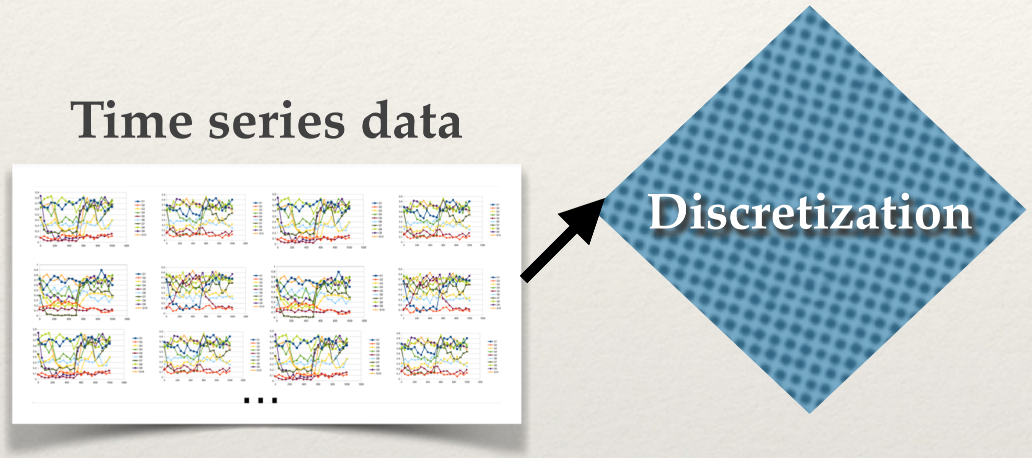
- What are **efficient algorithms** for **discretization** and **learning?**
- What is a *good* predictive model?
- How to validate the **benefits** of our algorithms on "**real-life**" data?

Our scientific challenge

Time series data

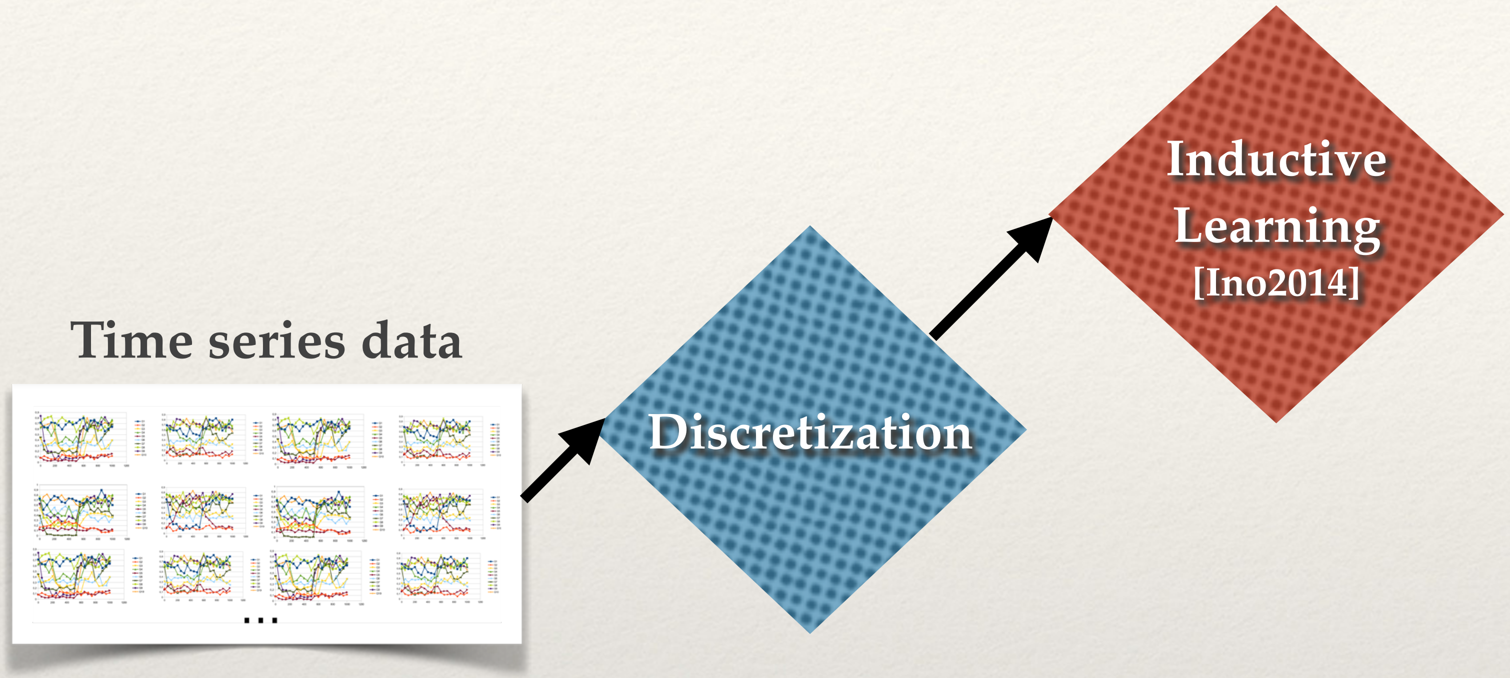


Our scientific challenge



7

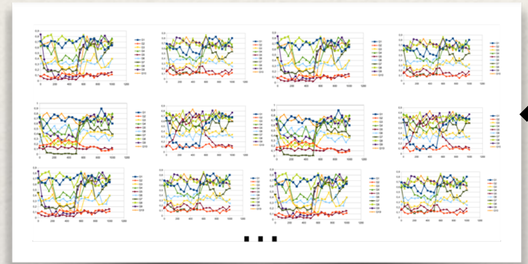
Our scientific challenge



7

Our scientific challenge

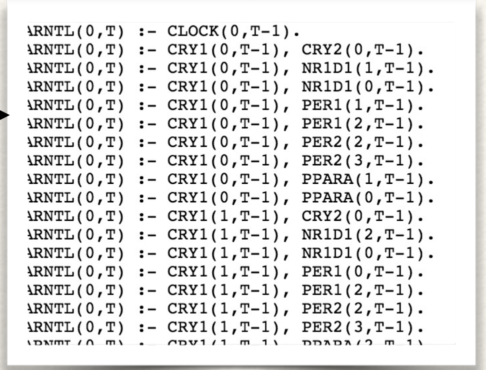
Time series data



Discretization

Inductive Learning [Ino2014]

Dynamical models



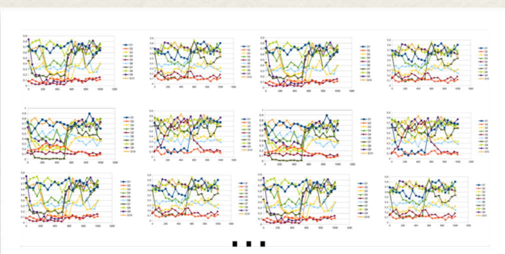
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
```

Logic programs

7

Our scientific challenge

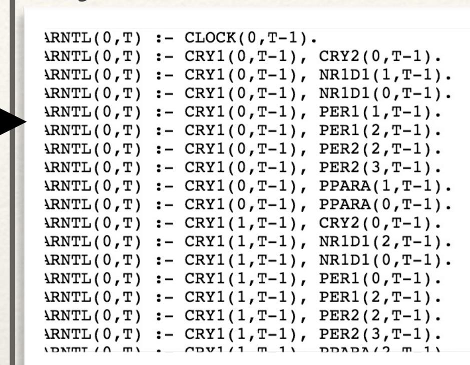
Time series data



Discretization

Inductive Learning [Ino2014]

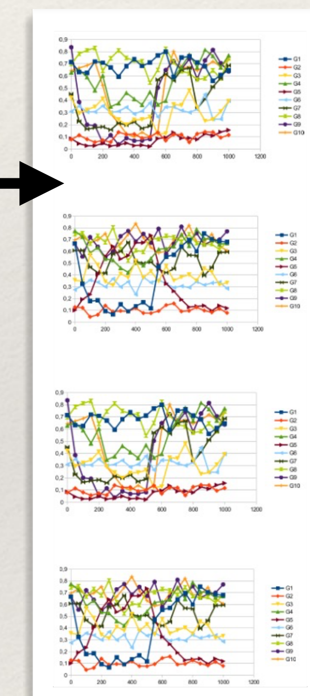
Dynamical models



```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
```

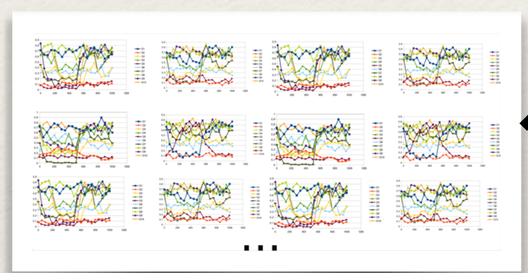
Logic programs

Analysis of dynamic behavior



Our scientific challenge

Time series data



Discretization

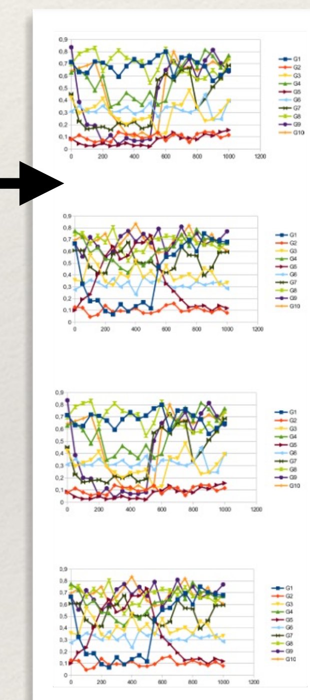
Inductive Learning [Ino2014]

Dynamical models

```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
```

Logic programs

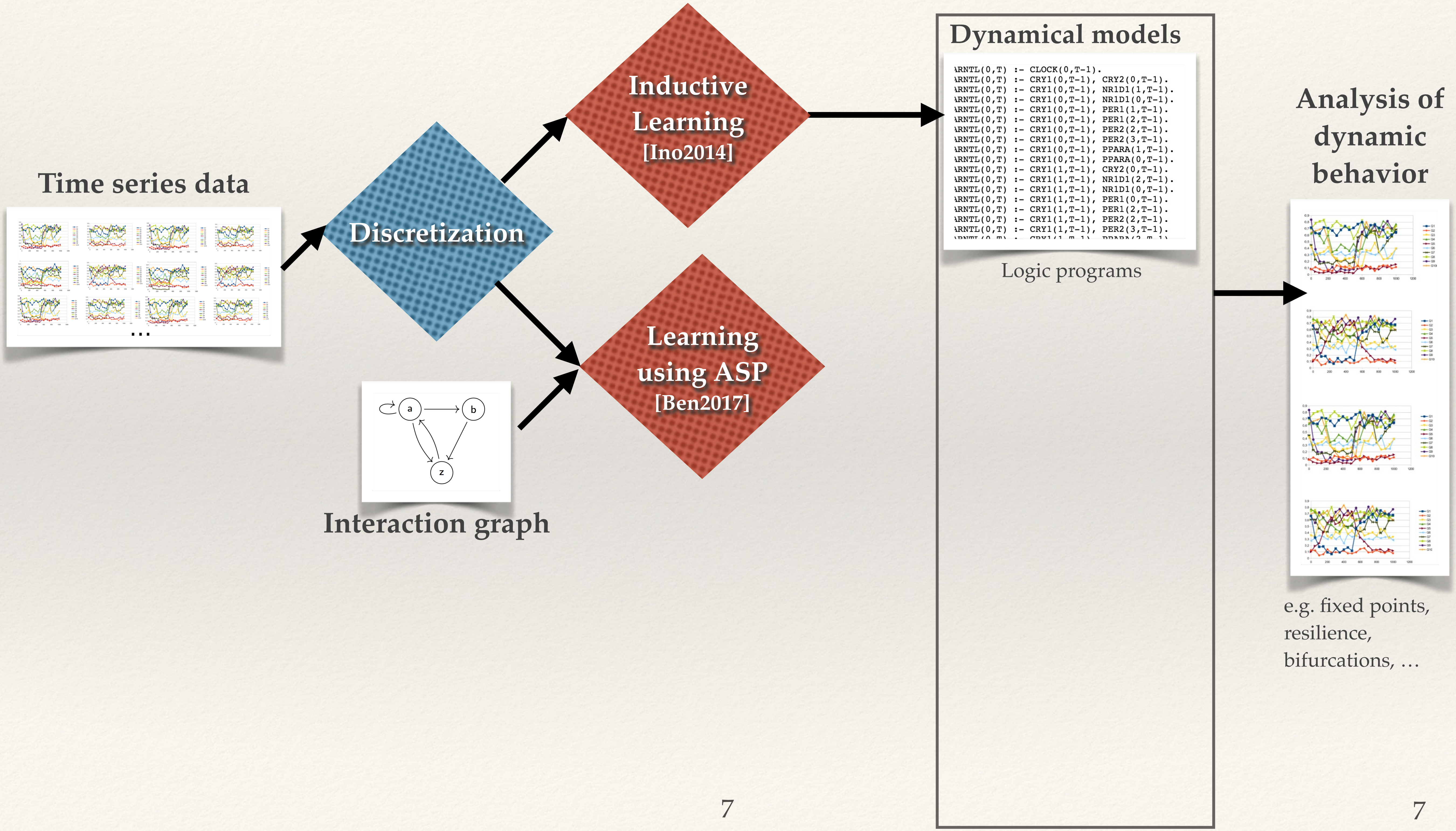
Analysis of dynamic behavior



e.g. fixed points, resilience, bifurcations, …

7

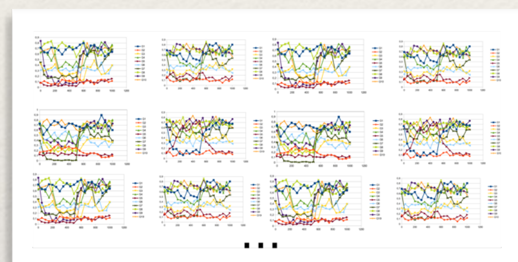
Our scientific challenge



7

Our scientific challenge

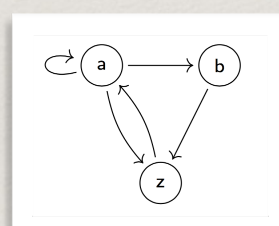
Time series data



Discretization

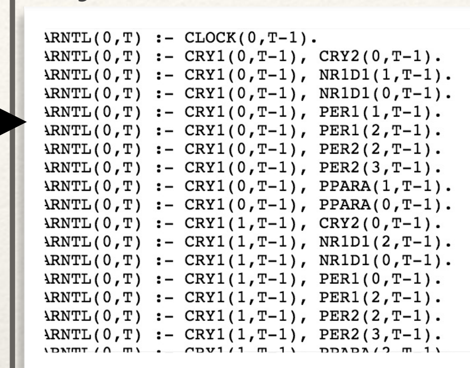
Inductive Learning [Ino2014]

Learning using ASP [Ben2017]



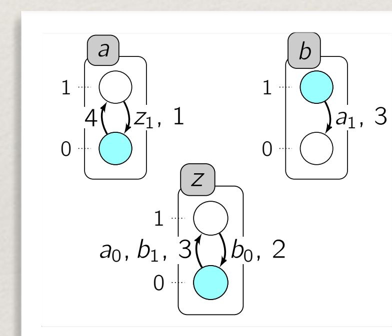
Interaction graph

Dynamical models



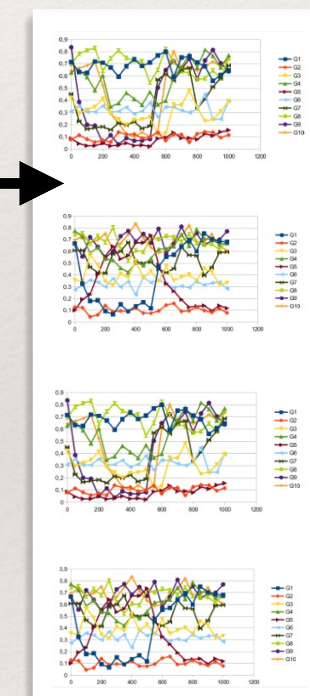
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
```

Logic programs



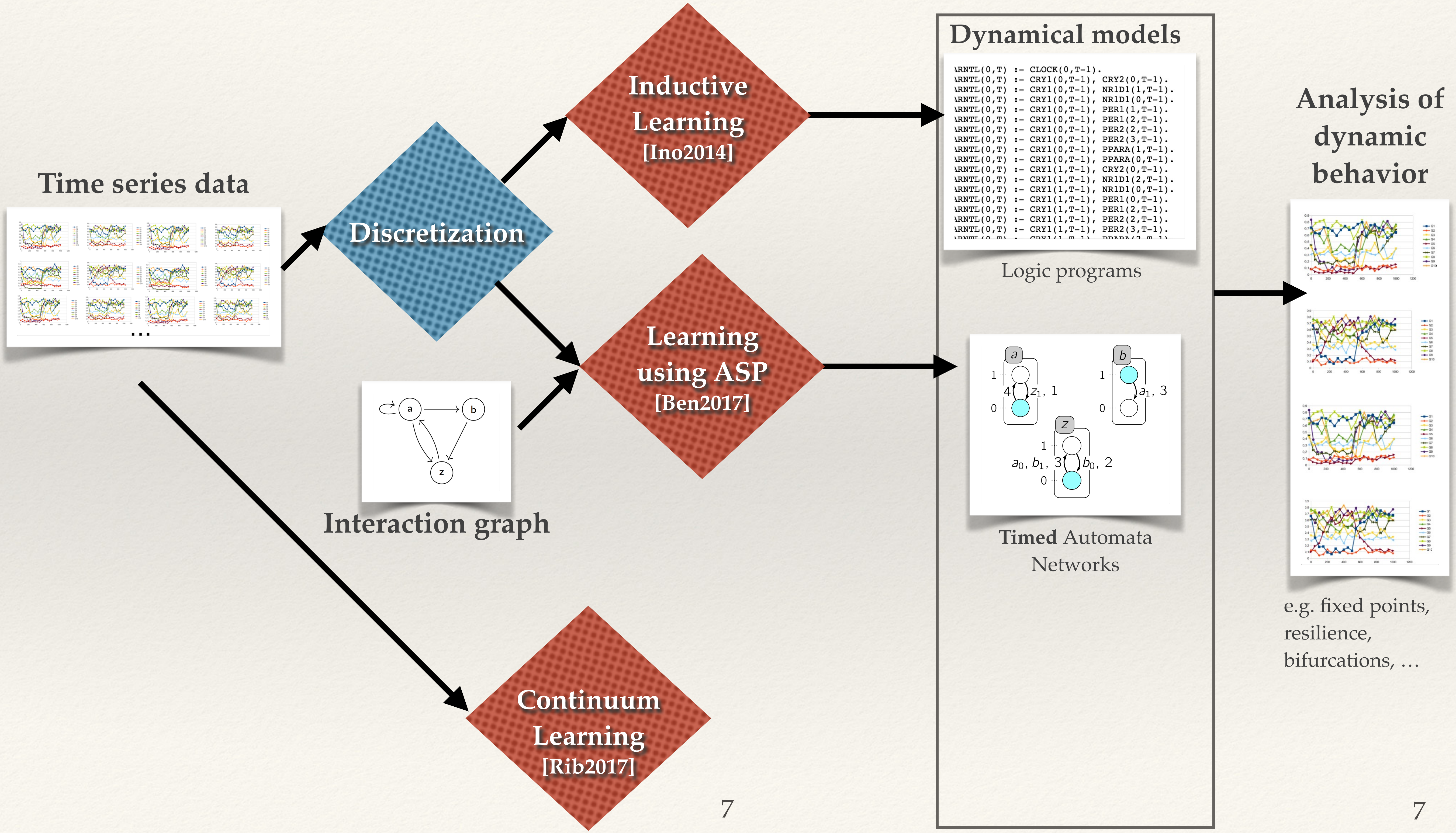
Timed Automata Networks

Analysis of dynamic behavior

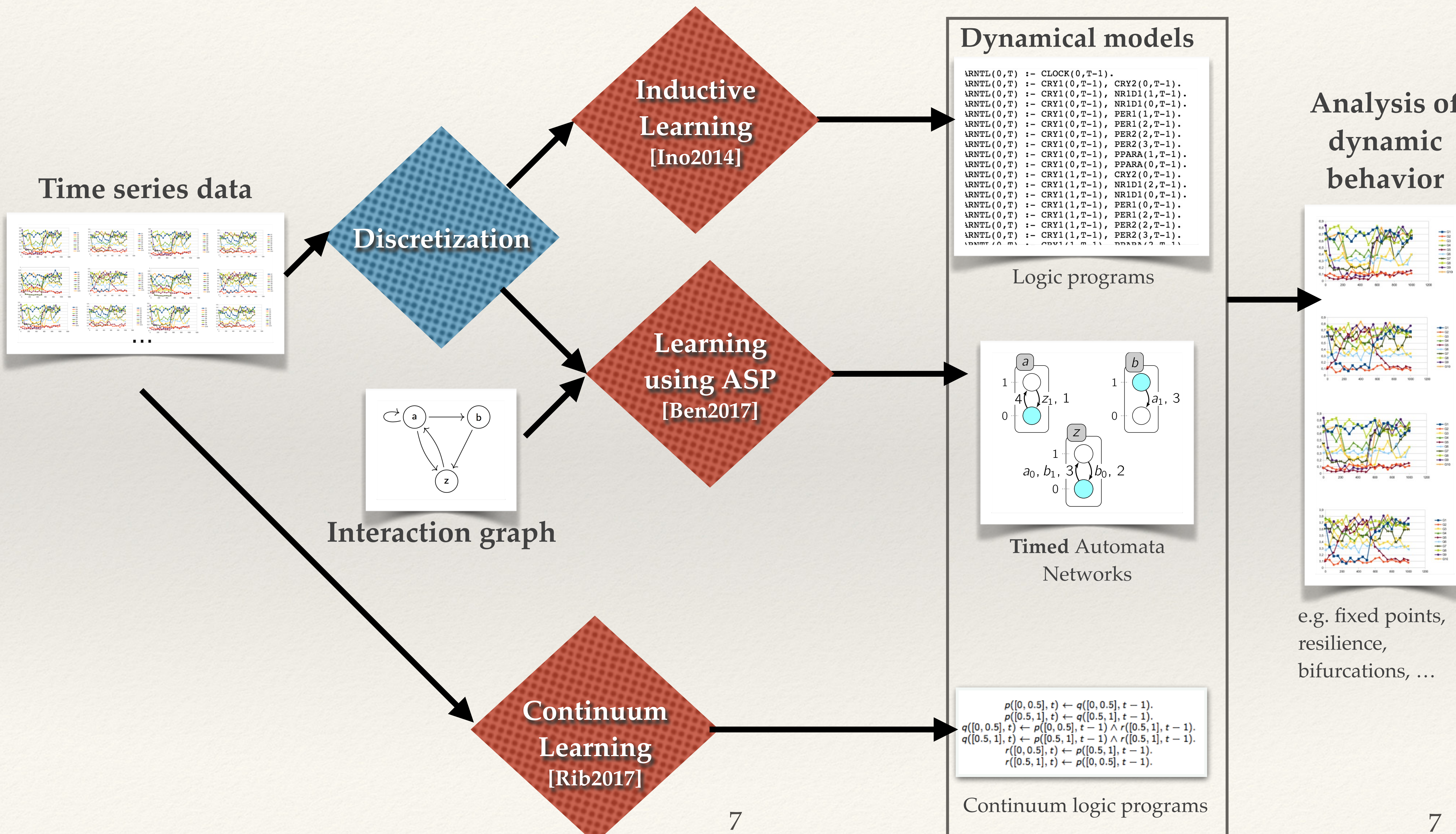


e.g. fixed points, resilience, bifurcations, …

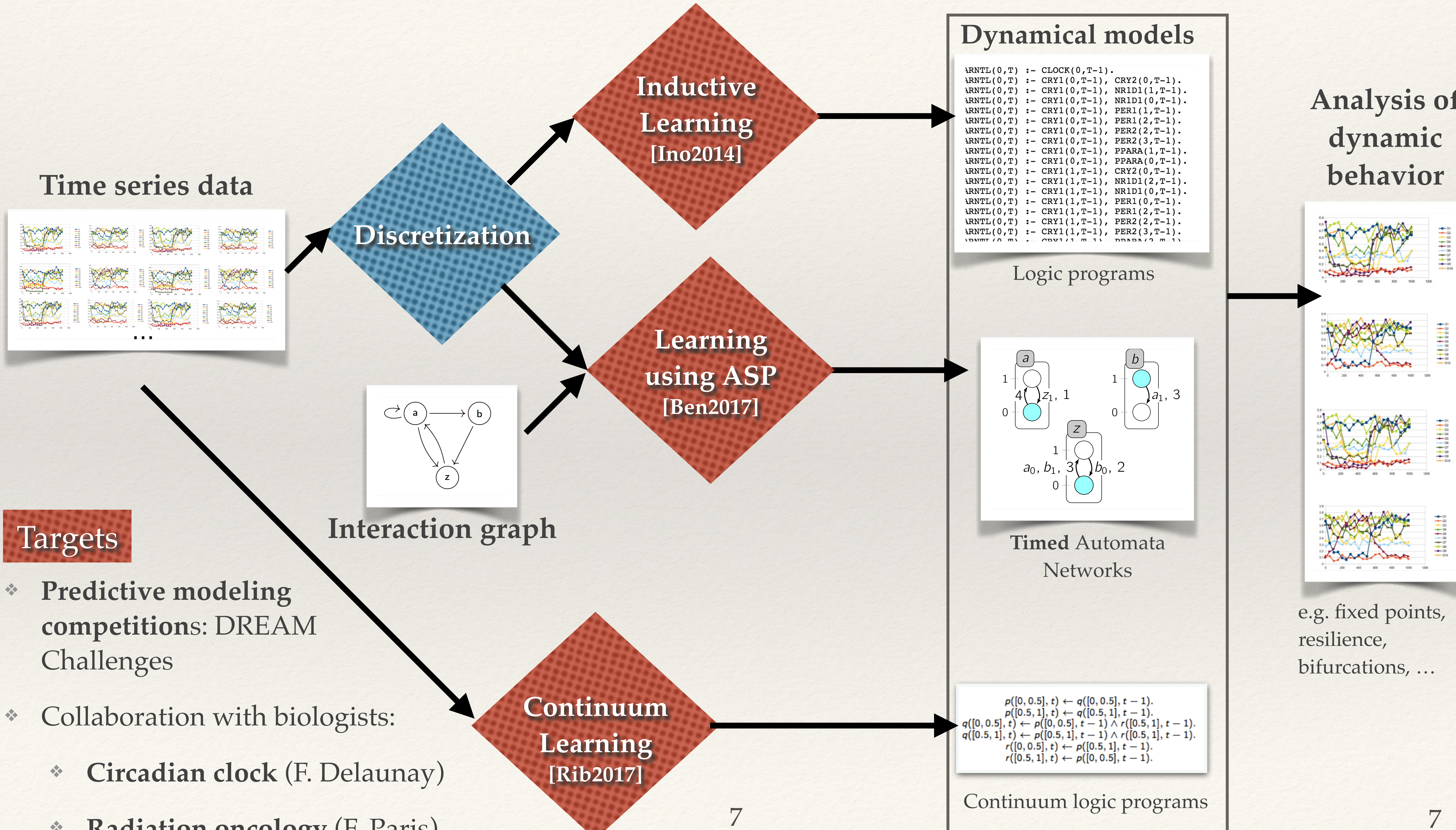
Our scientific challenge



Our scientific challenge



Our scientific challenge



Targets

- **Predictive modeling competitions**: DREAM Challenges
- Collaboration with biologists:
  - **Circadian clock** (F. Delaunay)
  - **Radiation oncology** (F. Paris)

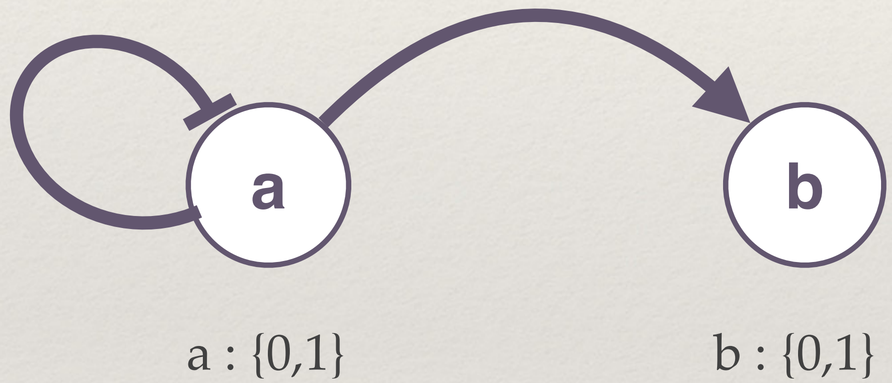
Overview

- Modeling frameworks for dynamical analysis
- Analysis of formal models
- Inference and Learning approaches
- Application to biological case studies

Overview

- **Modeling frameworks for dynamical analysis**
- Analysis of formal models
- Inference and Learning approaches
- Application to biological case studies

Principles of abstraction



Example of regulatory Boolean network

Abstract:

- Components
- Concentrations
- Interactions
- Time
- Order

Modeling the dynamics

$t_1\ t_2\ t_3\ t_1\ t_1\ t_5$ ...

Discrete event systems:

- Modeled through transition systems
- Capture the **chronology** (the order) of the events

Automata Networks (a.k.a. Process Hitting), **Boolean Networks / Thomas networks**, **Logic Programs**

$(t_1,d_1)\ (t_2,d_2)\ (t_3,d_3)\ (t_1,d_4)\ (t_1,d_5)\ (t_5,d_6)$ ...

Timed systems:

- Modeled through timed transition systems
- Capture the **chronometry** of the events (with their dates)

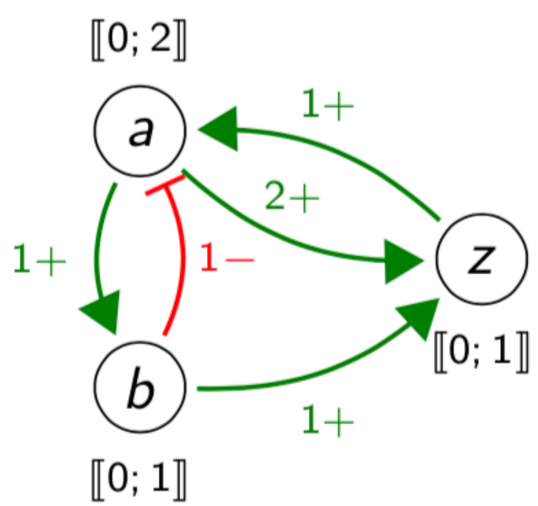
Timed Automata Networks

Different modeling frameworks

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 $\mathrm{dom}(a) = [\![0; 2]\!]$
- Discrete parameters / evolution functions $f^a : \mathcal{S} \to \mathrm{dom}(a)$
- Signs & thresholds on the edges (redundant) $a \xrightarrow{2+} z$



a	f^b
0	**0**
1	**1**
2	**1**

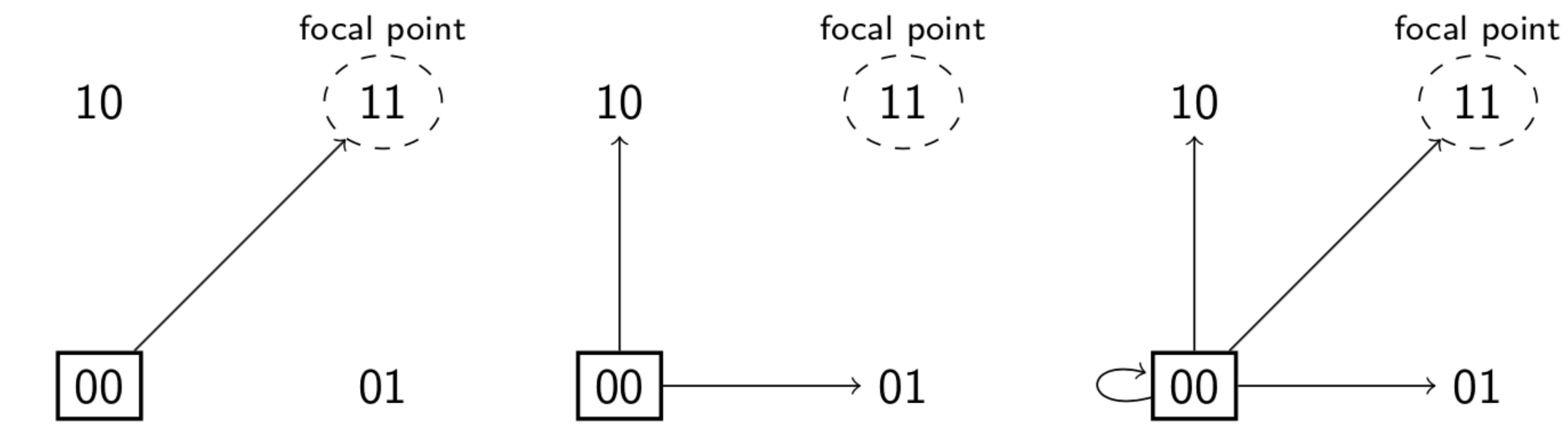
z	b	f^a
0	0	**1**
0	1	**0**
1	0	**1**
1	1	**2**

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

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

Choice of an appropriate semantics

Semantics



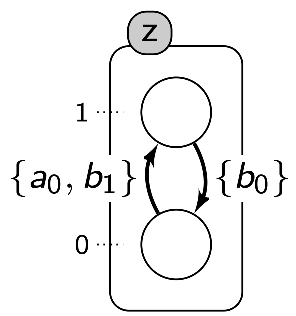
Synchronous **Asynchronous** **Generalized**

Other semantics:

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

Different modeling frameworks

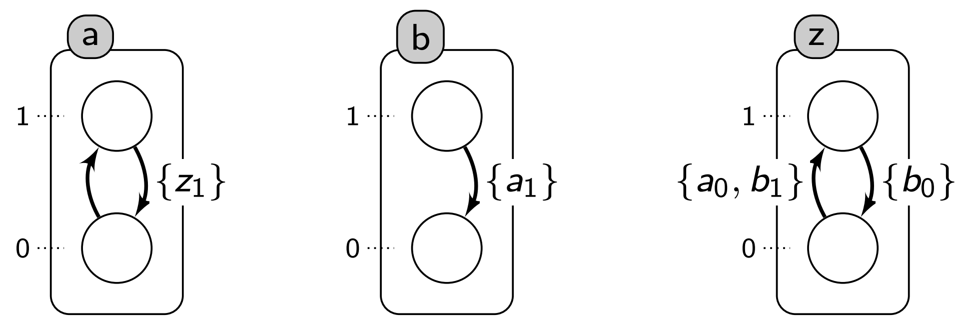
Automata Networks



Automata: components a, b, z

Different modeling frameworks

Automata Networks

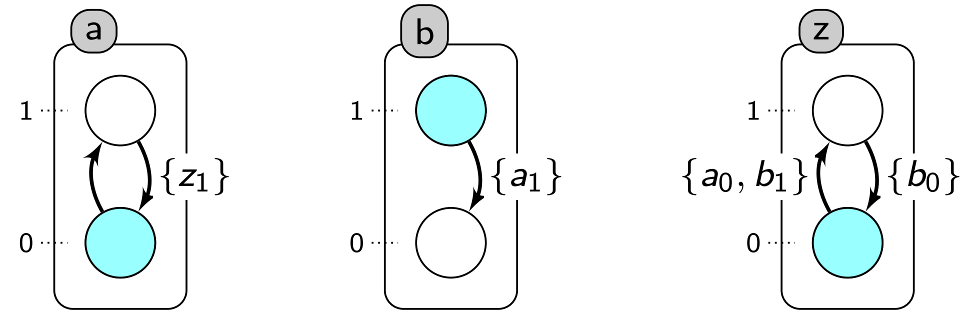


Automata: components a, b, z

Local states: levels of expression $\{a_0, a_1\}$ $\{b_0, b_1\}$ $\{z_0, z_1\}$

Different modeling frameworks

Automata Networks



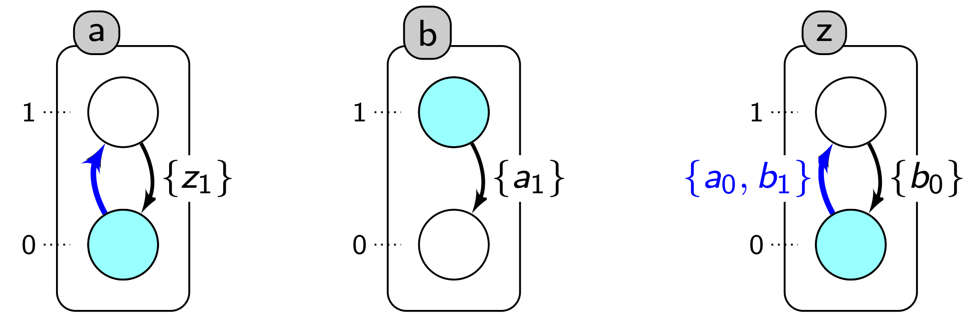
Automata: components $a,\ b,\ z$

Local states: levels of expression $\{a_0,\ a_1\}$ $\{b_0,\ b_1\}$ $\{z_0,\ z_1\}$

State: set of active local states $\langle a_0, b_1, z_0 \rangle \sim$ 010

Different modeling frameworks

Automata Networks



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

Local states: levels of expression $\{a_0, \ a_1\}$ $\{b_0, \ b_1\}$ $\{z_0, \ z_1\}$

State: set of active local states $\langle a_0, b_1, z_0 \rangle \sim$ 010

Local transitions:

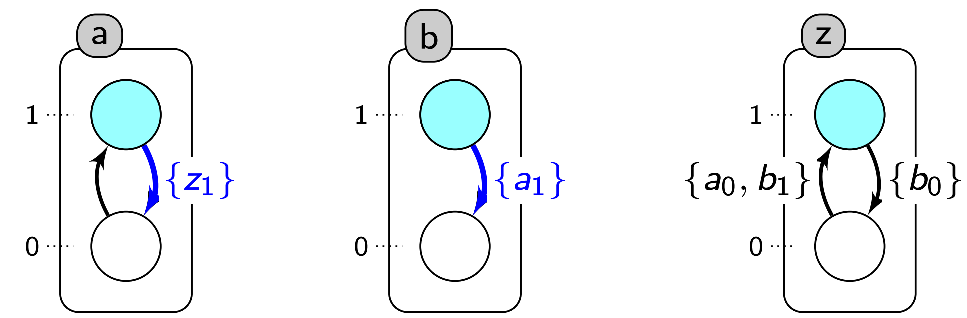
$\tau_1 = a_0 \longrightarrow a_1; \ \tau_3 = a_1 \xrightarrow{\{z_1\}} a_0; \ \tau_4 = b_1 \xrightarrow{\{a_1\}} b_0; \ \tau_2 = z_0 \xrightarrow{\{a_0, b_1\}} z_1; \ \tau_5 = z_1 \xrightarrow{\{b_0\}} z_0;$

Synchronous : [Kauffman, 1969]

010

Different modeling frameworks

Automata Networks



Automata: components a, b, z

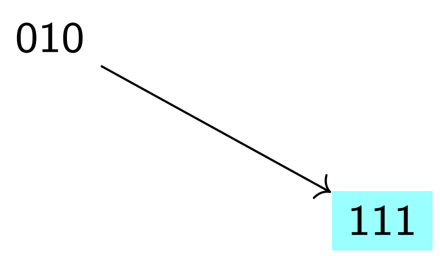
Local states: levels of expression $\{a_0, a_1\}$ $\{b_0, b_1\}$ $\{z_0, z_1\}$

State: set of active local states $\langle a_1, b_1, z_1 \rangle \sim$ 111

Local transitions:

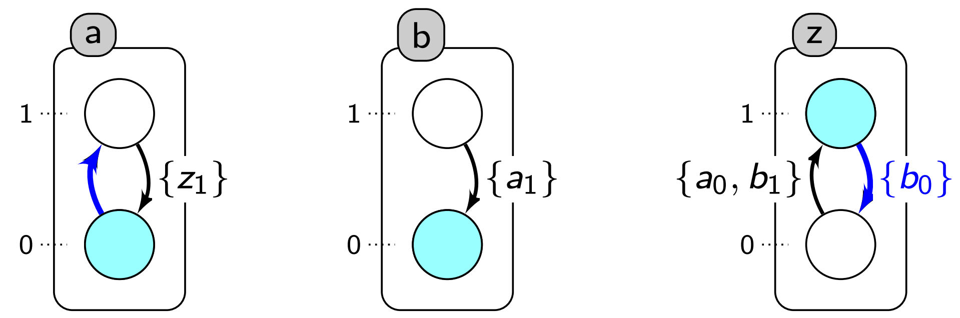
$\tau_1 = a_0 \longrightarrow a_1$; $\tau_3 = a_1 \xrightarrow{\{z_1\}} a_0$; $\tau_4 = b_1 \xrightarrow{\{a_1\}} b_0$; $\tau_2 = z_0 \xrightarrow{\{a_0, b_1\}} z_1$; $\tau_5 = z_1 \xrightarrow{\{b_0\}} z_0$;

Synchronous : [Kauffman, 1969]



Different modeling frameworks

Automata Networks



Automata: components a, b, z

Local states: levels of expression $\{a_0, a_1\}$ $\{b_0, b_1\}$ $\{z_0, z_1\}$

State: set of active local states $\langle a_0, b_0, z_1 \rangle \sim$ 001

Local transitions:

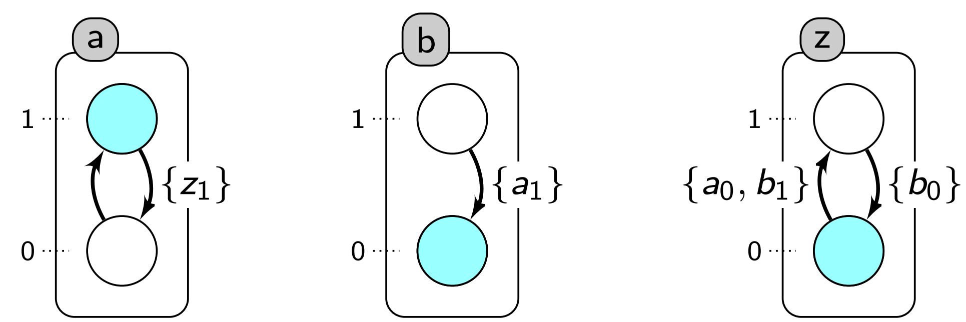
$\tau_1 = a_0 \longrightarrow a_1$; $\tau_3 = a_1 \xrightarrow{\{z_1\}} a_0$; $\tau_4 = b_1 \xrightarrow{\{a_1\}} b_0$; $\tau_2 = z_0 \xrightarrow{\{a_0, b_1\}} z_1$; $\tau_5 = z_1 \xrightarrow{\{b_0\}} z_0$;

Synchronous : [Kauffman, 1969]

010 $\longrightarrow$ 111 $\longrightarrow$ 001

Different modeling frameworks

Automata Networks



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

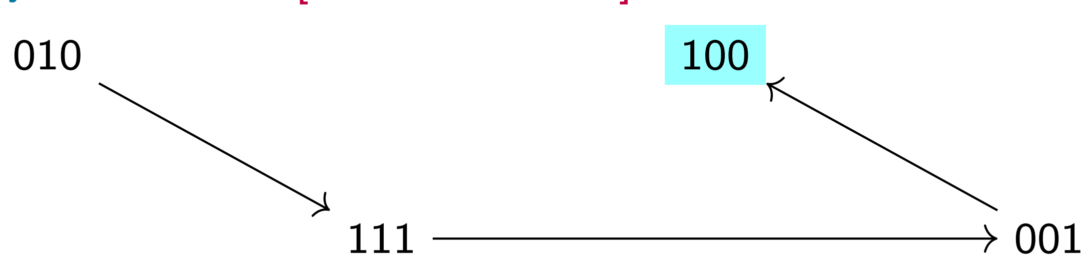
Local states: levels of expression $\{a_0,\ a_1\}$ $\{b_0,\ b_1\}$ $\{z_0,\ z_1\}$

State: set of active local states $\langle a_1, b_0, z_0 \rangle \sim$ 100

Local transitions:

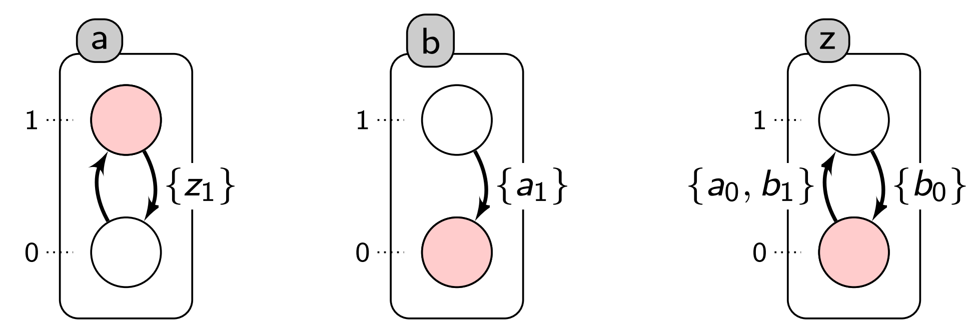
$\tau_1 = a_0 \longrightarrow a_1$; $\tau_3 = a_1 \xrightarrow{\{z_1\}} a_0$; $\tau_4 = b_1 \xrightarrow{\{a_1\}} b_0$; $\tau_2 = z_0 \xrightarrow{\{a_0, b_1\}} z_1$; $\tau_5 = z_1 \xrightarrow{\{b_0\}} z_0$;

Synchronous : [Kauffman, 1969]



Different modeling frameworks

Automata Networks



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

Local states: levels of expression $\{a_0, \ a_1\}$ $\{b_0, \ b_1\}$ $\{z_0, \ z_1\}$

State: set of active local states $\langle a_1, b_0, z_0 \rangle \sim$ 100

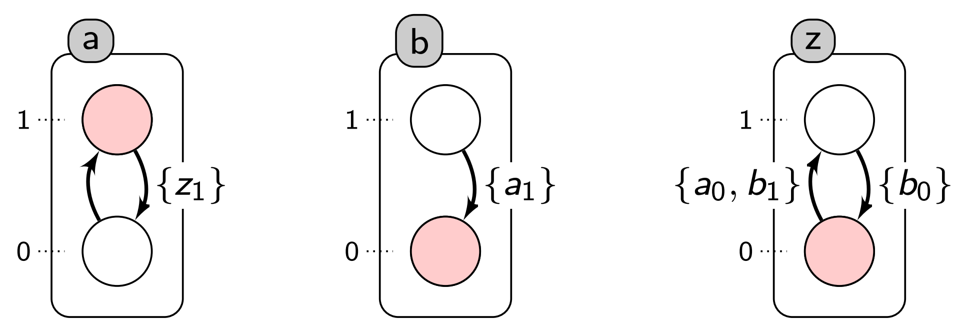
Local transitions:

$\tau_1 = a_0 \longrightarrow a_1; \ \tau_3 = a_1 \xrightarrow{\{z_1\}} a_0; \ \tau_4 = b_1 \xrightarrow{\{a_1\}} b_0; \ \tau_2 = z_0 \xrightarrow{\{a_0, b_1\}} z_1; \ \tau_5 = z_1 \xrightarrow{\{b_0\}} z_0;$

Synchronous : [Kauffman, 1969]

Different modeling frameworks

Automata Networks



Automata: components a, b, z

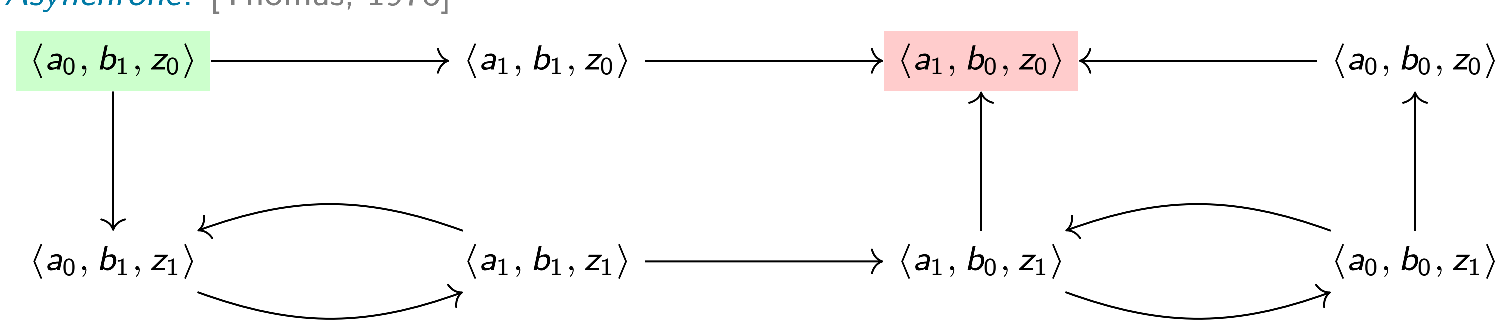
Local states: levels of expression $\{a_0, a_1\}$ $\{b_0, b_1\}$ $\{z_0, z_1\}$

State: set of active local states $\langle a_1, b_0, z_0 \rangle \sim 100$

Local transitions:

$$\tau_1 = a_0 \longrightarrow a_1;\ \tau_3 = a_1 \xrightarrow{\{z_1\}} a_0;\ \tau_4 = b_1 \xrightarrow{\{a_1\}} b_0;\ \tau_2 = z_0 \xrightarrow{\{a_0, b_1\}} z_1;\ \tau_5 = z_1 \xrightarrow{\{b_0\}} z_0;$$

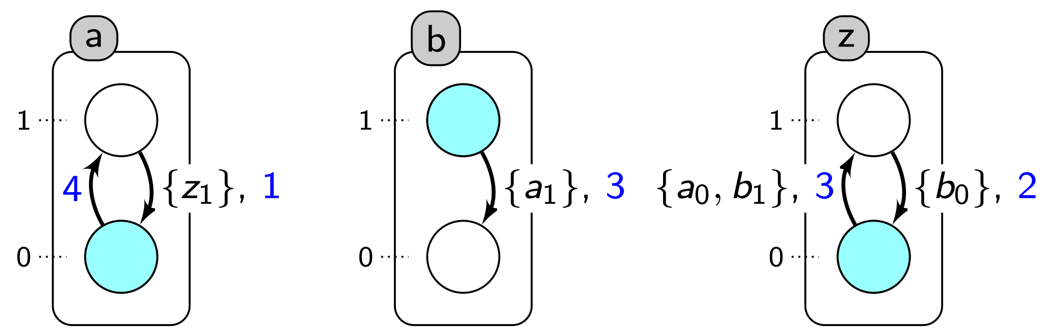
Asynchrone: [Thomas, 1978]



Different modeling frameworks

Timed Automata Networks

Definition



Refinement of th AN dynamics by delays integration : $\delta \in \mathbb{N}$

each delay δ is specific of a local transition τ

and δ represents the necessary time duration of the local transition τ to occur.

Example:

$b_1 \xrightarrow[3]{\{a_1\}} b_0$: 3 represents the time it takes for changing b from the local level 1 to 0.

$\Rightarrow$ How to calculate the states graph of the refined dynamics of a T-AN ?

$\Rightarrow$ What would be the semantics of the dynamics of a T-AN that shows this refinement?

Logic programs

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

Where:

$x_0, x_1, x_2, \ldots, x_n$: Variables

$val_0, val_1, val_2, \ldots, val_n$: Values

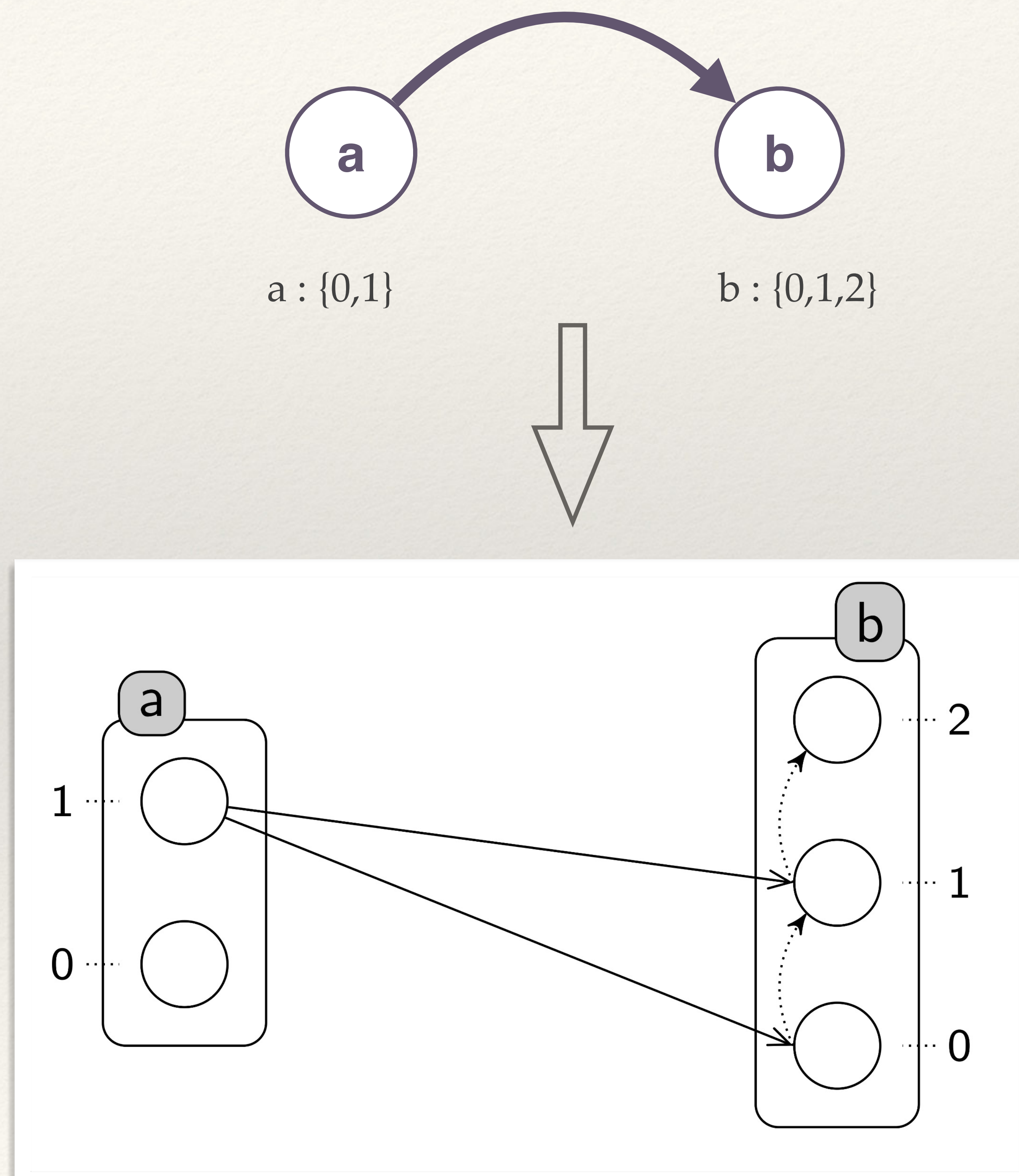
t: Time step

$\leftarrow$ Reverse implication

All atoms in the body are in conjunction

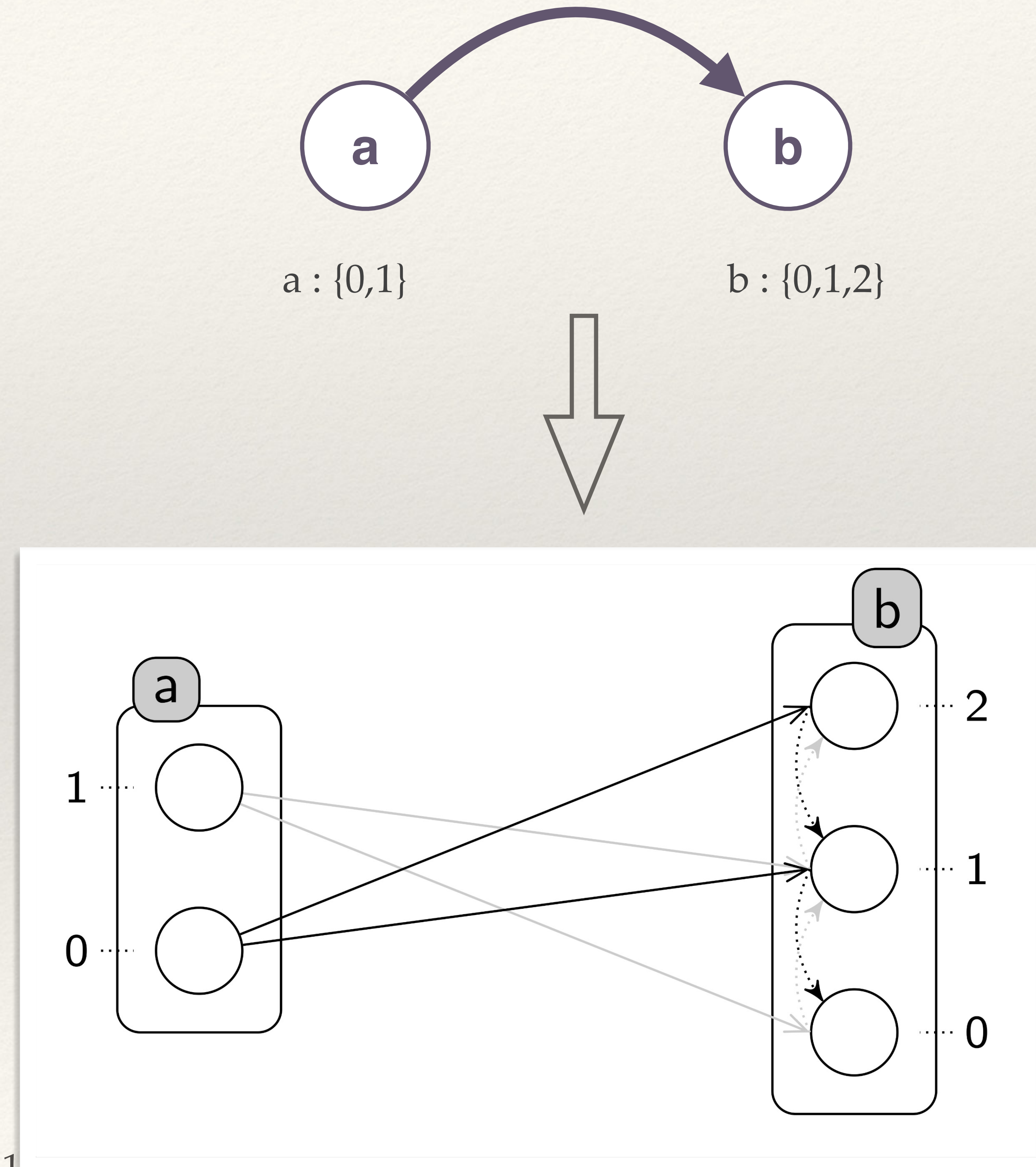
Hypotheses for translation from biology

- Discretization of the **concentration (expression)** of a component: Boolean or multi-valued
- Discretization of **time**: event-bases
- **Unitarian** dynamics
- Semantics: **non-deterministic asynchronous**



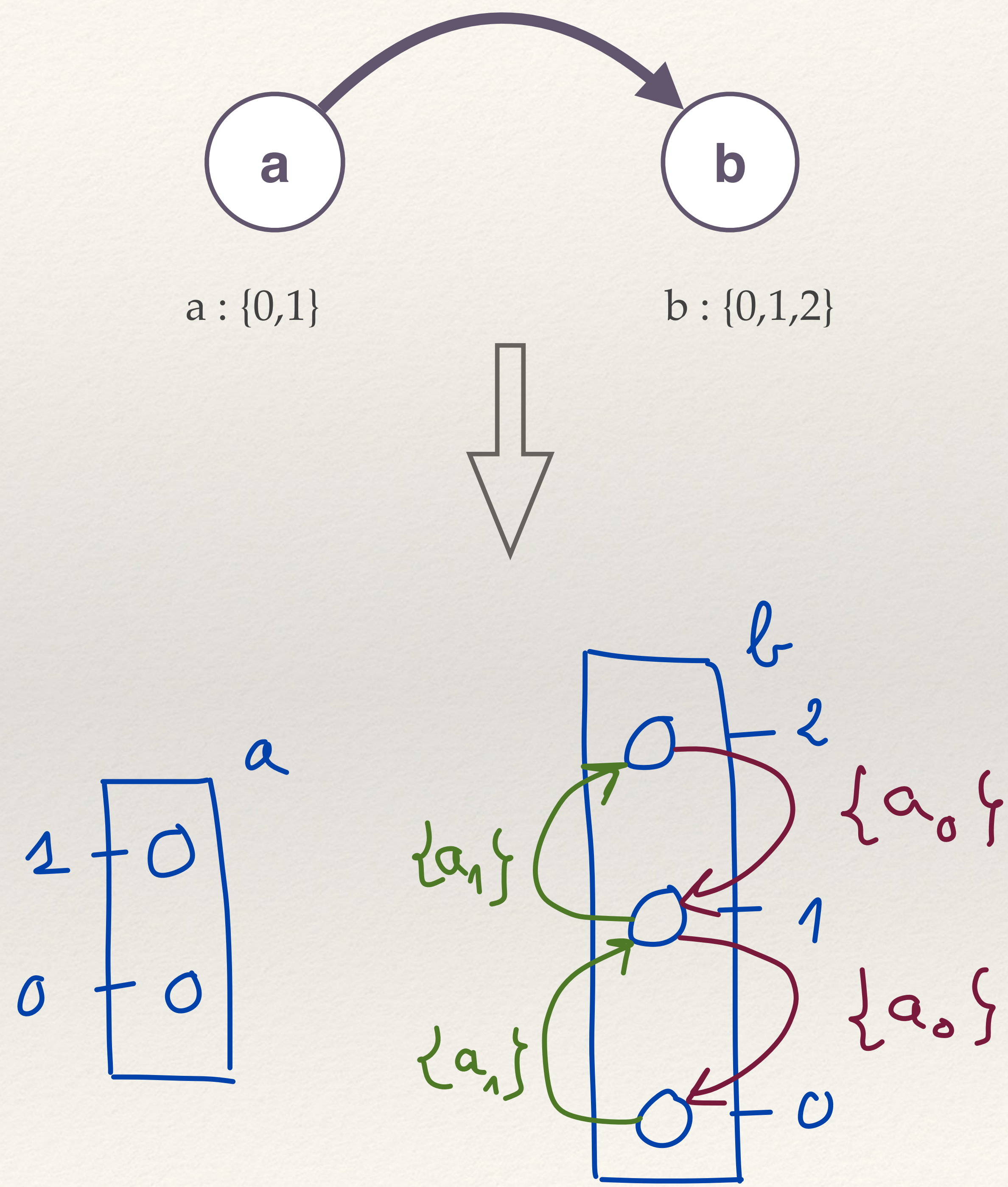
Hypotheses for translation from biology

- Discretization of the **concentration (expression)** of a component: Boolean or multi-valued
- Discretization of **time**: event-bases
- **Unitarian** dynamics
- Semantics: **non-deterministic asynchronous**



Hypotheses for translation from biology

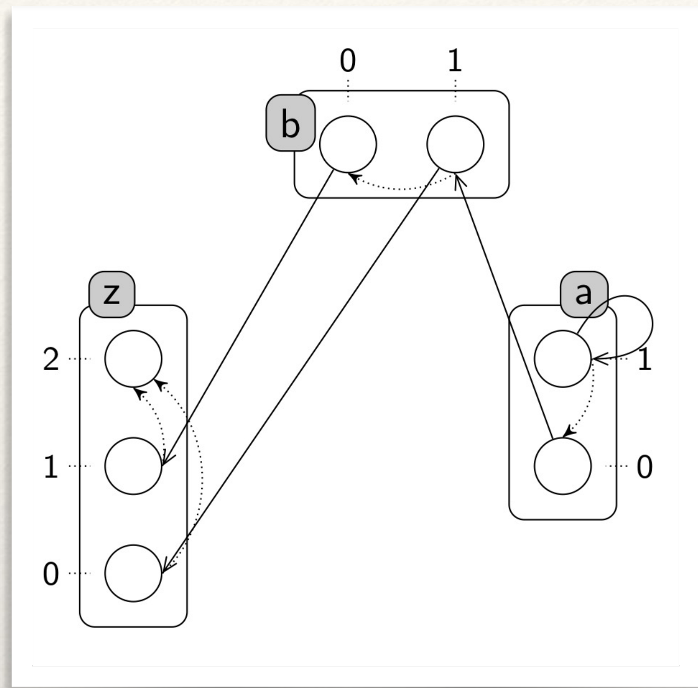
- Discretization of the **concentration** (**expression**) of a component: Boolean or multi-valued
- Discretization of **time**: event-bases
- **Unitarian** dynamics
- Semantics: **non-deterministic asynchronous**



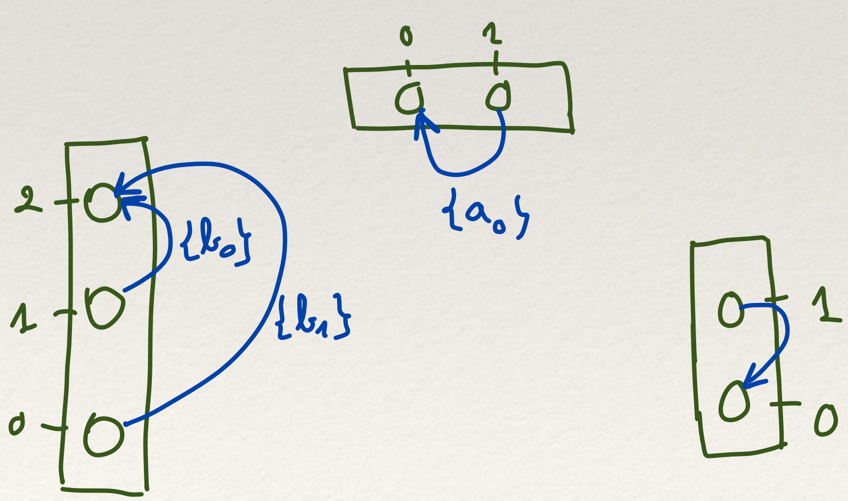
Overview

- Modeling frameworks for dynamical analysis
- **Analysis of formal models**
- Inference and Learning approaches
- Application to biological case studies

An example of static analysis [PMR11a,PCF+14]

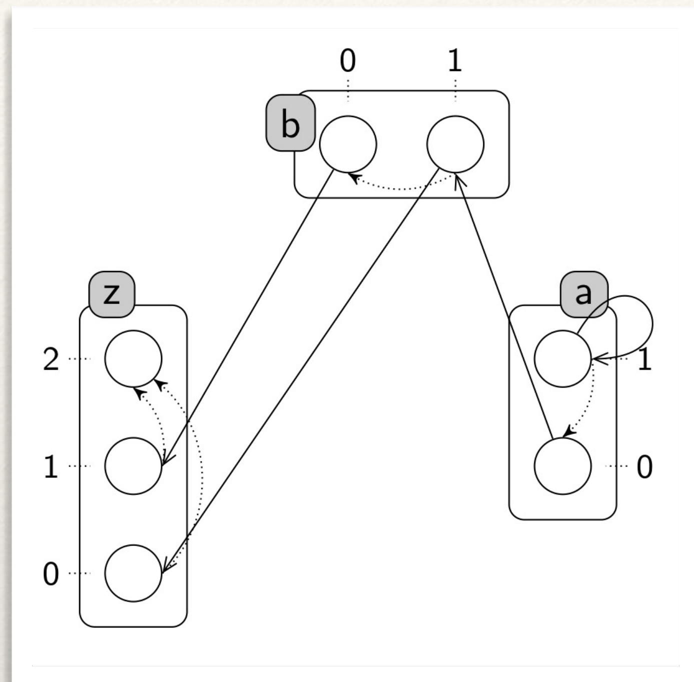


Process Hitting framework



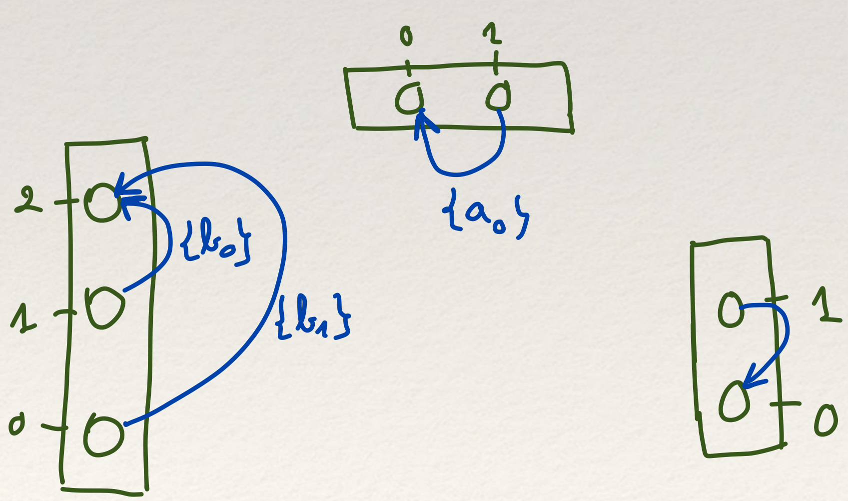
Asynchronous automata network

An example of static analysis [PMR11a,PCF+14]



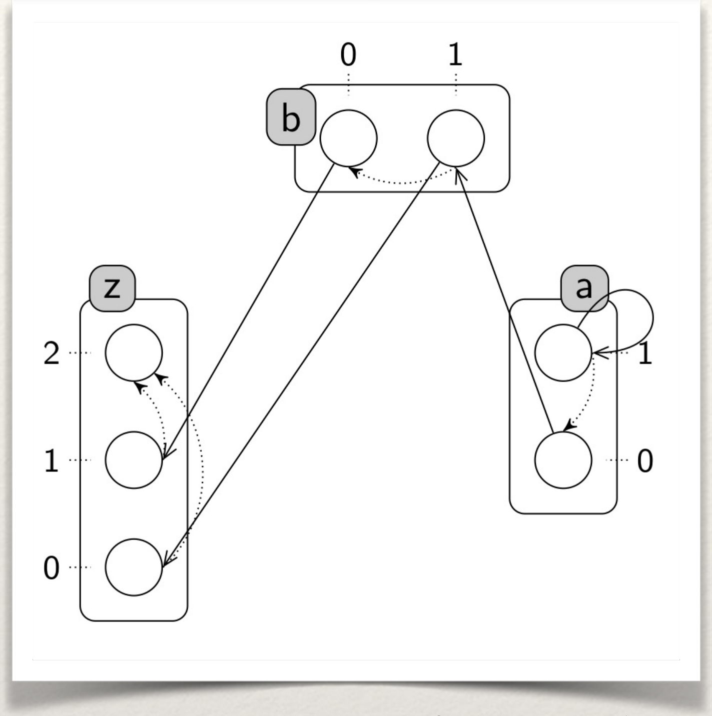
Process Hitting framework

What are the **fixed points?**

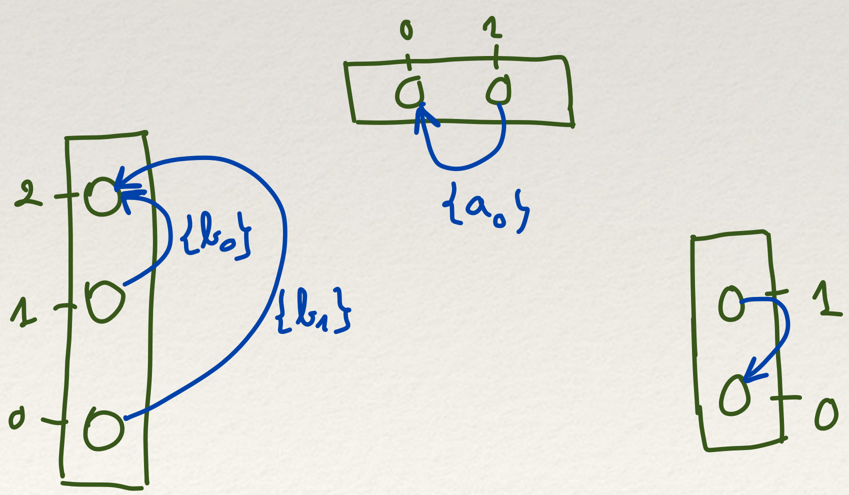


Asynchronous automata network

An example of static analysis
[PMR11a,PCF+14]



Process Hitting framework

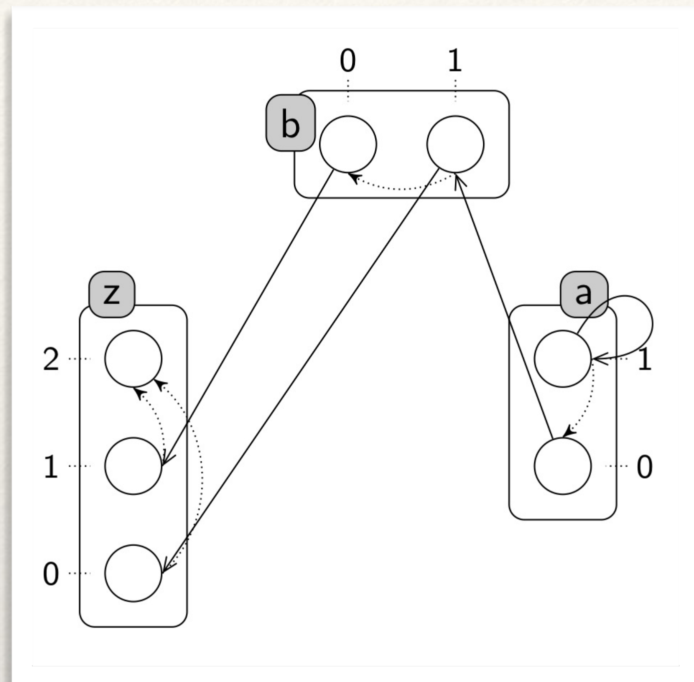


Asynchronous automata network

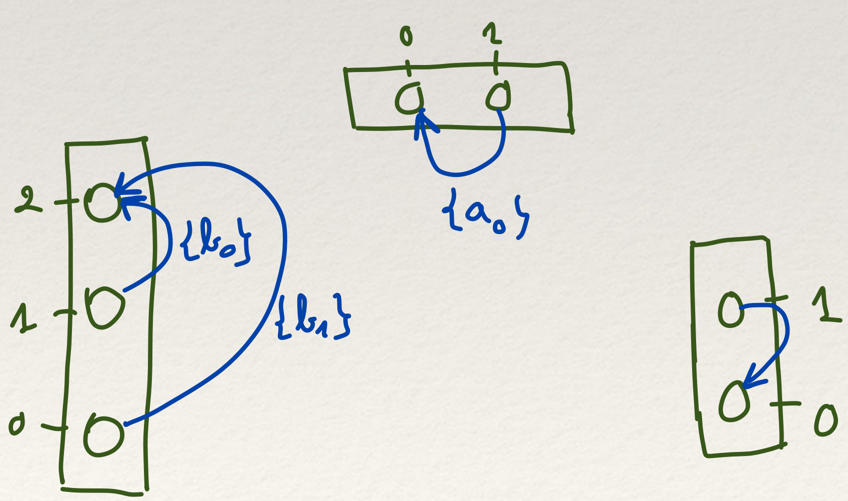
What are the **fixed points?**

- Naive method: compute all possible scenarios

An example of static analysis [PMR11a,PCF+14]



Process Hitting framework



Asynchronous automata network

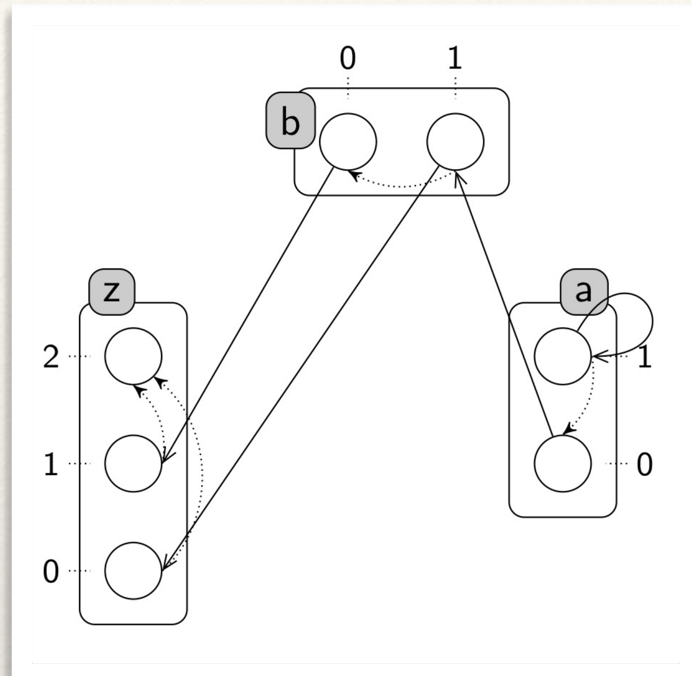
What are the **fixed points?**

- Our proposition: static analysis

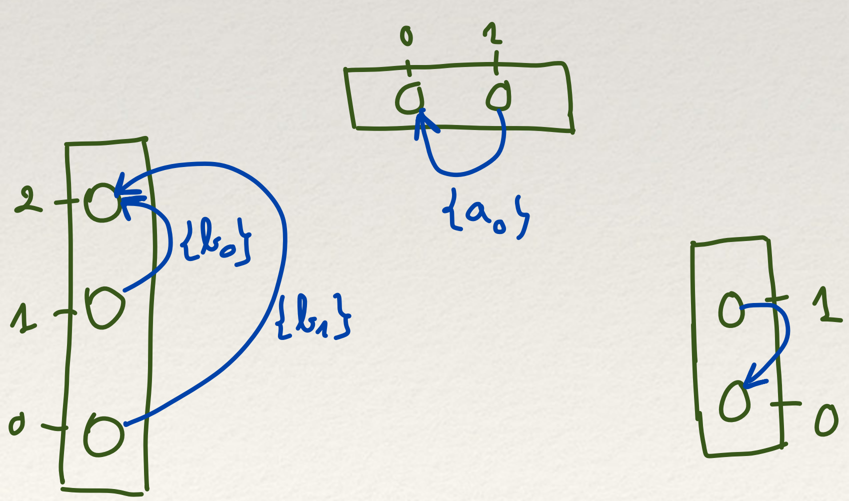
An example of static analysis
[PMR11a,PCF+14]

What are the **fixed points?**

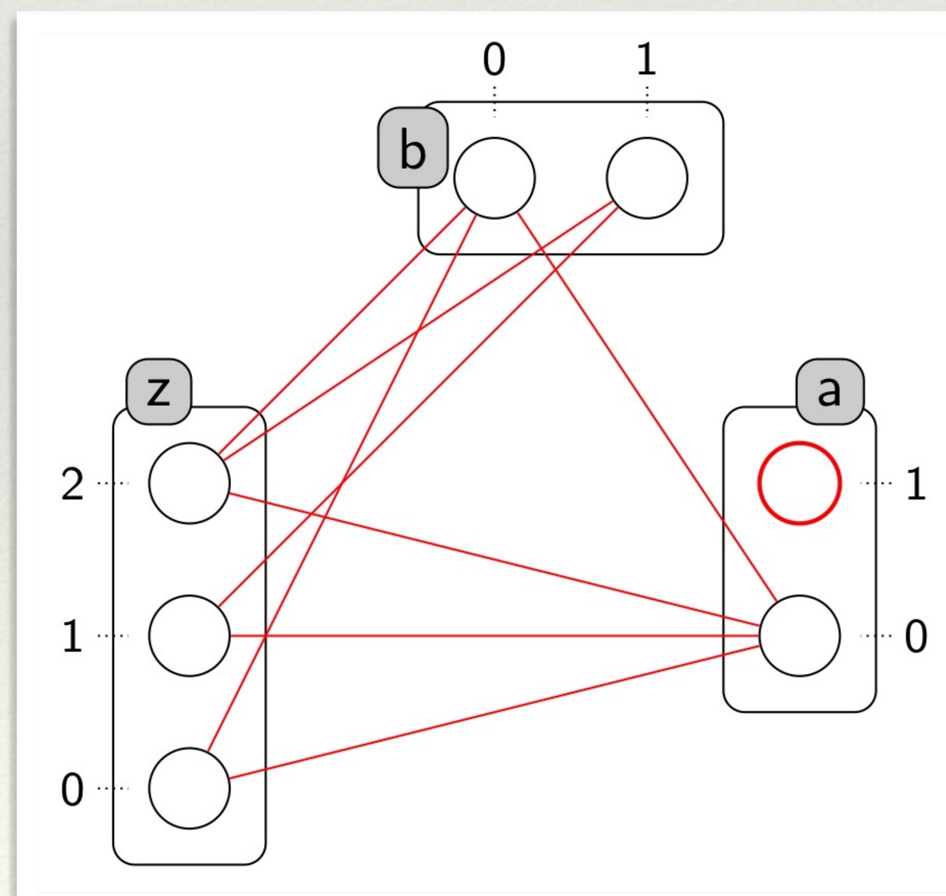
- Our proposition: static analysis
  - Compute hitless graph



Process Hitting framework

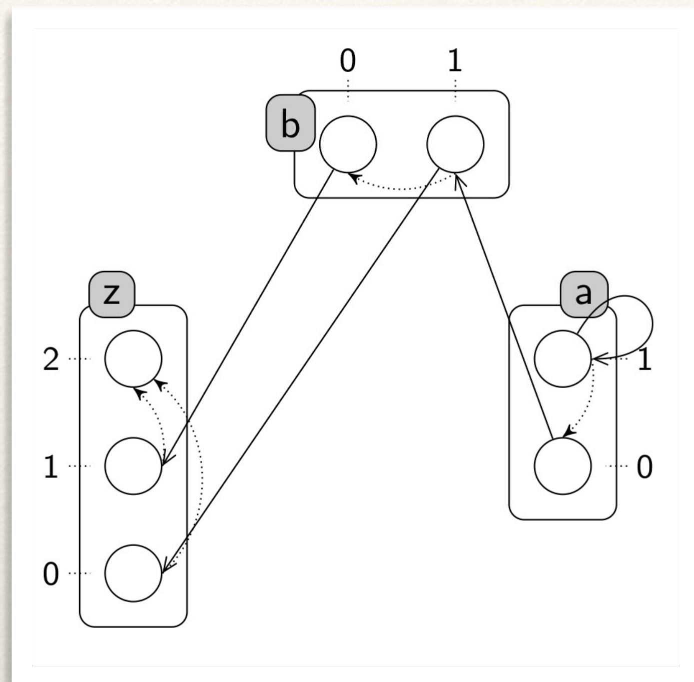


Asynchronous automata network

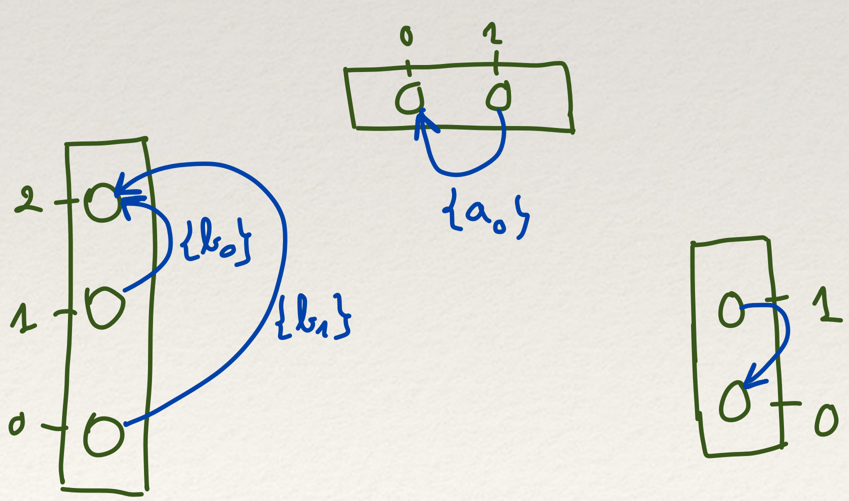


Hitless graph

An example of static analysis
[PMR11a,PCF+14]



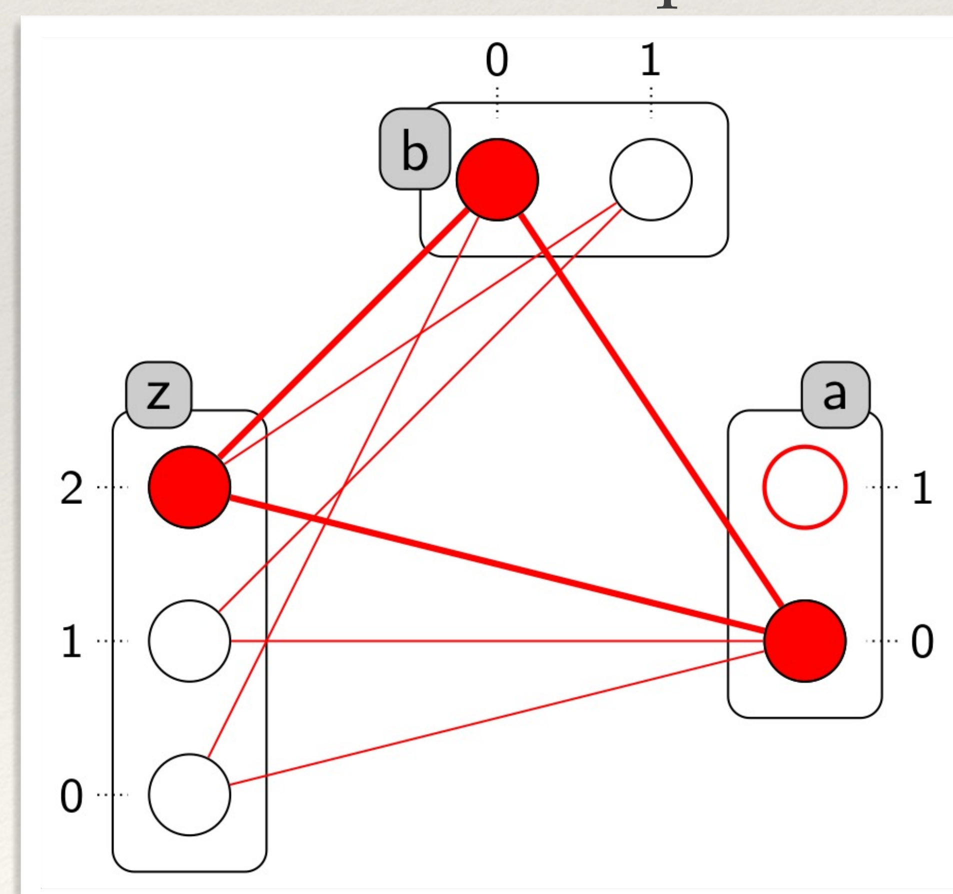
Process Hitting framework



Asynchronous automata network

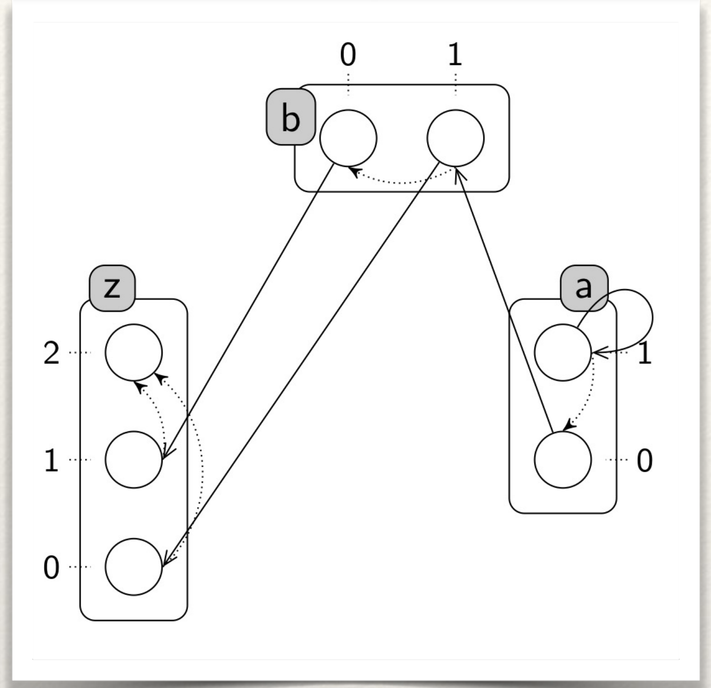
What are the **fixed points?**

- Our proposition: static analysis
  - Compute hitless graph
  - Enumerate its n-cliques



Hitless graph

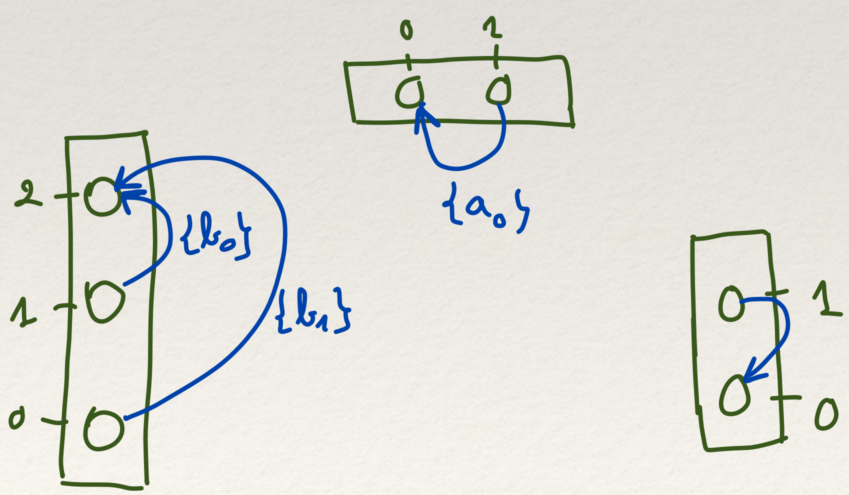
An example of static analysis [PMR11a,PCF+14]



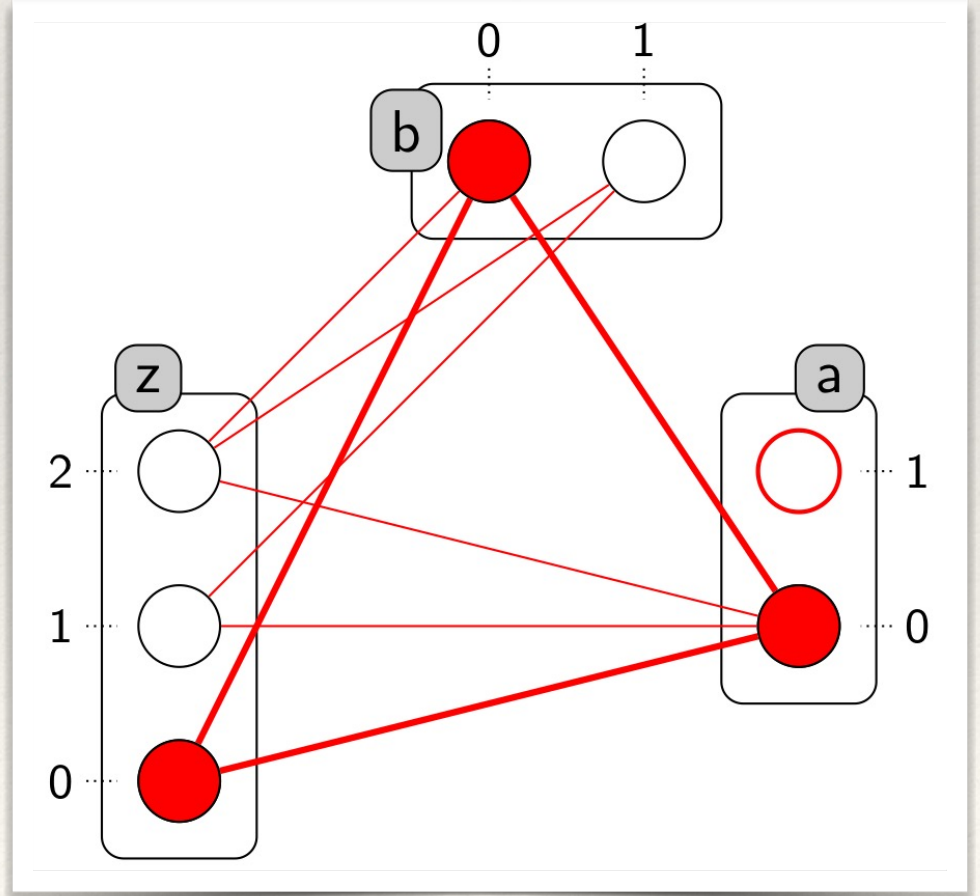
Process Hitting framework

What are the **fixed points?**

- Our proposition: static analysis
  - Compute hitless graph
  - Enumerate its *n*-cliques



Asynchronous automata network

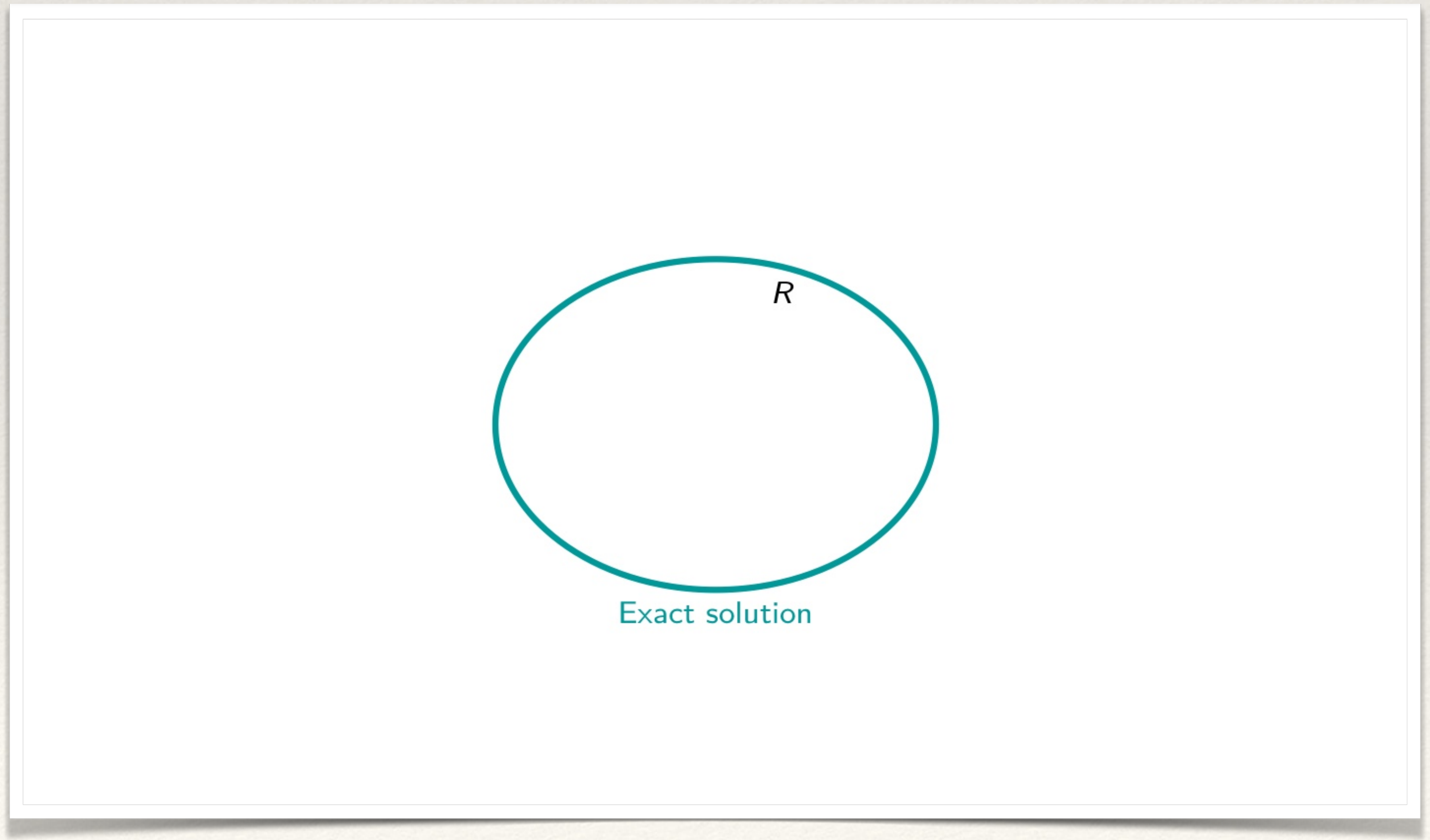


Hitless graph

Reachability analysis [PMR12]

Goal: check properties of the following type
« From an initial state s_0, can we reach a state s_n where a_i is active?»

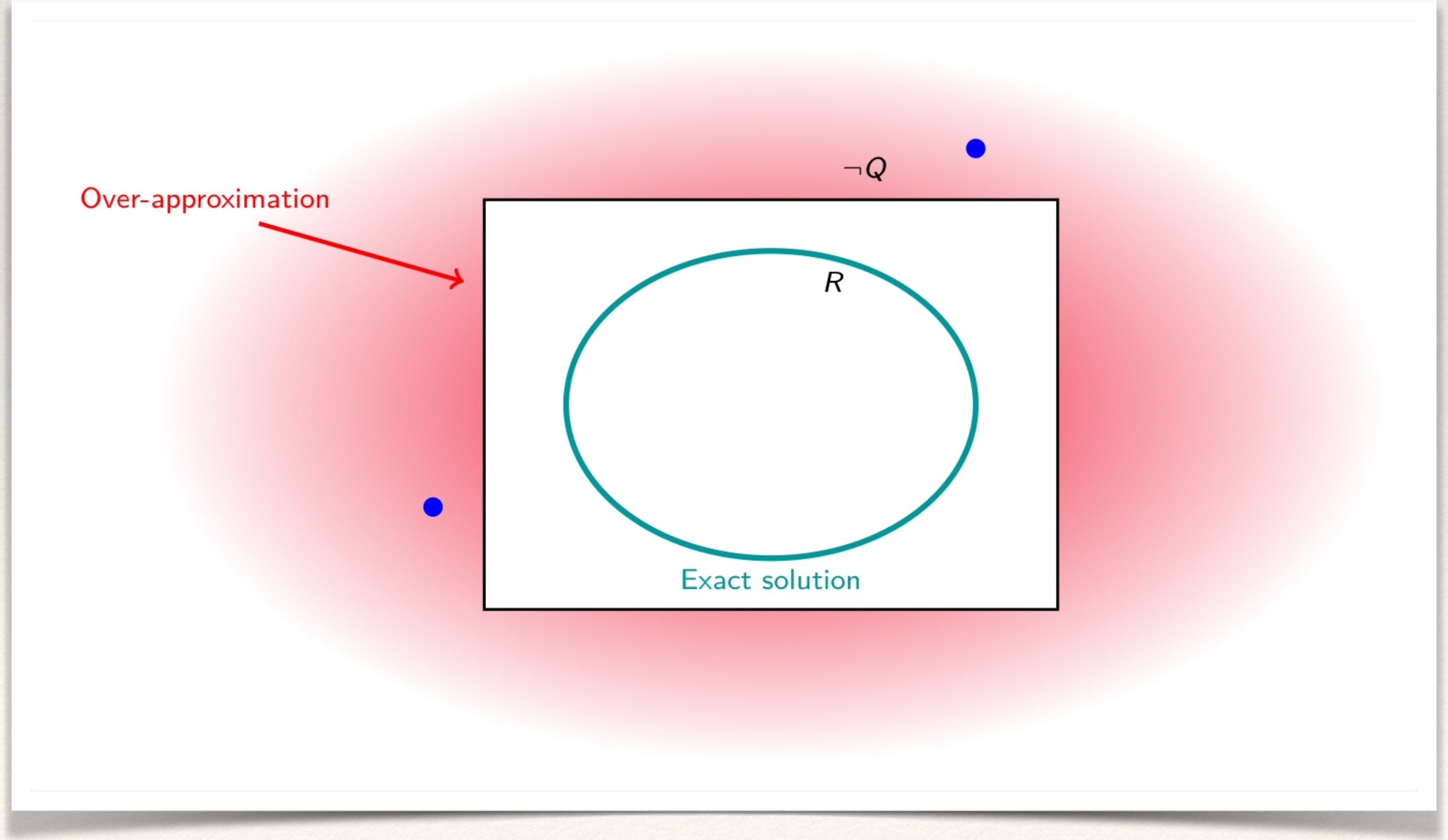
Our proposition: **under-approximation** P and **over-approximation** Q of the dynamics s.t. $P \Rightarrow R \Rightarrow Q$



Reachability analysis [PMR12]

Goal: check properties of the following type
« From an initial state s_0, can we reach a state s_n where a_i is active?»

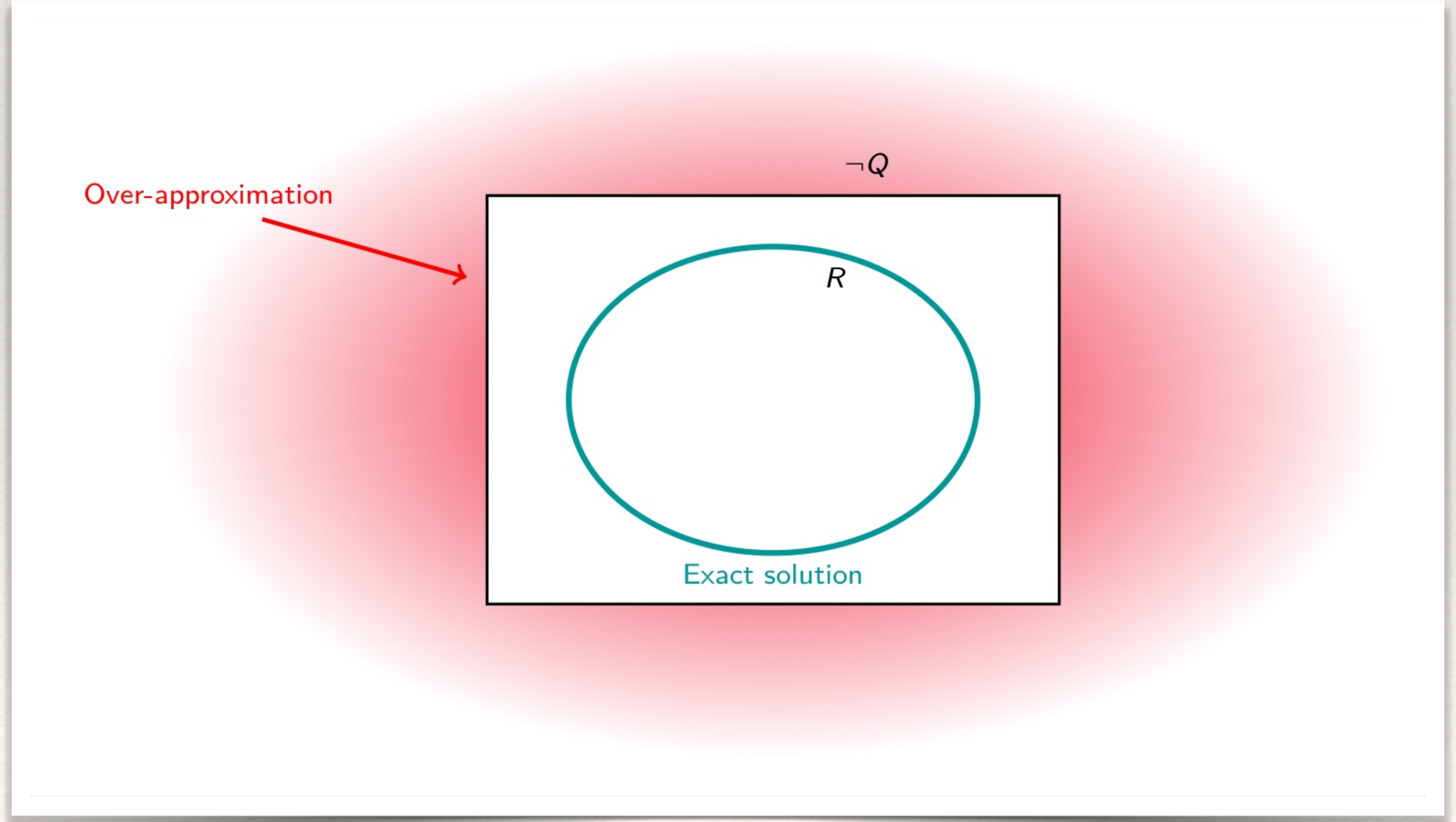
Our proposition: **under-approximation** P and **over-approximation** Q of the dynamics s.t. $P \Rightarrow R \Rightarrow Q$



Reachability analysis [PMR12]

Goal: check properties of the following type
« From an initial state s_0, can we reach a state s_n where a_i is active?»

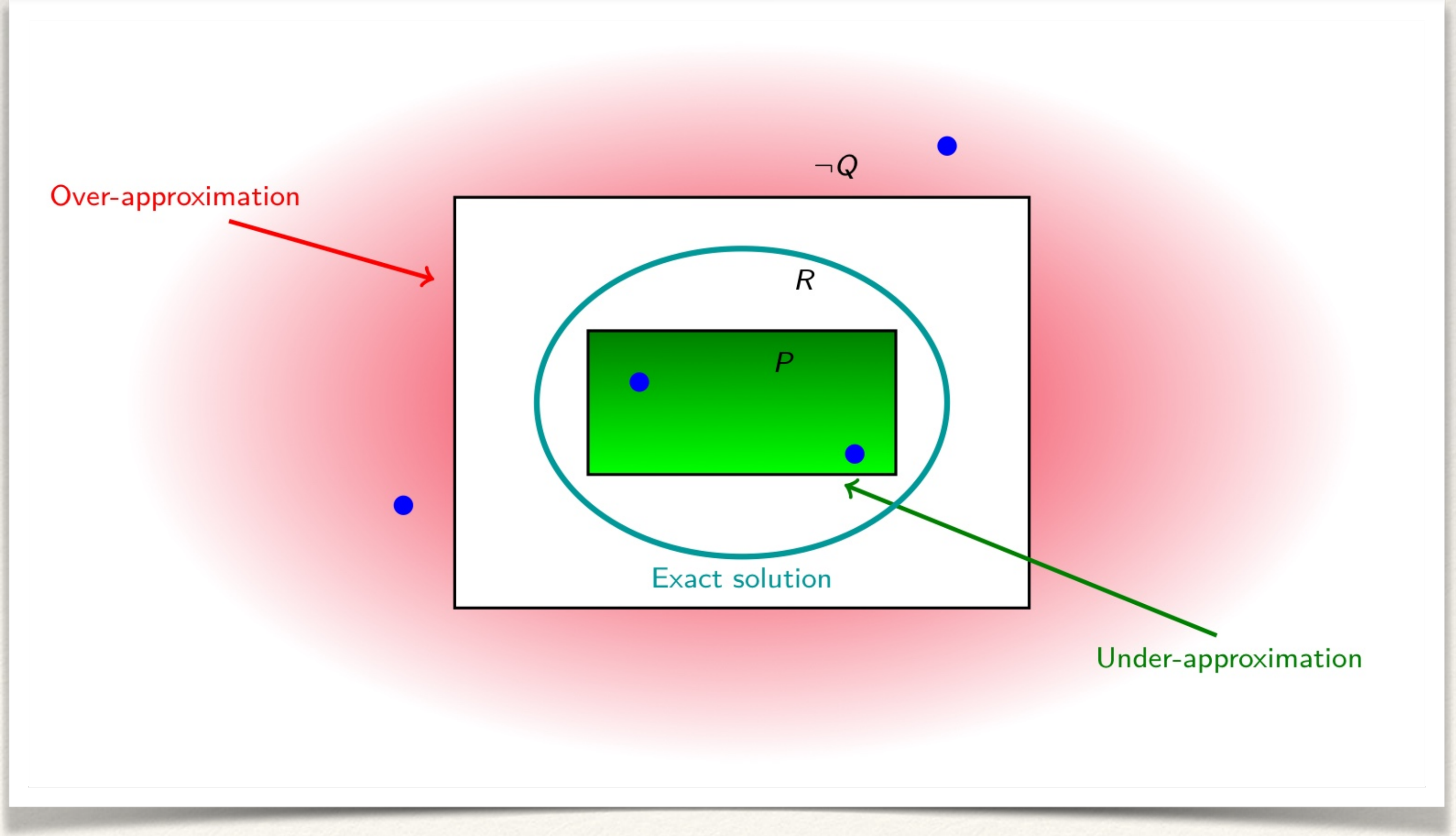
Our proposition: **under-approximation** P and **over-approximation** Q of the dynamics s.t. $P \Rightarrow R \Rightarrow Q$



Reachability analysis [PMR12]

Goal: check properties of the following type
« From an initial state s_0, can we reach a state s_n where a_i is active?»

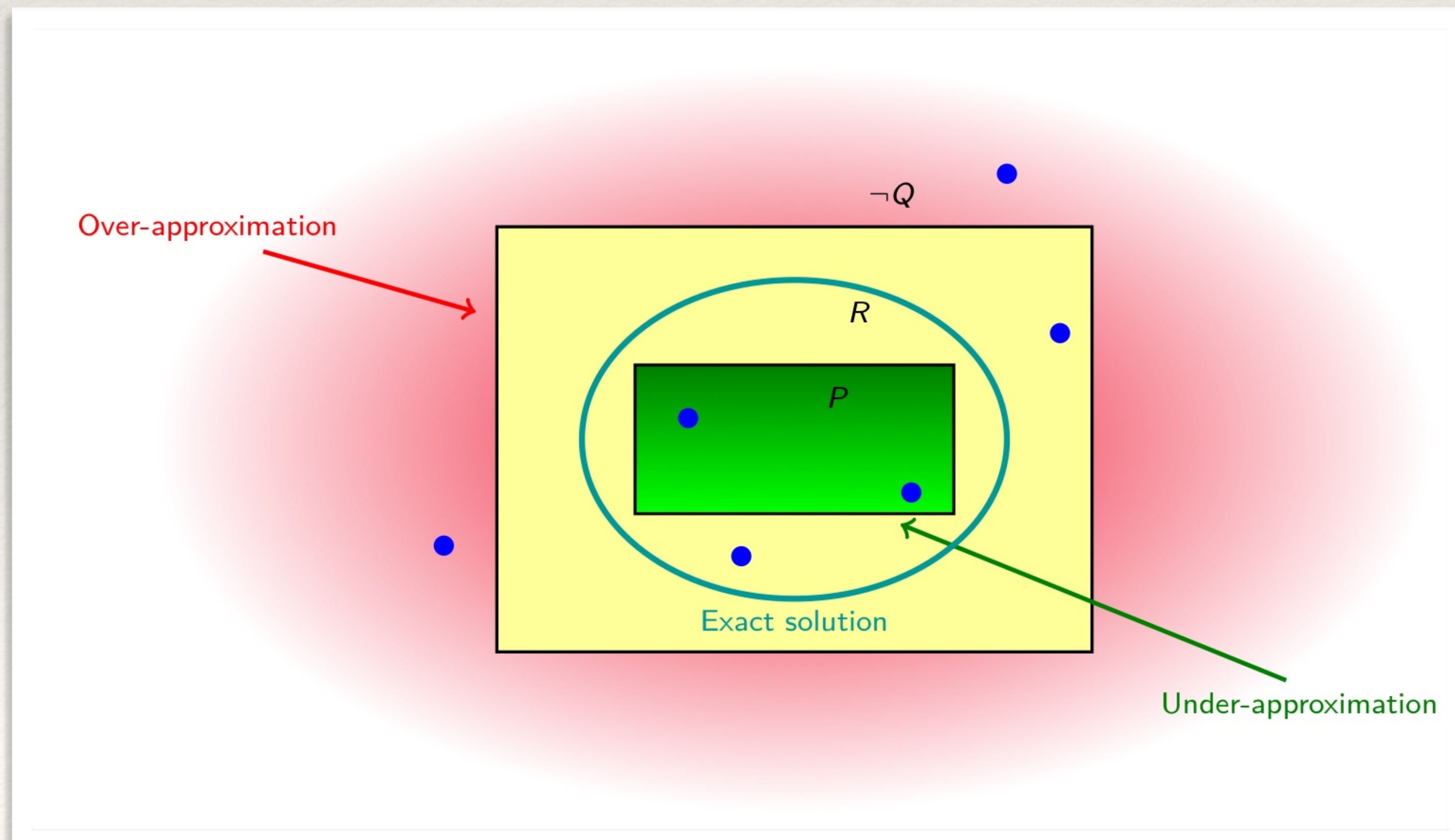
Our proposition: **under-approximation** P and **over-approximation** Q of the dynamics s.t. $P \Rightarrow R \Rightarrow Q$



Reachability analysis [PMR12]

Goal: check properties of the following type
« From an initial state s_0, can we reach a state s_n where a_i is active?»

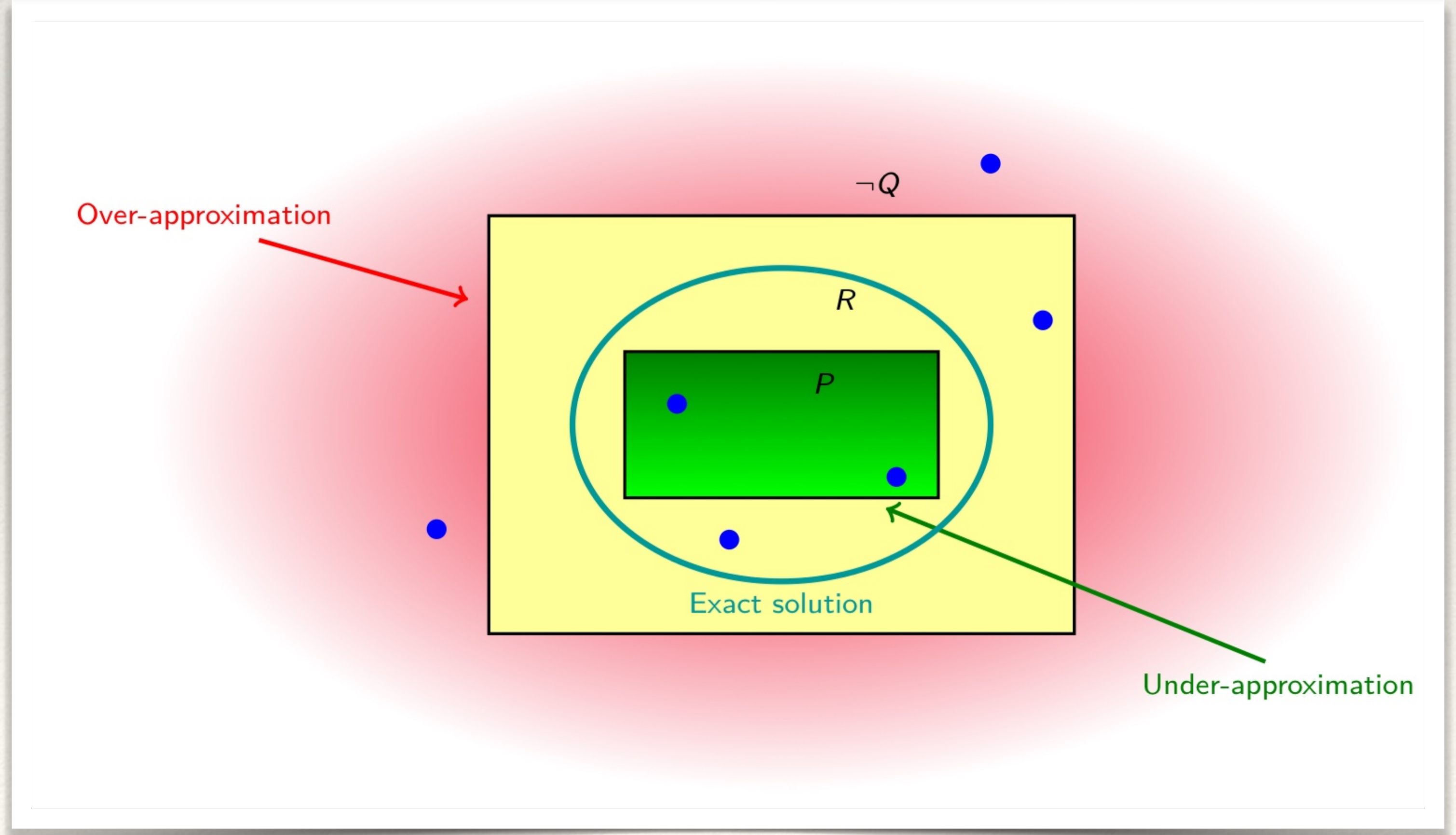
Our proposition: **under-approximation** P and **over-approximation** Q of the dynamics s.t. $P \Rightarrow R \Rightarrow Q$



Reachability analysis [PMR12]

Goal: check properties of the following type
« From an initial state s_0, can we reach a state s_n where a_i is active?»

Our proposition: **under-approximation** P and **over-approximation** Q of the dynamics s.t. $P \Rightarrow R \Rightarrow Q$



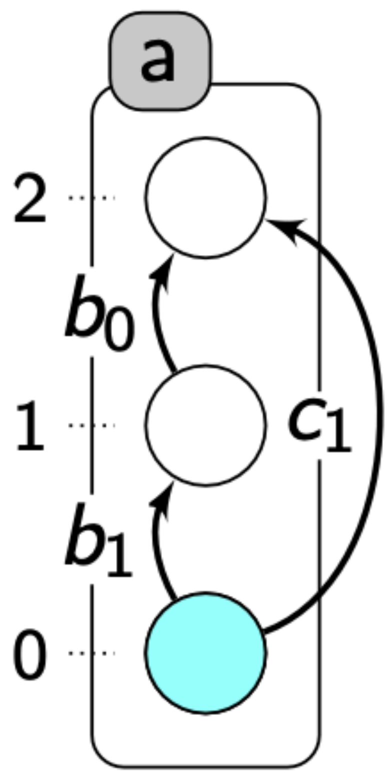
Polynomial in the number of automata

Exponential in the number of local states of each automaton (usually very low, max. 4)

Reachability analysis [PMR12]

- Idea: resolve the reachability of a local state of automata

Example : reachability from a_0 to a_2 : (objective $a_0 \leadsto a_2$)



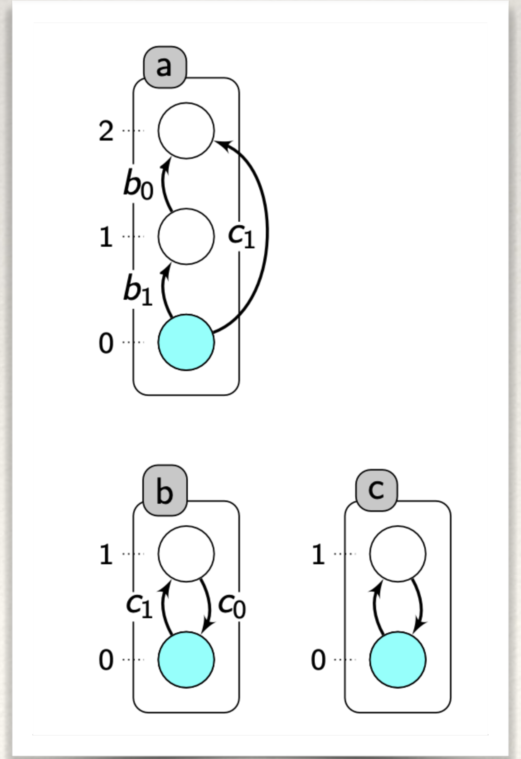
- 2 local solutions :
  - $a_0 \xrightarrow{c_1} a_2$ (c_1 required)
  - $a_0 \xrightarrow{b_1} a_1$; $a_1 \xrightarrow{b_0} a_2$ (b_0 and b_1 required)

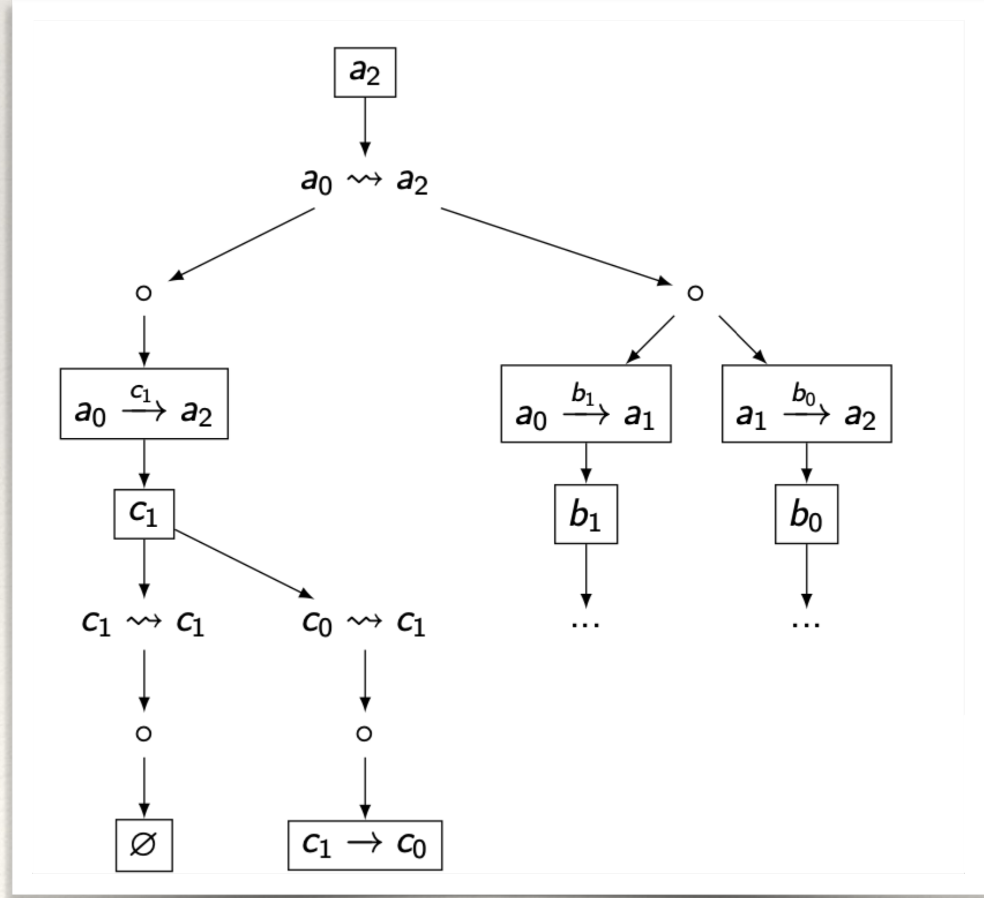
Reachability analysis [PMR12]

- Idea: resolve the reachability of a local state of automata
- **Over-approximation**: **abstraction of the order** in which the required processes are necessary
- **Under-approximation**: all required processes must be activated in all possible orders

Reachability analysis [PMR12]

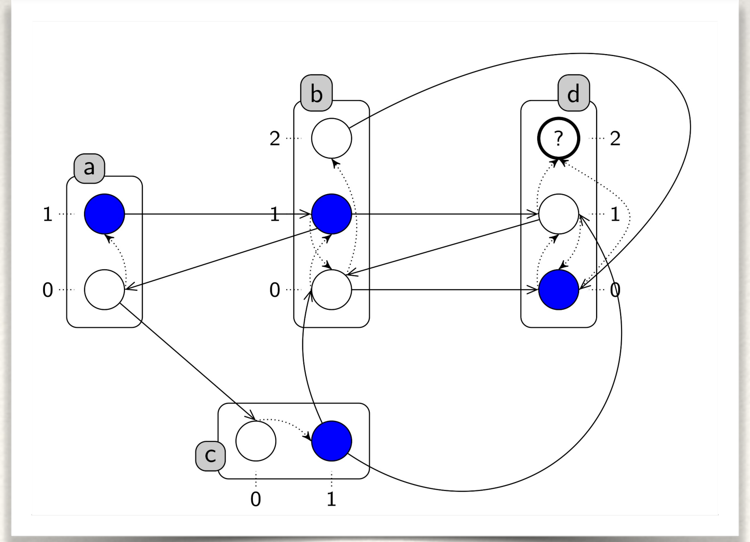
- Idea: resolve the reachability of a local state of automata
- **Over-approximation**: **abstraction of the order** in which the required processes are necessary
- **Under-approximation**: all required processes must be activated in all possible orders
- Construction of **local causality graph**





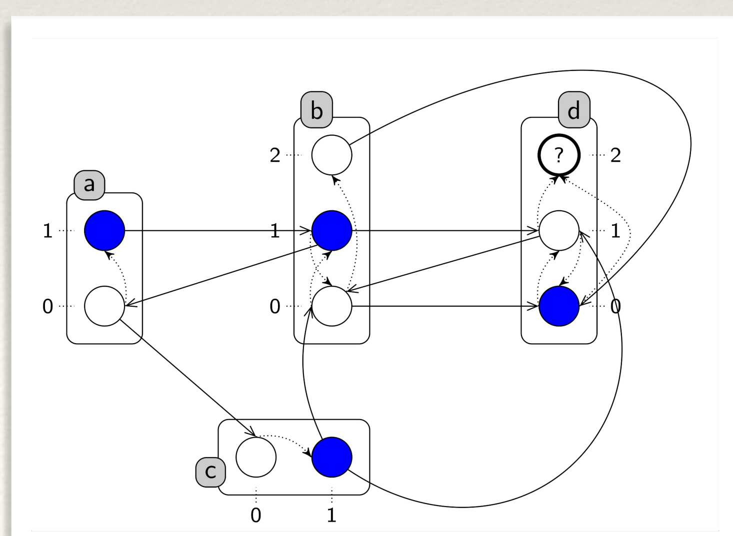
Implementation of Local Causality Graph [PMR12]

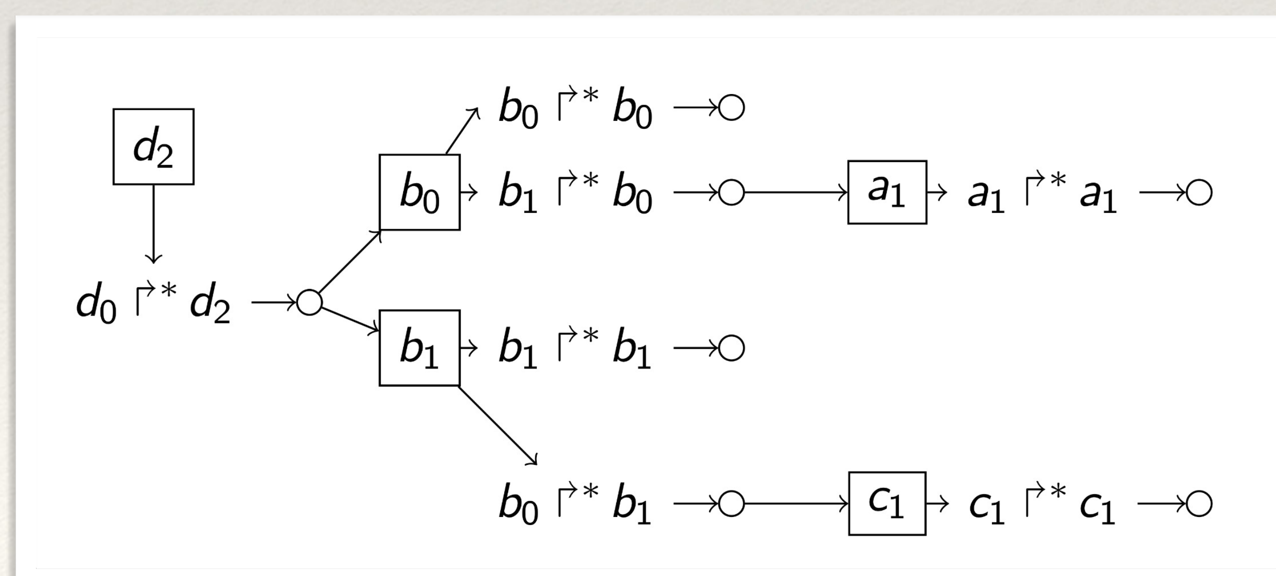
- **Computation** of the local causality graph:
  - **Polynomial** in the number of automata
  - **Exponential** in the number of local states of each automaton (usually low, max. 4)
- **Analysis** of the graph (sufficient condition): **polynomial** in the size of the abstract graph
- **Enumeration** of the subsets of solutions (if needed): **exponential** in the size of the abstract graph



Implementation of Local Causality Graph [PMR12]

- **Computation** of the local causality graph:
  - **Polynomial** in the number of automata
  - **Exponential** in the number of local states of each automaton (usually low, max. 4)
- **Analysis** of the graph (sufficient condition): **polynomial** in the size of the abstract graph
- **Enumeration** of the subsets of solutions (if needed): **exponential** in the size of the abstract graph



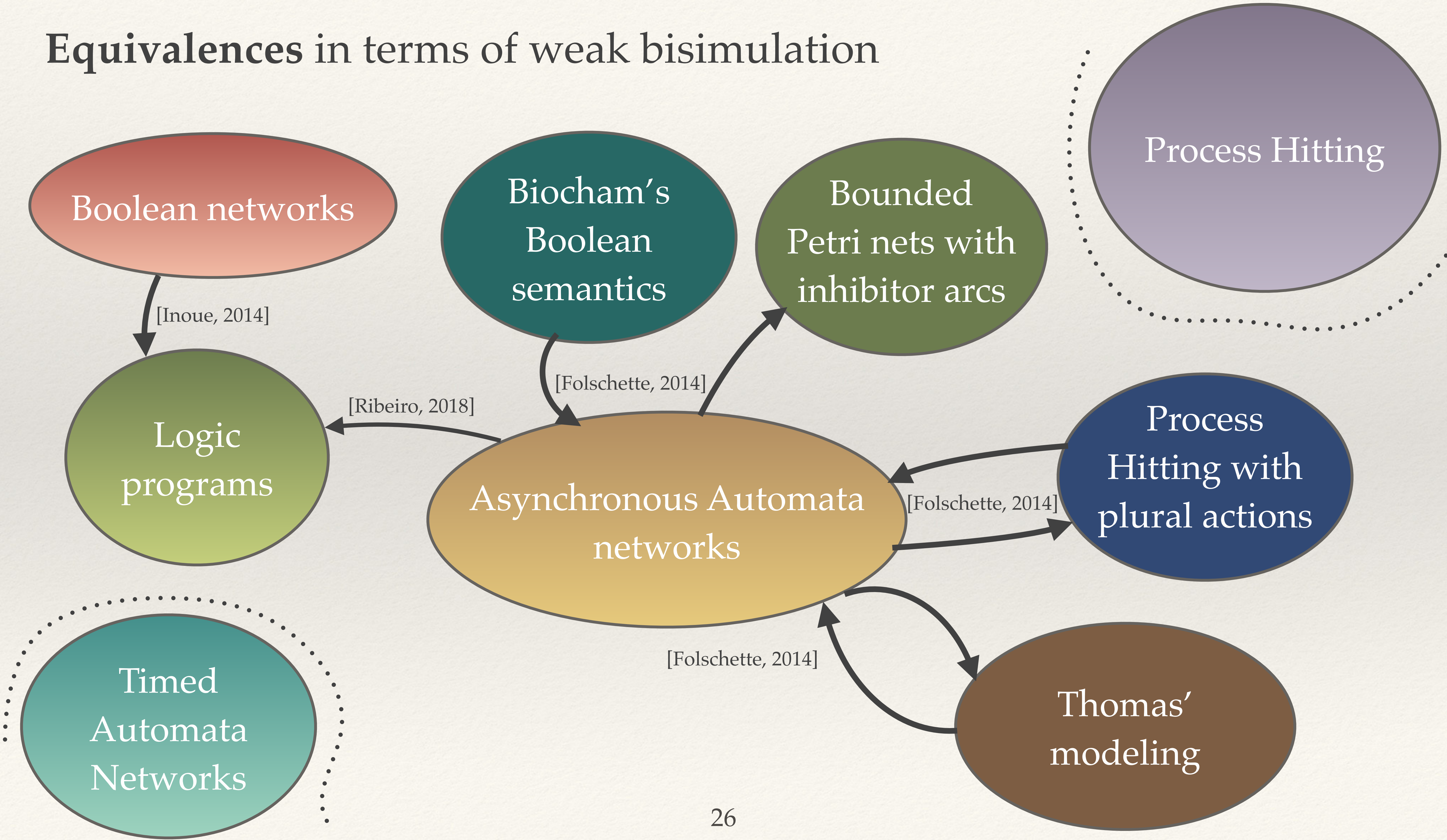


Overview

- Modeling frameworks for dynamical analysis
- Analysis of formal models
- **Inference and Learning approaches**
- Application to biological case studies

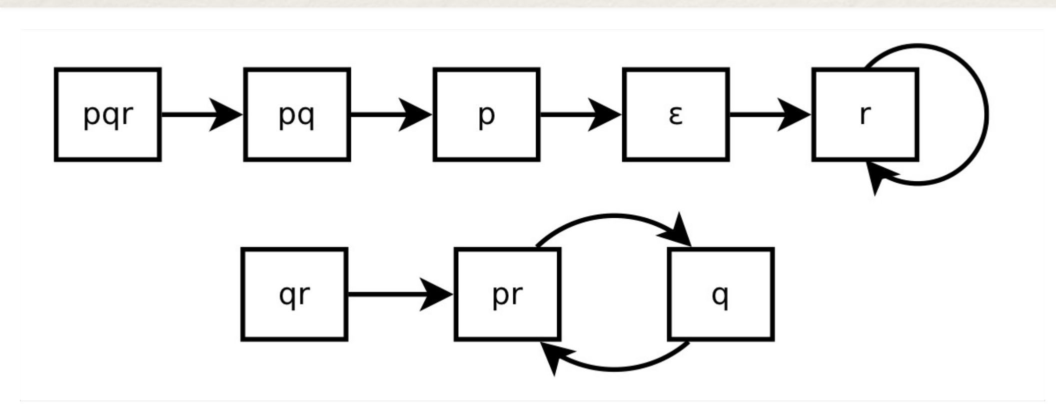
Correspondance between formalisms

Equivalences in terms of weak bisimulation



Learn systems dynamics

Principle of the LFIT framework [IRS14]: learn a **logic program** by observing the behavior of the system expressed as a succession **states-transitions**.



LFIT

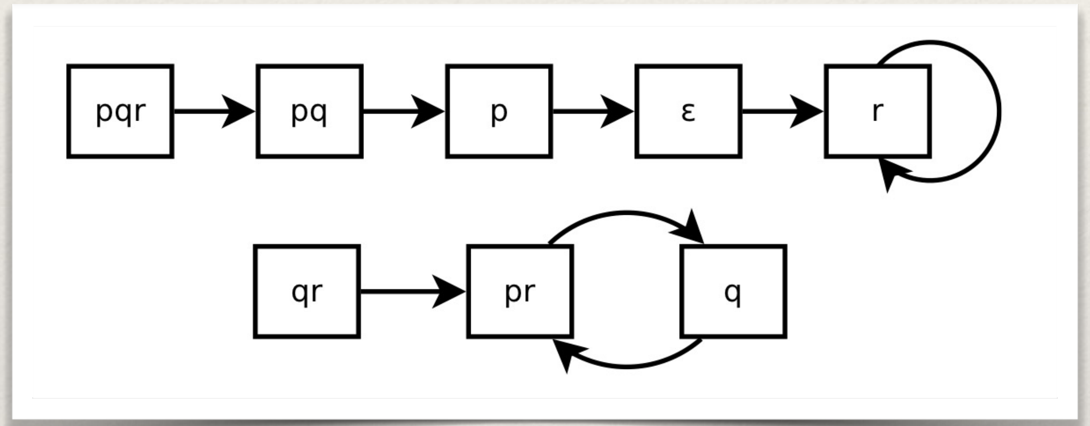
$$p(t+1) \leftarrow q(t).$$
$$q(t+1) \leftarrow p(t) \land r(t).$$
$$r(t+1) \leftarrow \neg p(t).$$

Input: observation of the behavior of the system

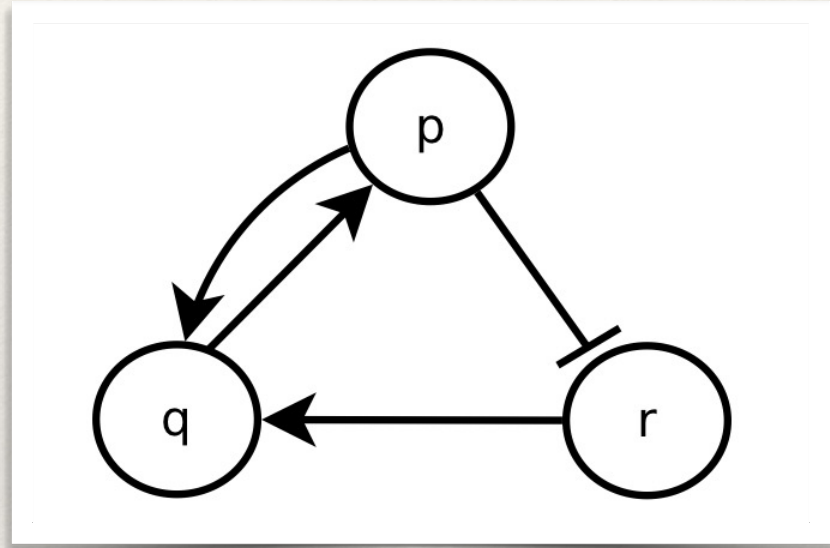
Output: logic program

Learn systems dynamics

Principle of the LFIT framework [IRS14]: learn a **logic program** by observing the behavior of the system expressed as a succession **states-transitions**.



LFIT

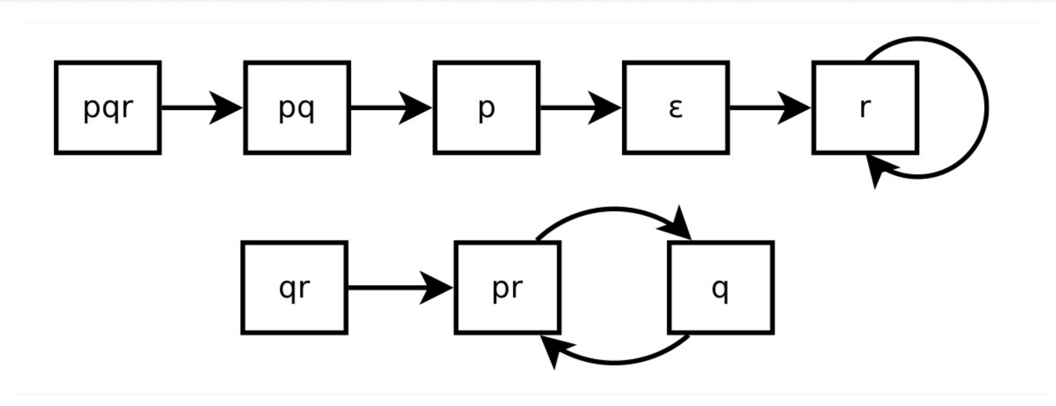


Input: observation of the behavior of the system

Output: Boolean network

Learn systems dynamics

Principle of the LFIT framework [IRS14]: learn a **logic program** by observing the behavior of the system expressed as a succession **states-transitions**.



LFIT

$$p(t+1) \leftarrow q(t).$$
$$q(t+1) \leftarrow p(t) \wedge r(t).$$
$$r(t+1) \leftarrow \neg p(t).$$

Input: observation of the behavior of the system

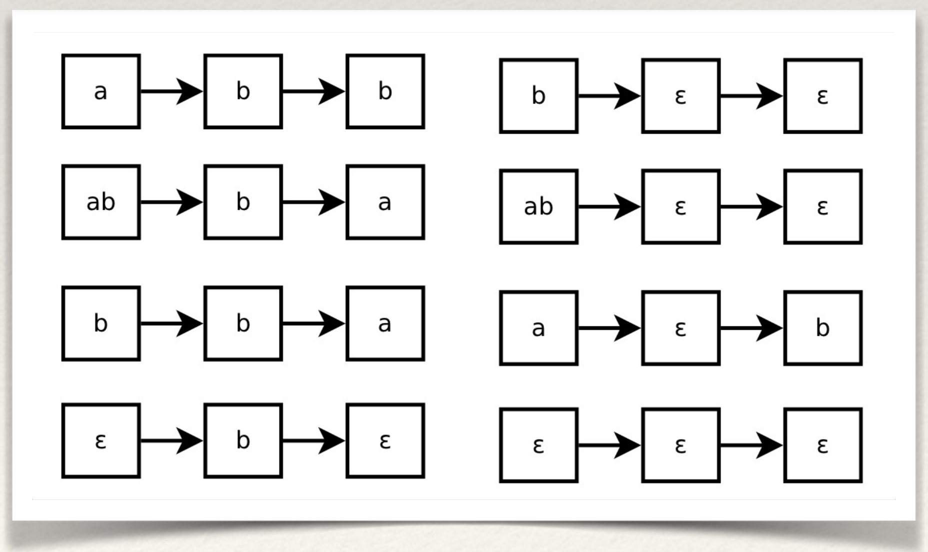
Output: logic program

Boolean variables

Synchronous and **deterministic** dynamics

Apply LFIT to biological systems
[RMI+15a,RMI+15b]

- Successive extensions :
  - **Delayed** influences (models with memory)
  - **Multi-valued** variables
  - **Asynchronous** dynamics
  - **Non-deterministic** semantics



LFkT

a(t) ← b(t−1), b(t−2)
b(t) ← a(t−2), ¬b(t−2)

Input: observation of the system's behavior

Output: logic program

A wide range of possible models (and inference algorithms)

https://github.com/Tony-sama/pylfit

Values	Boolean	Multi-valued	Captured symbolically (e.g. intervals)
Time	Chronological: Event-driven model	Chronometrical: Discrete time	Chronometrical: Dense time
Semantics of discrete transitions	Synchronous	Asynchronous	Generalized
Determinism of discrete transitions	Deterministic	Non-deterministic	Probabilistic
Markov property	With memory	Memoryless	

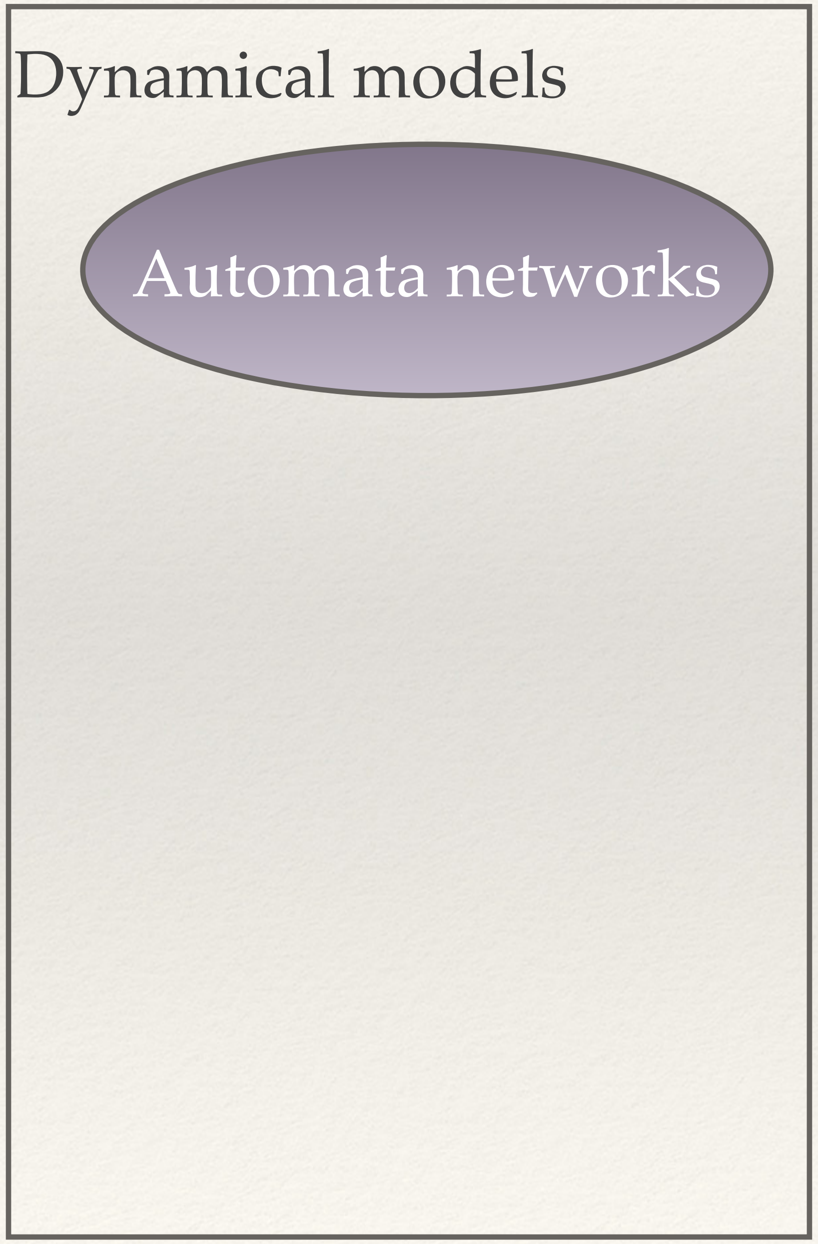
Overview

- Modeling frameworks for dynamical analysis
- Analysis of formal models
- Inference and Learning approaches
- **Application to biological case studies**

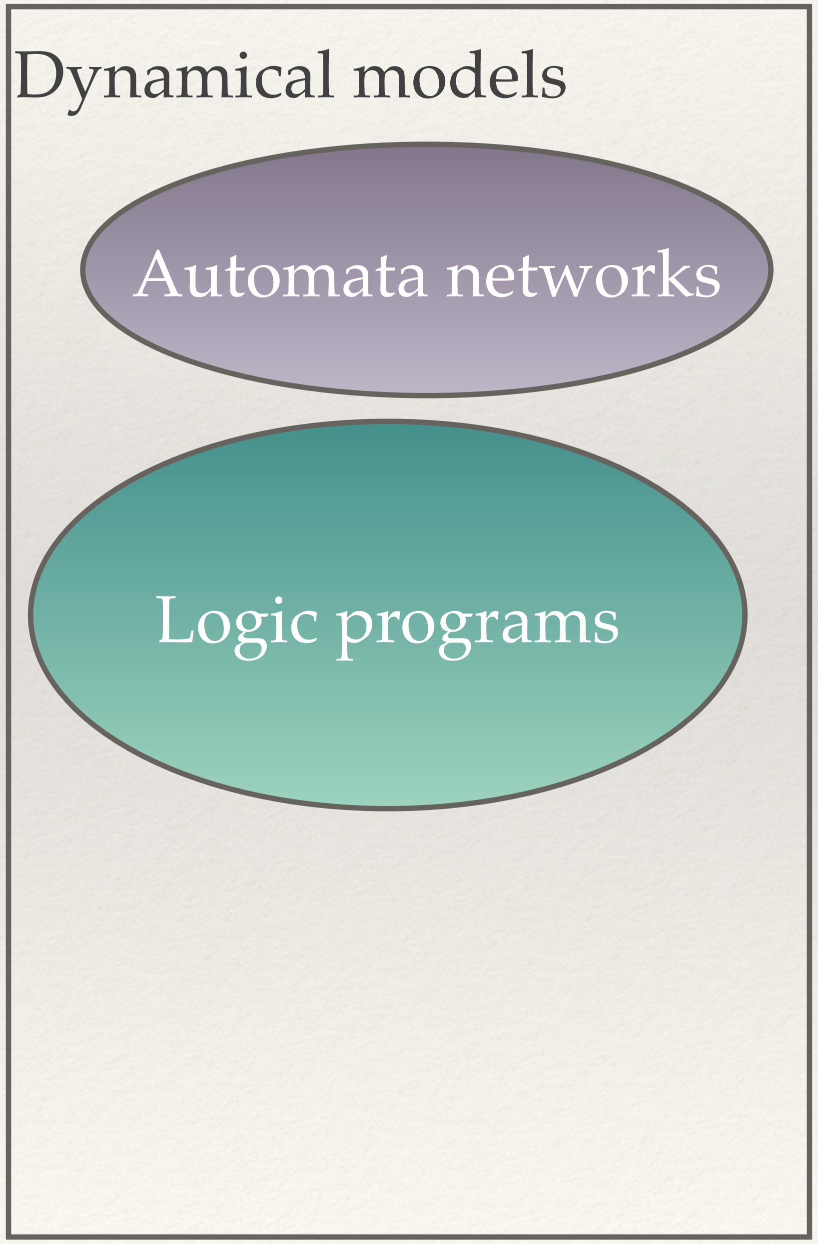
Synthesis of our contributions

Dynamical models

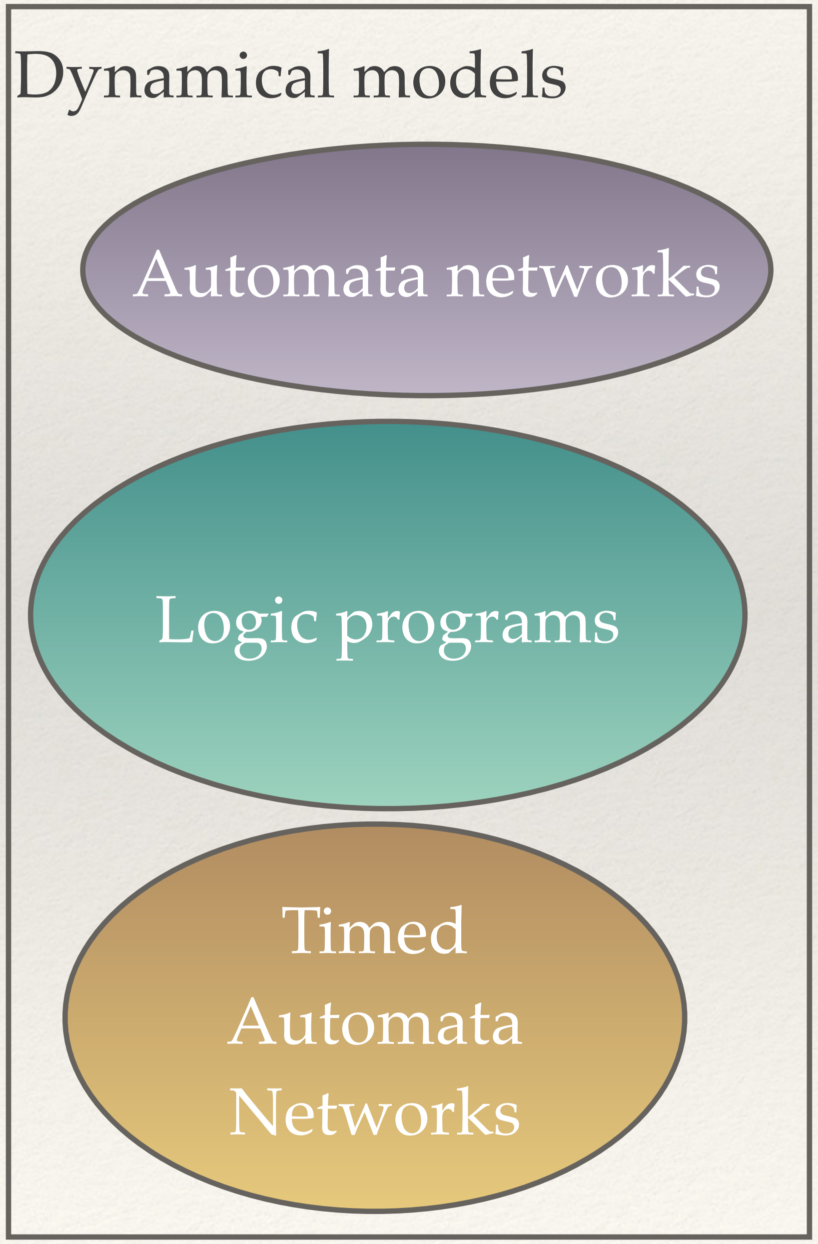
Synthesis of our contributions



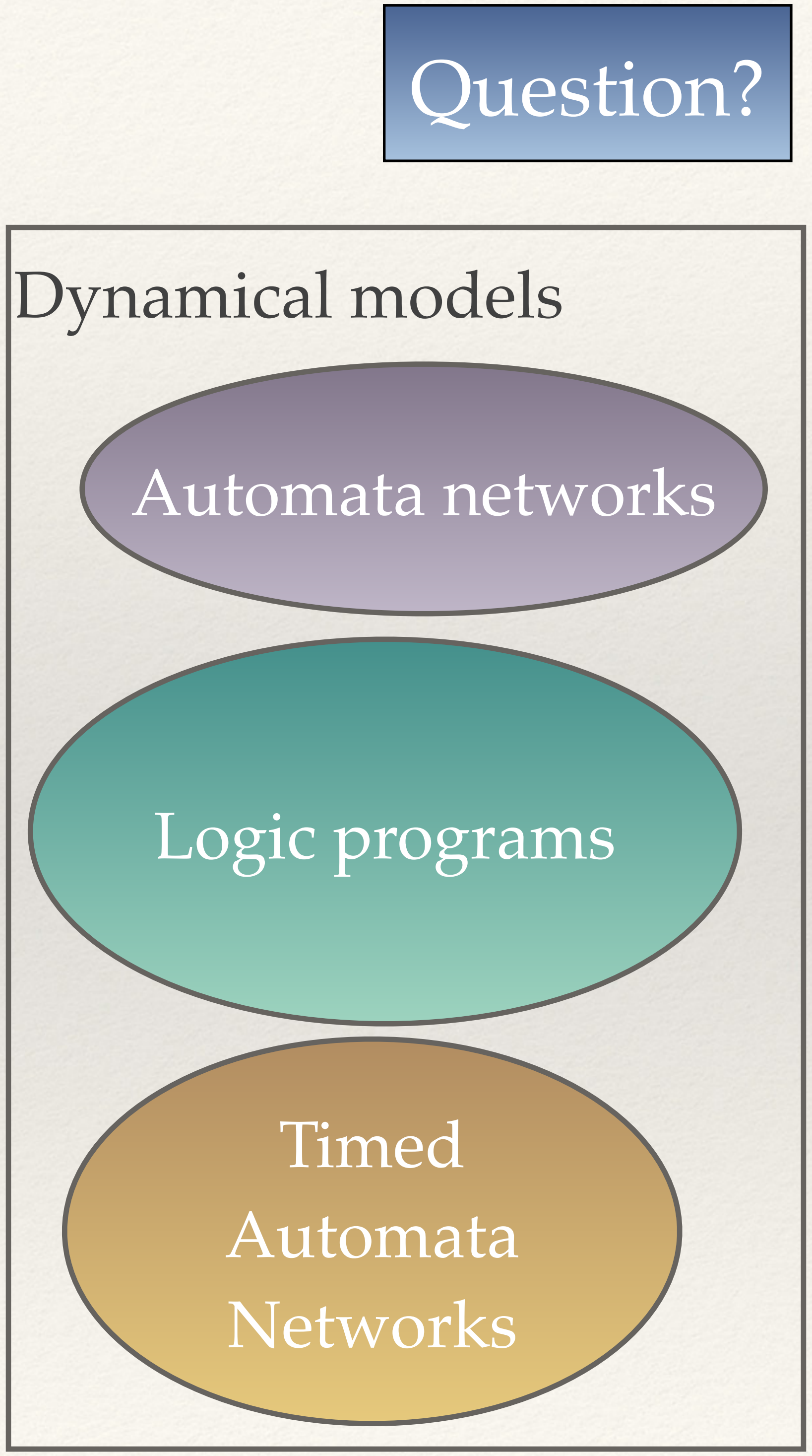
Synthesis of our contributions



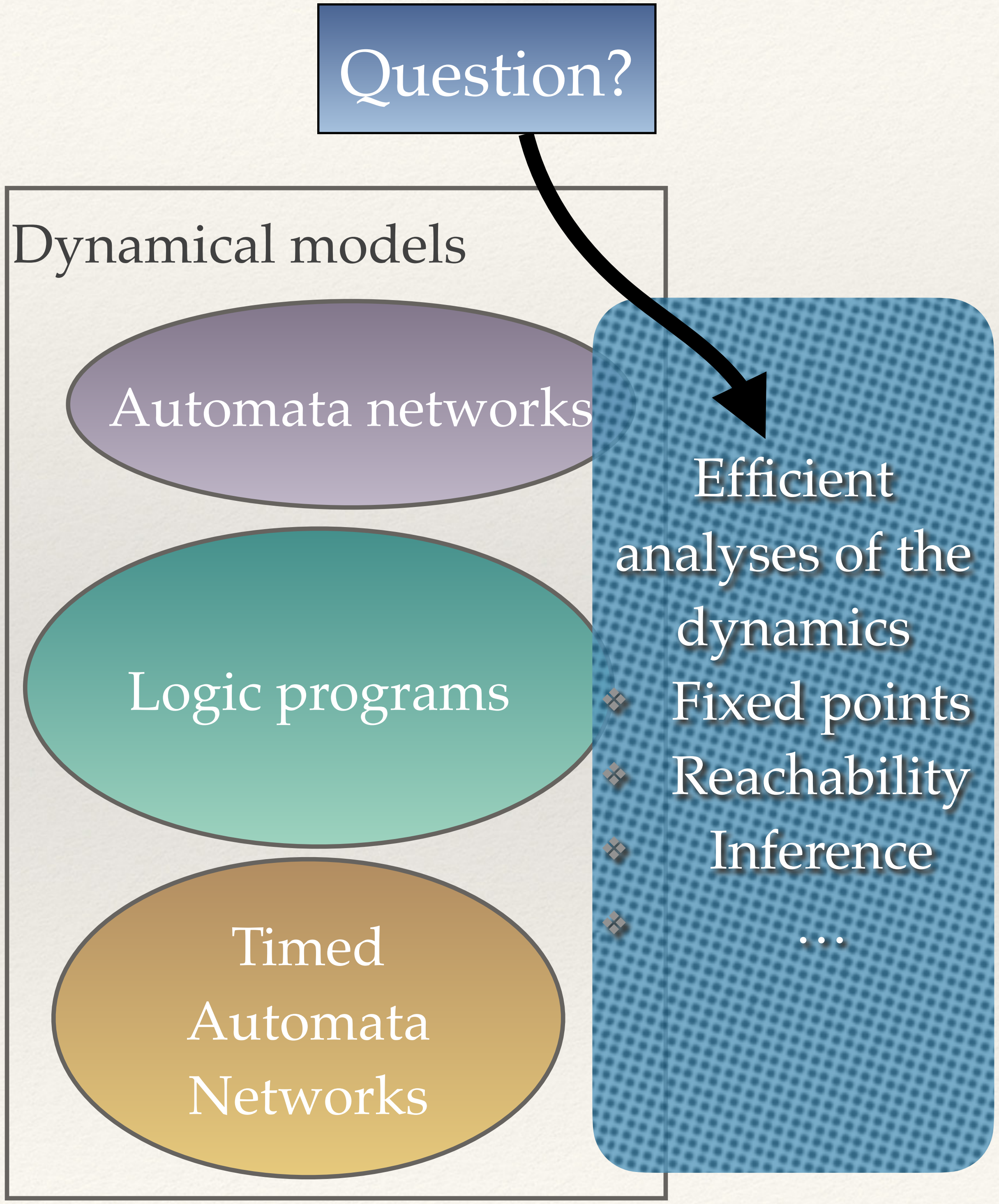
Synthesis of our contributions



Synthesis of our contributions



Synthesis of our contributions



Synthesis of our contributions

Question?

Dynamical models

Automata networks

Logic programs

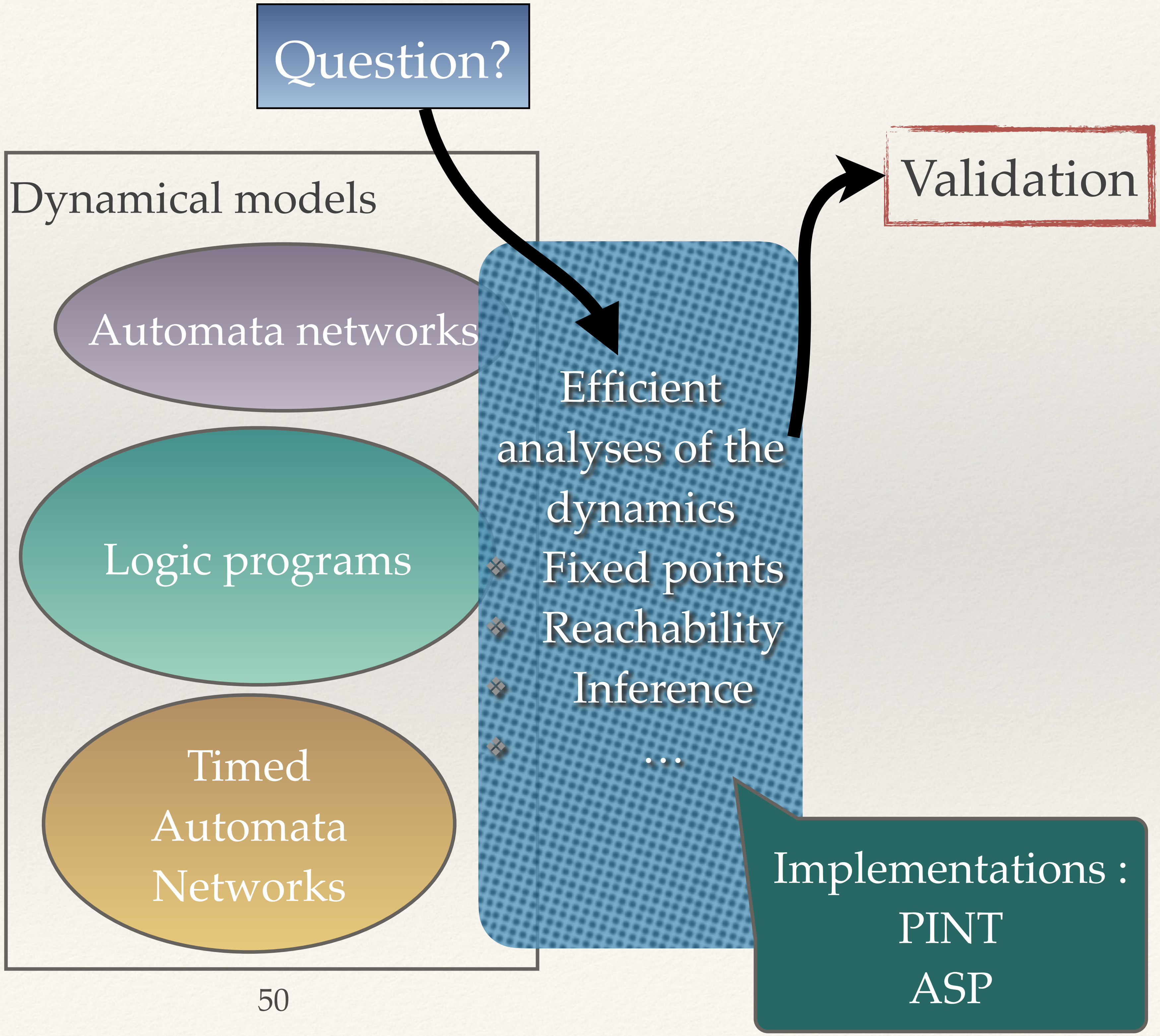
Timed Automata Networks

Efficient analyses of the dynamics

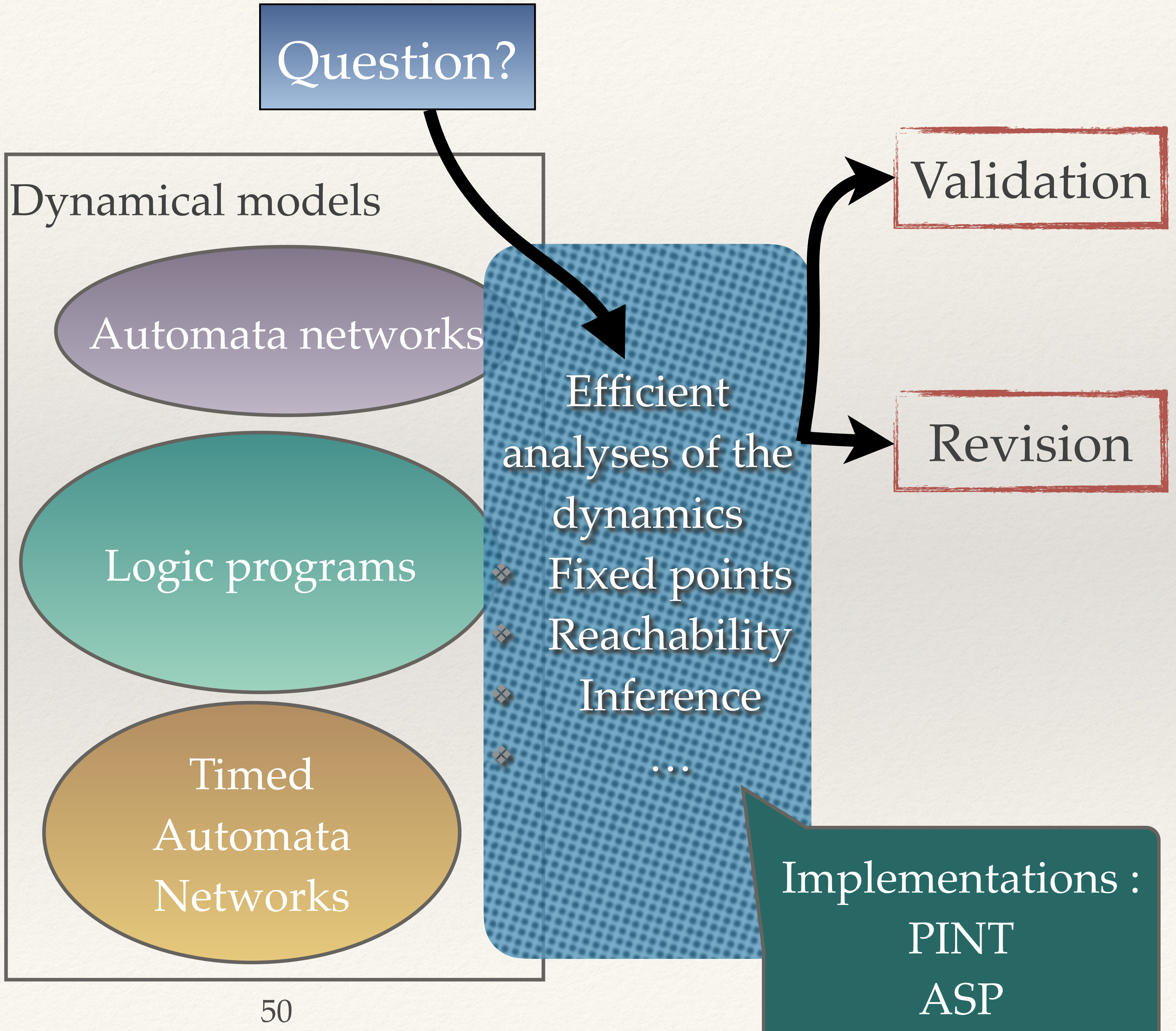
- Fixed points
- Reachability
- Inference
- …

Implementations :
PINT
ASP

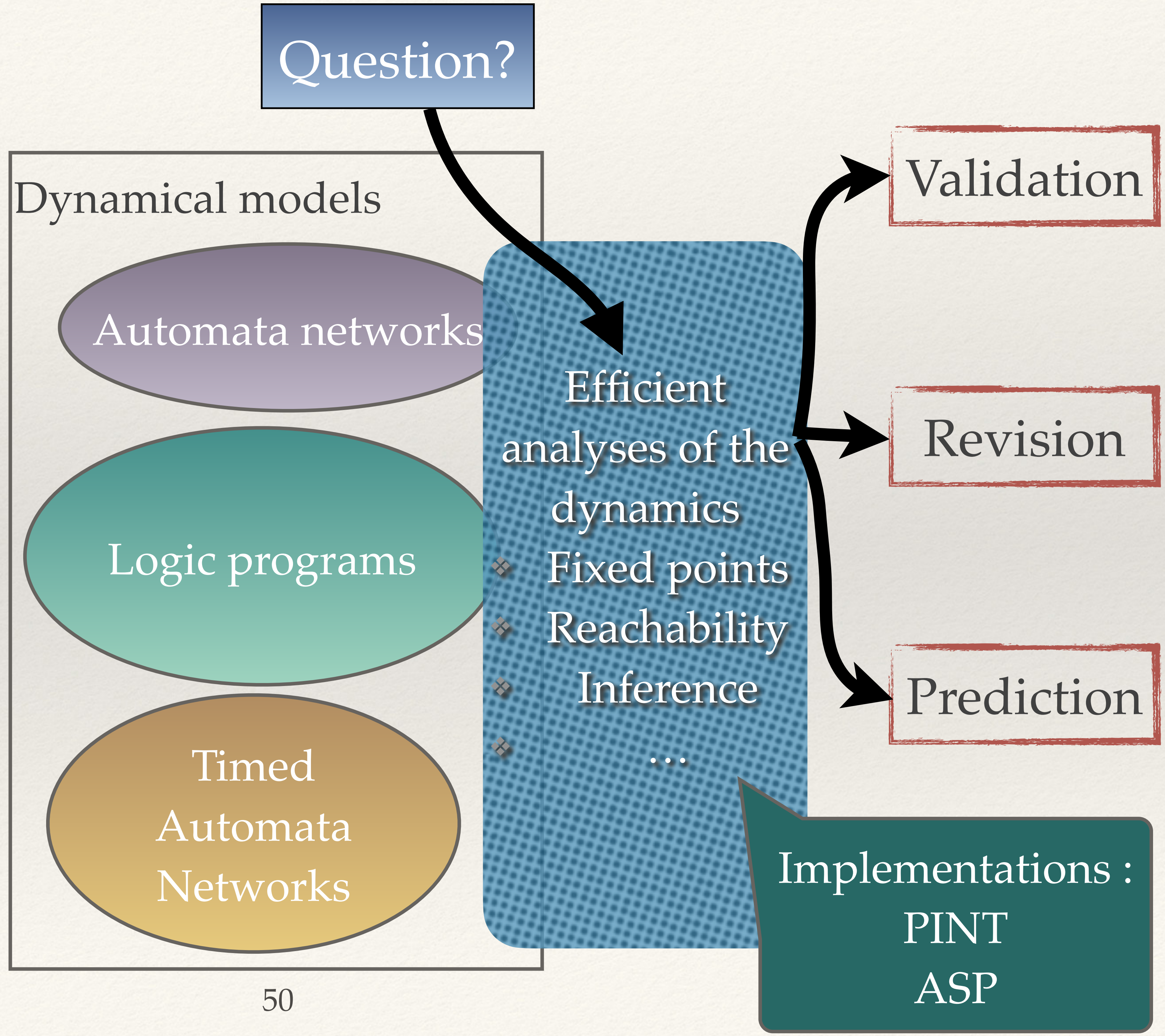
Synthesis of our contributions



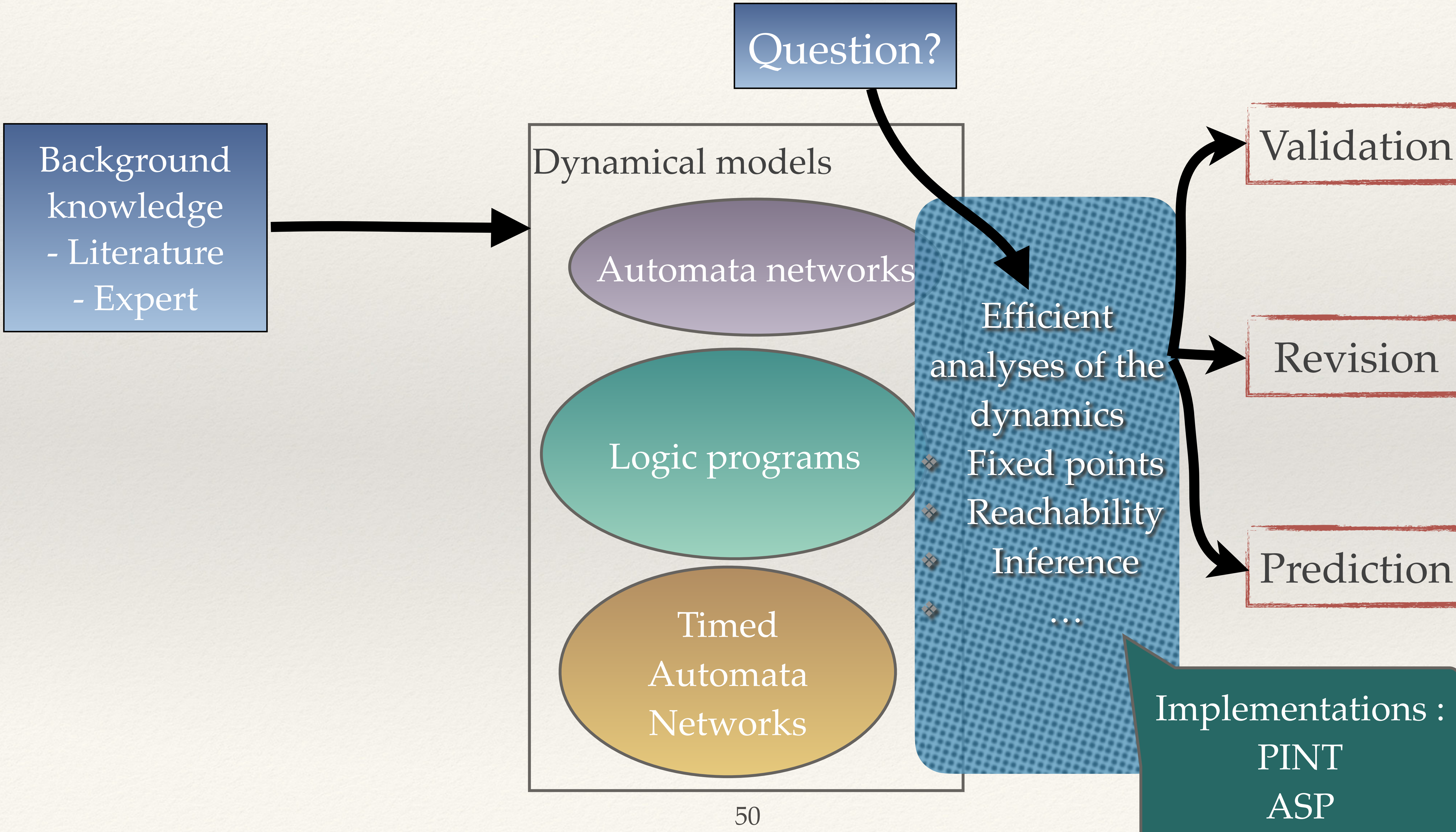
Synthesis of our contributions



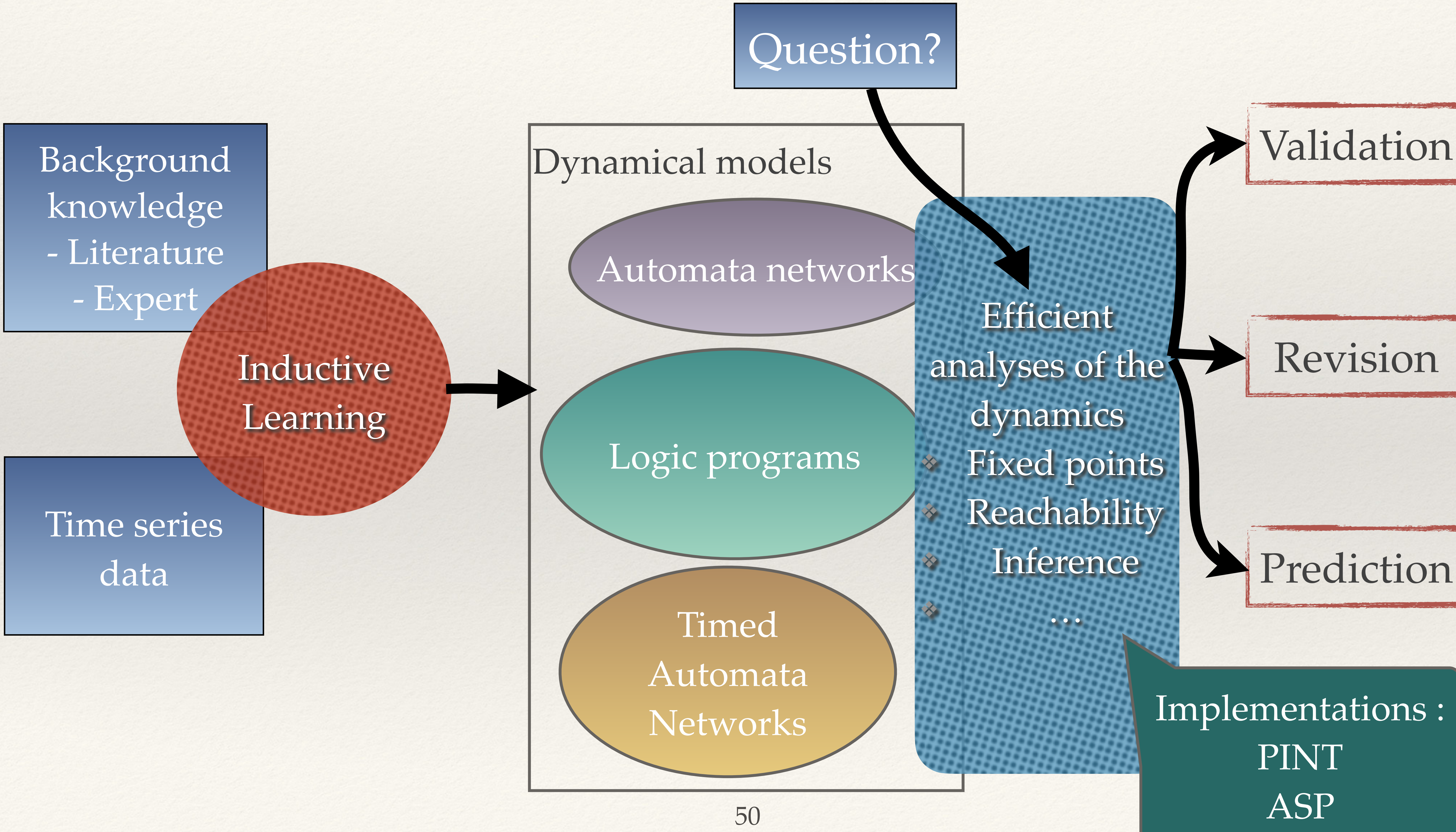
Synthesis of our contributions



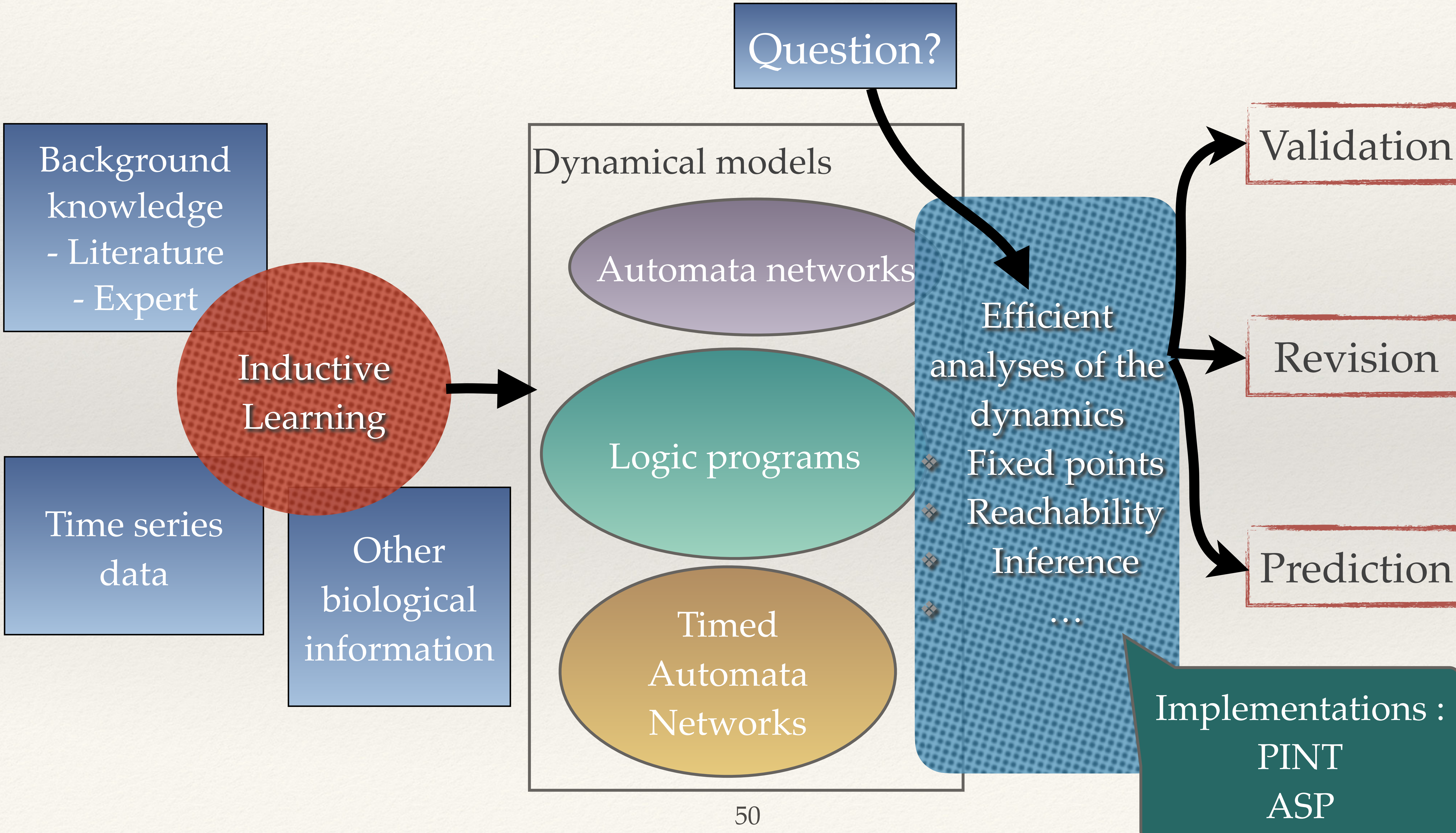
Synthesis of our contributions



Synthesis of our contributions



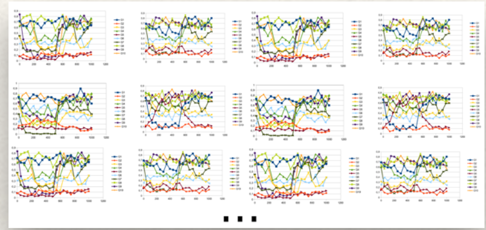
Synthesis of our contributions



Methodological goals

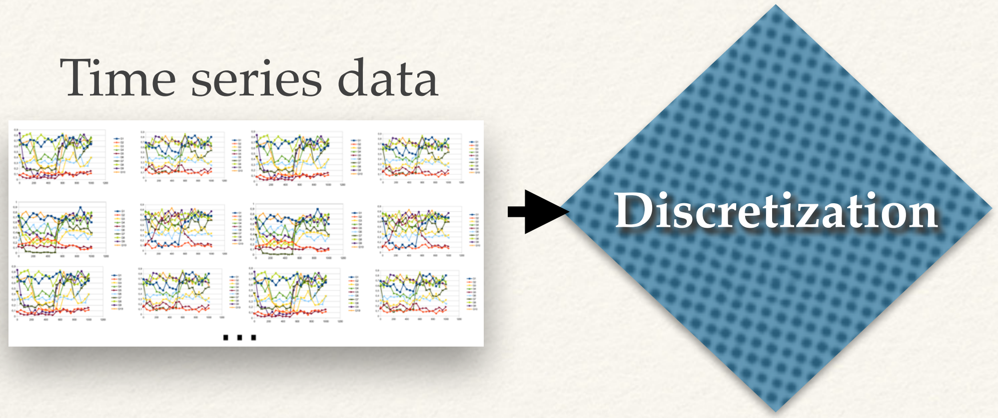
Methodological goals

Time series data



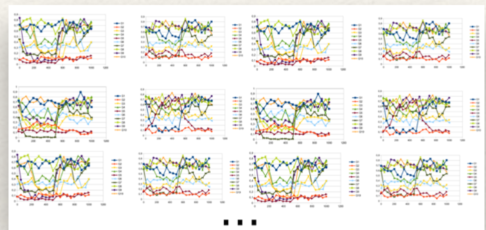
Methodological goals

Time series data



Methodological goals

Time series data

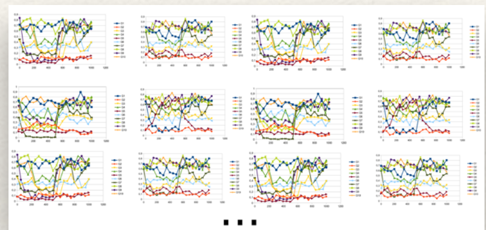


Discretization

Learning

Methodological goals

Time series data



...

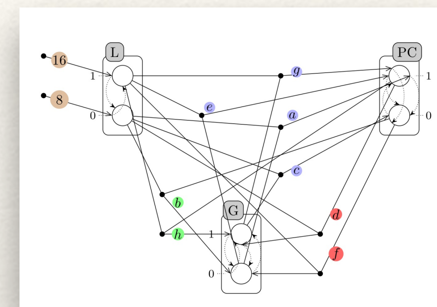
Discretization

Learning

Dynamical models

```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

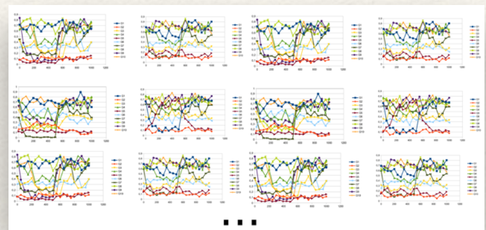
Logic programs



Automata networks

Methodological goals

Time series data



...

Discretization

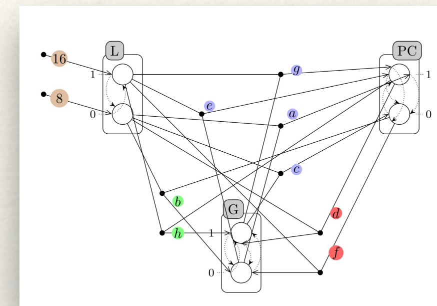
Learning

Goal n°1: improve learning by addressing noisy, incomplete and/or conflicting data

Dynamical models

```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

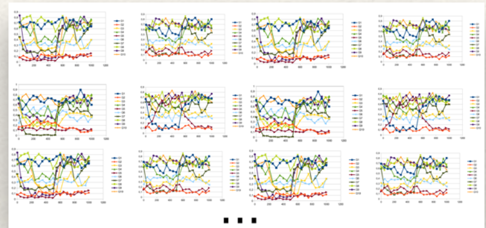
Logic programs



Automata networks

Methodological goals

Time series data



Discretization

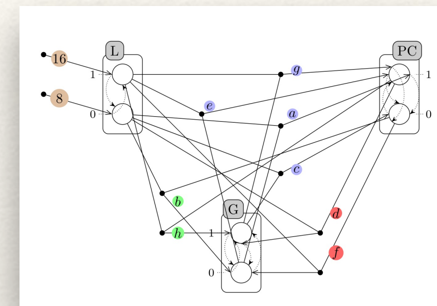
Learning

Goal n°1: improve learning by addressing noisy, incomplete and/or conflicting data

Dynamical models

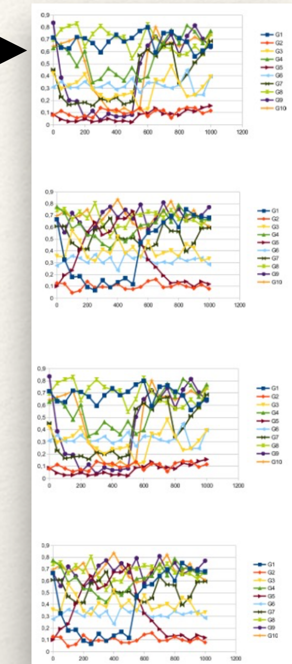
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

Logic programs



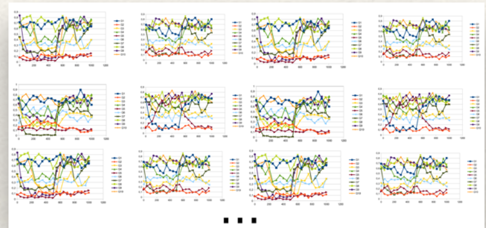
Automata networks

Analysis of dynamic behaviors



Methodological goals

Time series data



Discretization

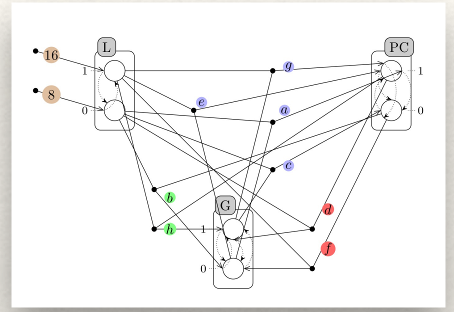
Learning

Goal n°1: improve learning by addressing noisy, incomplete and/or conflicting data

Dynamical models

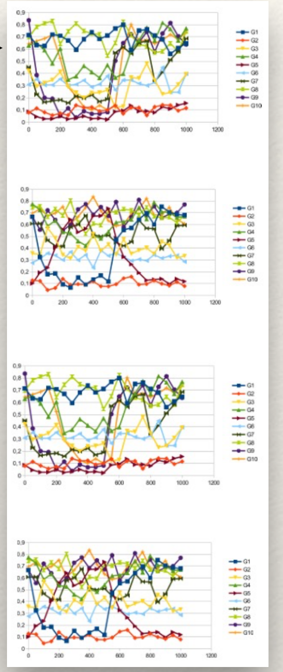
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

Logic programs



Automata networks

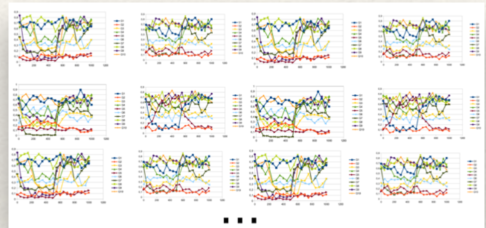
Analysis of dynamic behaviors



Background knowledge on the dynamics

Methodological goals

Time series data



...

Discretization

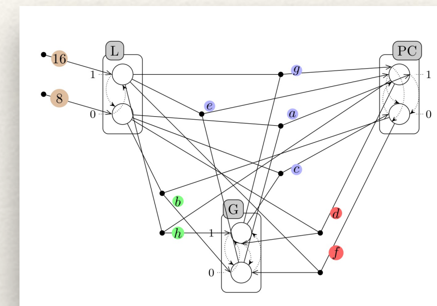
Learning

Goal n°1: improve learning by addressing noisy, incomplete and/or conflicting data

Dynamical models

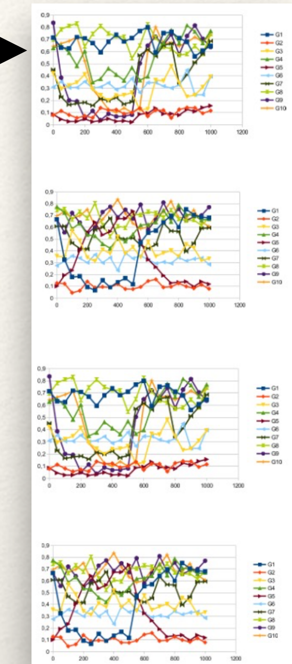
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

Logic programs



Automata networks

Analysis of dynamic behaviors



Background knowledge on the dynamics

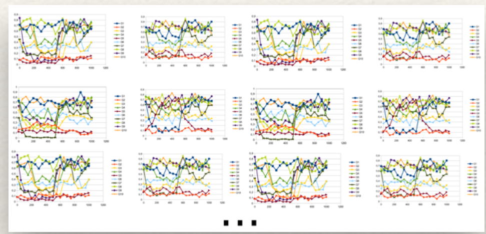
Formalization

Expected properties in temporal logics

$$(M(p_{G0}) = 1 \wedge M(p_{PC0}) = 1) \rightsquigarrow_{[0,\tau_{0,1}]} (M(p_{G1}) = 1)$$

Methodological goals

Time series data



...

Discretization

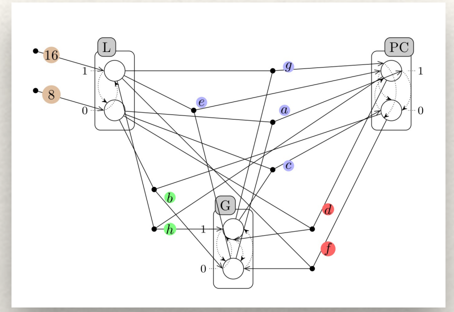
Learning

Goal n°1: improve learning by addressing noisy, incomplete and/or conflicting data

Dynamical models

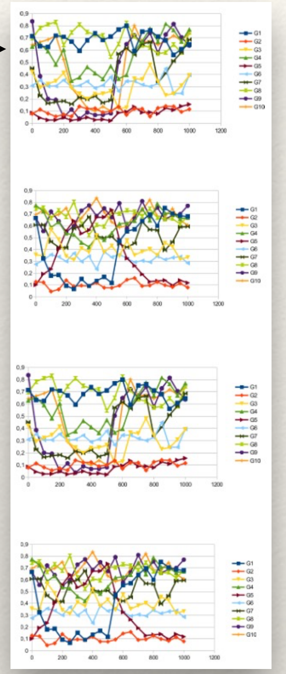
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

Logic programs



Automata networks

Analysis of dynamic behaviors



Background knowledge on the dynamics

Formalization

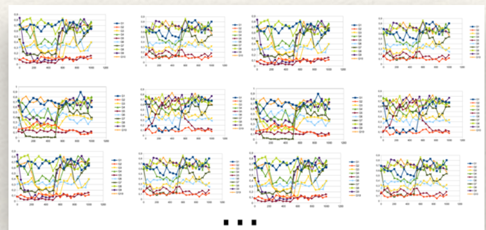
Expected properties in temporal logics

$(M(p_{G0}) = 1 \wedge M(p_{PC0}) = 1) \rightsquigarrow_{[0,\tau_{0,1}]} (M(p_{G1}) = 1)$

Check properties

Methodological goals

Time series data



Discretization

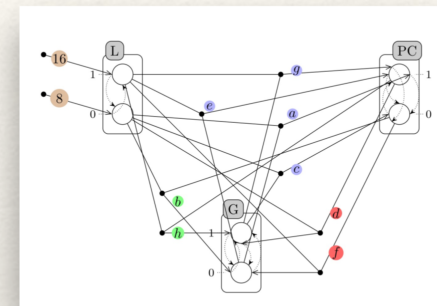
Learning

Goal n°1: improve learning by addressing noisy, incomplete and/or conflicting data

Dynamical models

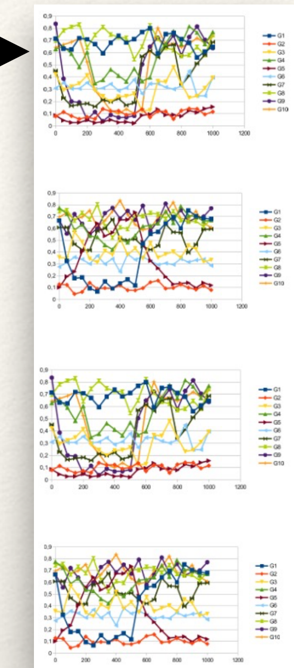
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

Logic programs



Automata networks

Analysis of dynamic behaviors



If properties are not verified on the model

Background knowledge on the dynamics

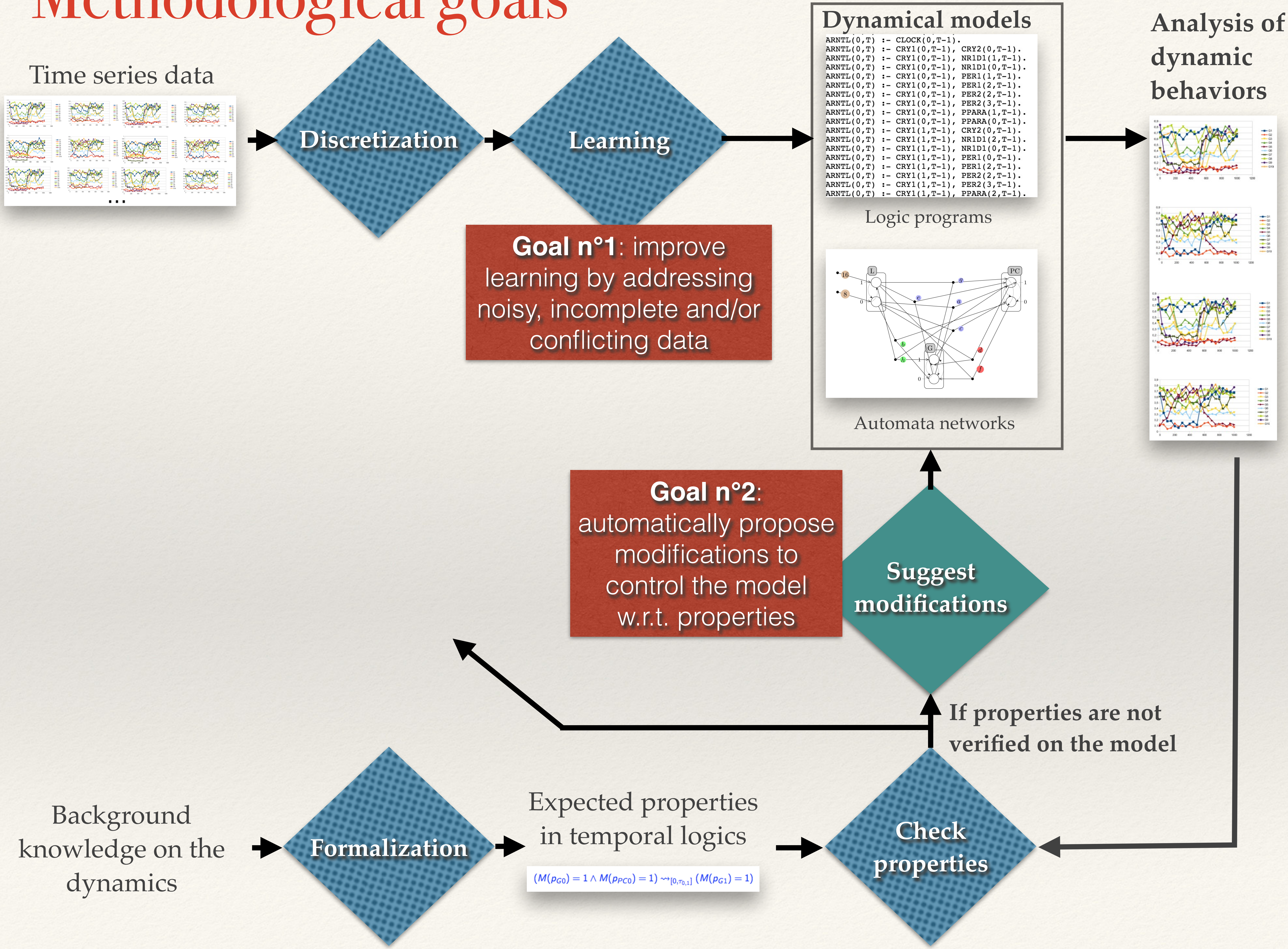
Formalization

Expected properties in temporal logics

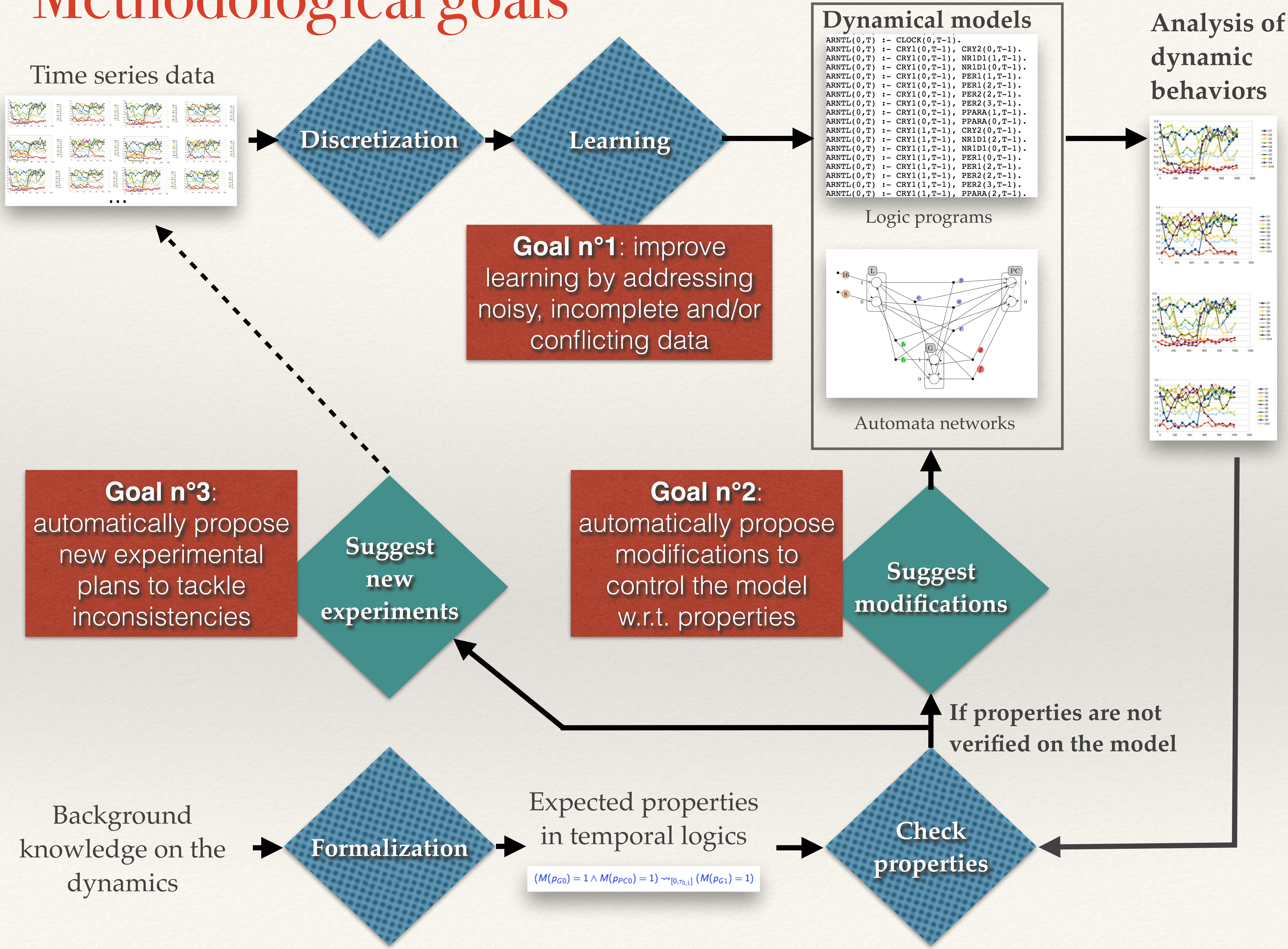
$(M(p_{G0}) = 1 \wedge M(p_{PC0}) = 1) \rightsquigarrow_{[0,\tau_{0,1}]} (M(p_{G1}) = 1)$

Check properties

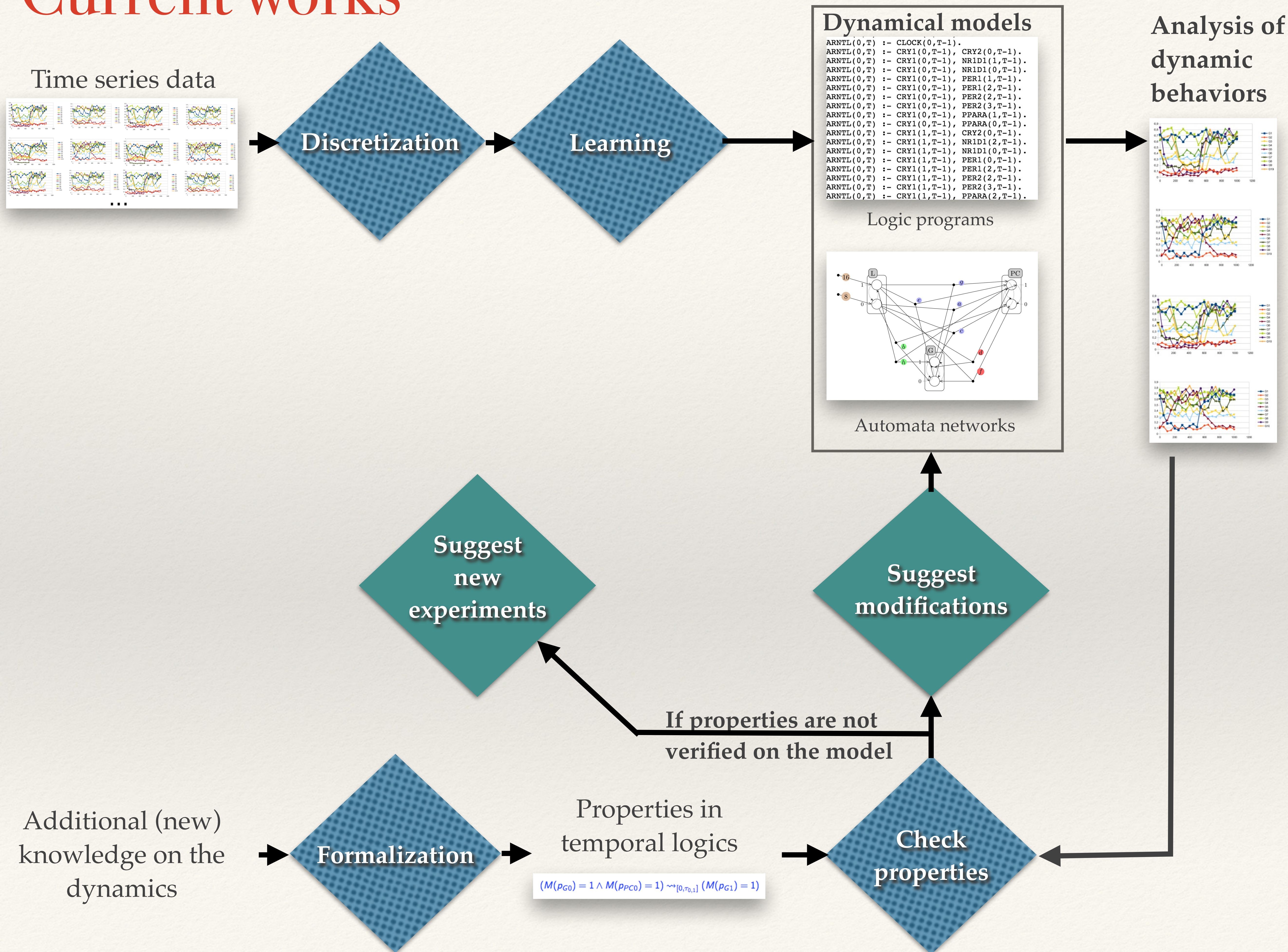
Methodological goals



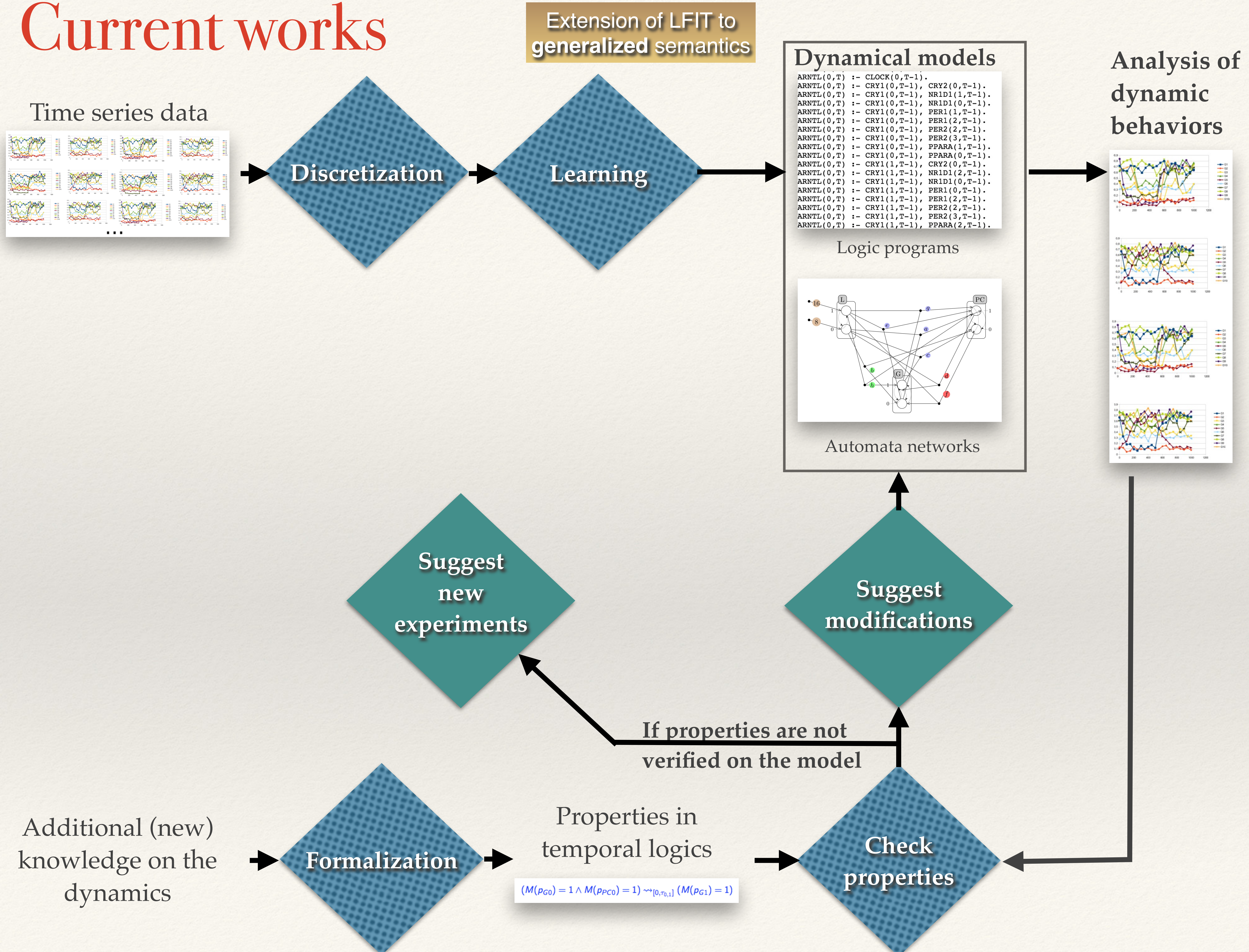
Methodological goals



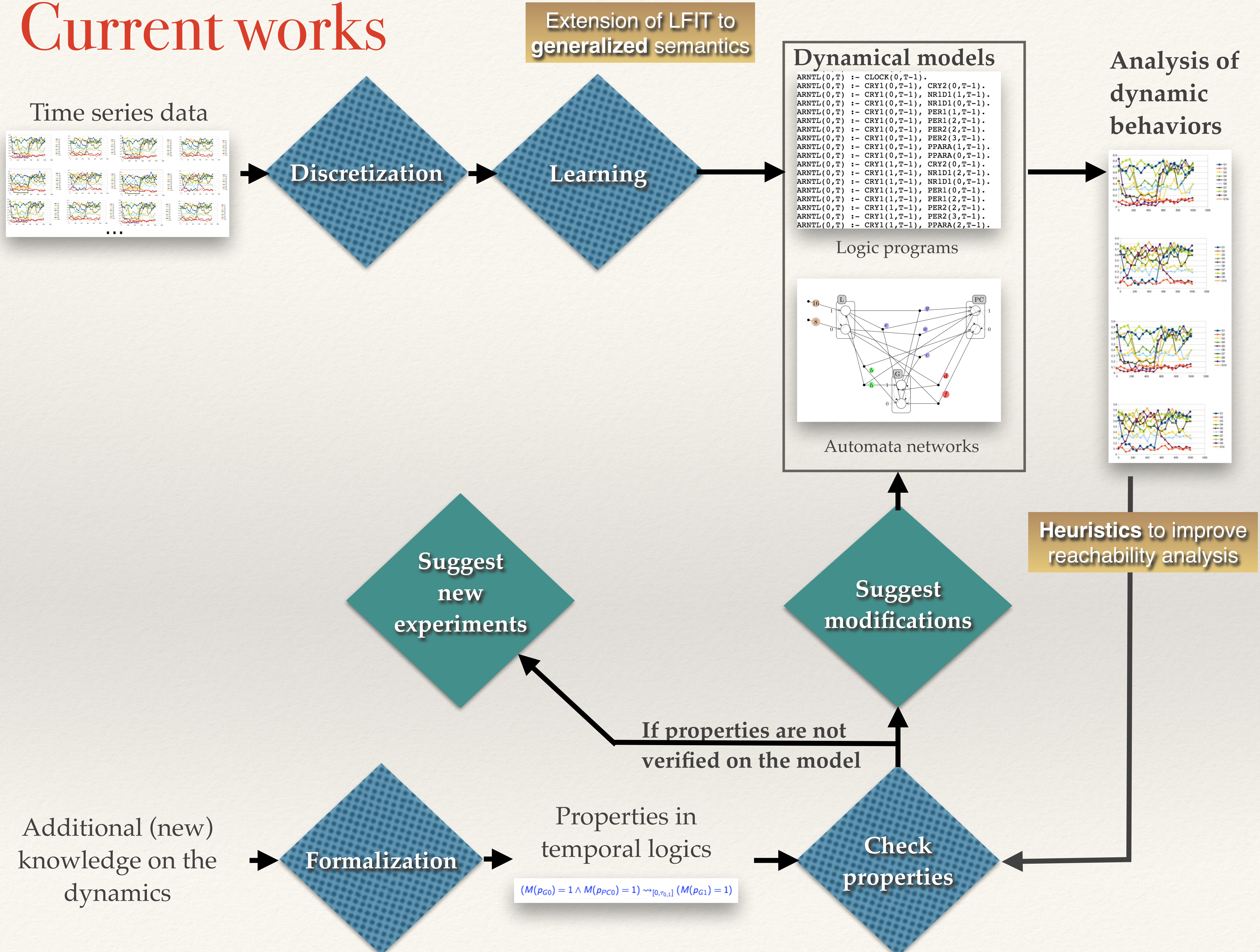
Current works



Current works



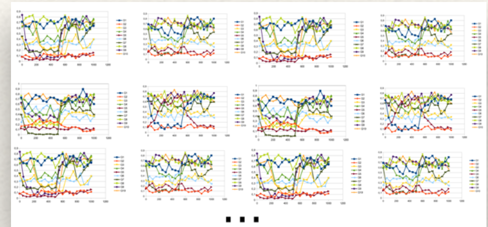
Current works



Current works

Extension of LFIT to **generalized** semantics

Time series data



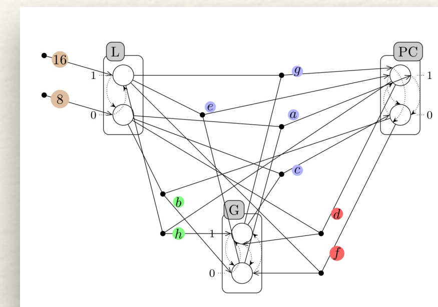
Discretization

Learning

Dynamical models

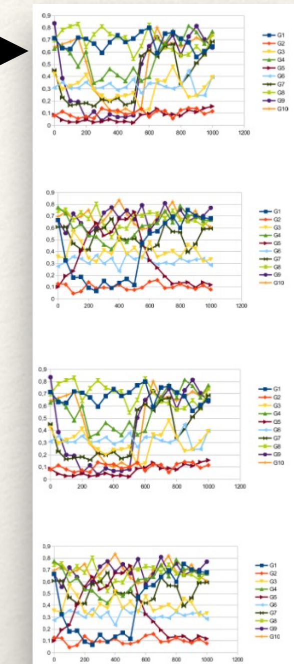
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
```

Logic programs



Automata networks

Analysis of dynamic behaviors



Heuristics to improve reachability analysis

Use of (i) **ASP** (ii) **SAT-solvers** to analyze reachability

Suggest new experiments

Suggest modifications

If properties are not verified on the model

Additional (new) knowledge on the dynamics

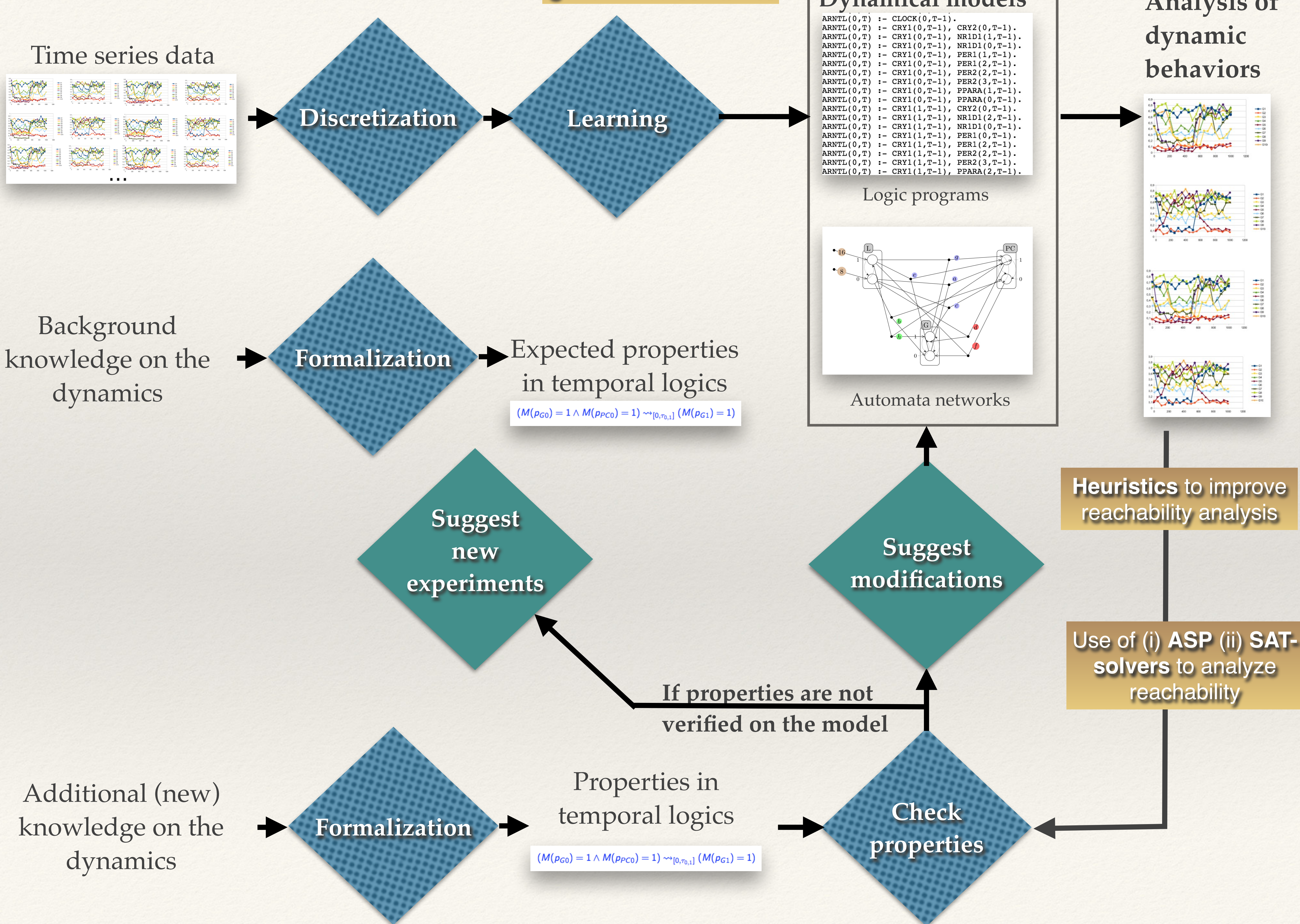
Formalization

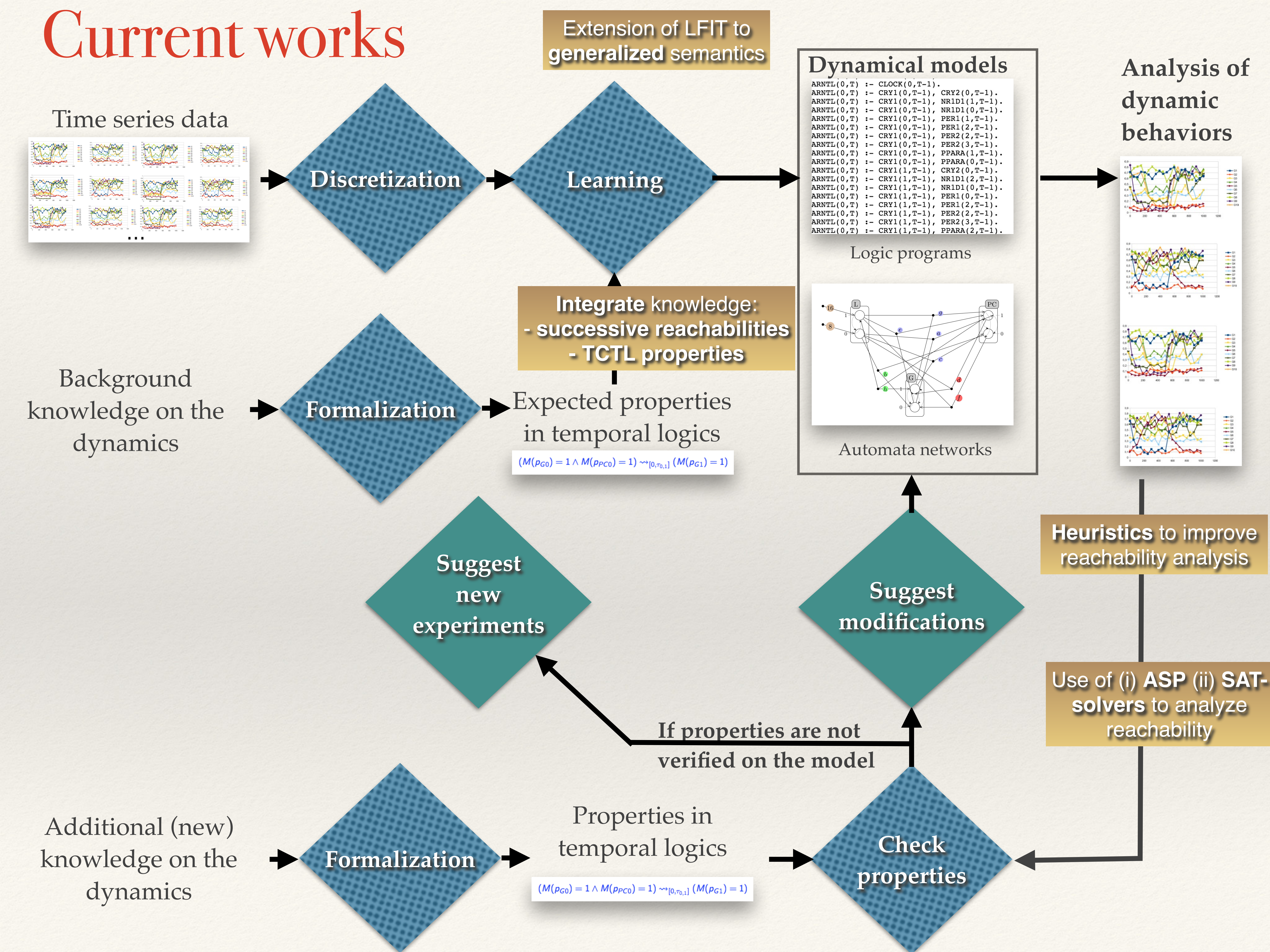
Properties in temporal logics

$(M(p_{G0}) = 1 \wedge M(p_{PC0}) = 1) \rightsquigarrow_{[0,\tau_{0,1}]} (M(p_{G1}) = 1)$

Check properties

Current works



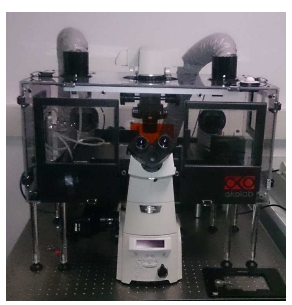

Current works
Extension of LFIT to generalized semantics
Time series data
...
Discretization
Learning
Dynamical models
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PPARA(2,T-1).
Logic programs
Automata networks
Analysis of dynamic behaviors
Integrate knowledge:
- successive reachabilities
- TCTL properties
Background knowledge on the dynamics
Formalization
Expected properties in temporal logics
$(M(p_{G0}) = 1 \wedge M(p_{PC0}) = 1) \leadsto_{[0,\tau_{0,1}]} (M(p_{G1}) = 1)$
Heuristics to improve reachability analysis
Suggest new experiments
Suggest modifications
Use of (i) ASP (ii) SAT-solvers to analyze reachability
If properties are not verified on the model
Additional (new) knowledge on the dynamics
Formalization
Properties in temporal logics
$(M(p_{G0}) = 1 \wedge M(p_{PC0}) = 1) \leadsto_{[0,\tau_{0,1}]} (M(p_{G1}) = 1)$
Check properties

With biologists working on radiotherapy

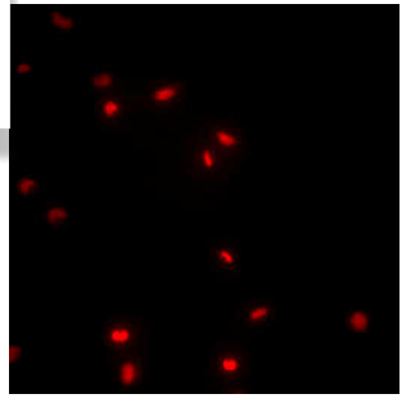
Team of François Paris
(INSERM Nantes)

- Goals:
  - Gain **better knowledge** of the radiobiology in function of the radiation therapy (RT) schemes & time
  - **Assess *in silico* the effects** of new RT protocols, reducing the need of *in vivo* experiments and helping oncologists on their clinical decision allowing personalized treatments

Time lapse



1 image every 10 minutes



1 RT = 1 timelapse

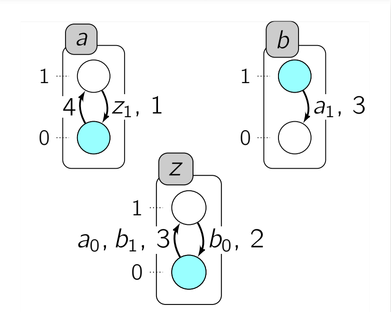
Learning

Each cell has a finite number of states (<10)

Dynamical models

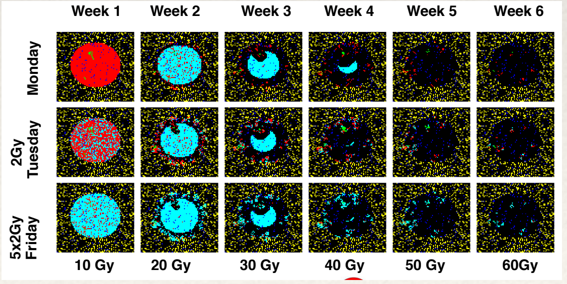
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
```

Logic program



Automata networks

Simulation of novel radiation protocols

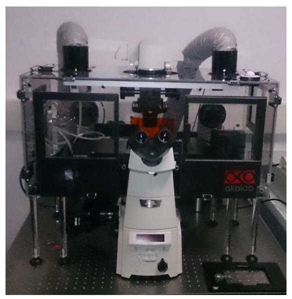


With biologists working on radiotherapy

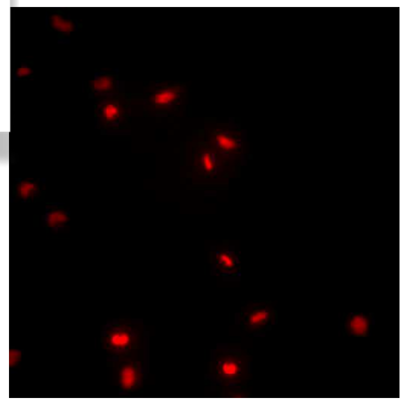
Team of François Paris
(INSERM Nantes)

- New challenges w.r.t. LFIT:
  - Model cell **division** (*mitosis*)
  - Model **spatial interactions** between cells

Time lapse



1 image every
10 minutes



1 RT = 1 timelapse

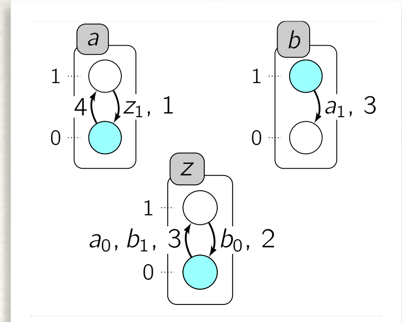
Learning

Each cell has a
finite number
of states (<10)

Dynamical models

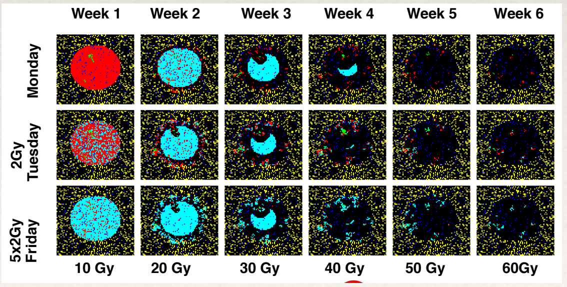
```
ARNTL(0,T) :- CLOCK(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PER2(3,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(1,T-1).
ARNTL(0,T) :- CRY1(0,T-1), PPARA(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), CRY2(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), NR1D1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(0,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER1(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(2,T-1).
ARNTL(0,T) :- CRY1(1,T-1), PER2(3,T-1).
```

Logic program



Automata networks

Simulation of
novel radiation
protocols



Références

- [BFR+15] Abdallah, E. B., Folschette, M., Roux, O., & Magnin, M. (2015, November). Exhaustive analysis of dynamical properties of Biological Regulatory Networks with Answer Set Programming. In Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on (pp. 281-285). IEEE.
- [FPI+12] Folschette, M., Paulevé, L., Inoue, K., Magnin, M., & Roux, O. (2012). Concretizing the process hitting into biological regulatory networks. In *Computational methods in systems biology* (pp. 166-186). Springer Berlin Heidelberg.
- [FMP+13] Folschette, M., Paulevé, L., Magnin, M., & Roux, O. (2013). Under-approximation of reachability in multivalued asynchronous networks. *Electronic Notes in Theoretical Computer Science*, *299*, 33-51.
- [FPI+15] Folschette, M., Paulevé, L., Inoue, K., Magnin, M., & Roux, O. (2015). Identification of biological regulatory networks from Process Hitting models. Theoretical Computer Science, 568, 49-71.
- [FPM+15] Folschette, M., Paulevé, L., Magnin, M., & Roux, O. (2015). Sufficient conditions for reachability in automata networks with priorities. *Theoretical Computer Science*, *608*, 66-83.

Références

- [PMR11a] Paulevé, L., Magnin, M., & Roux, O. (2011). Refining dynamics of gene regulatory networks in a stochastic π-calculus framework. In *Transactions on computational systems biology xiii* (pp. 171-191). Springer Berlin Heidelberg.
- [PMR11b] Paulevé, L., Magnin, M., & Roux, O. (2011). Tuning temporal features within the stochastic π-calculus. *Software Engineering, IEEE Transactions on*, *37*(6), 858-871.
- [PMR12] Paulevé, L., Magnin, M., & Roux, O. (2012). Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*, *22*(04), 651-685.
- [PCF+14] Paulevé, L., Chancellor, C., Folschette, M., Magnin, M., & Roux, O. (2014). Analyzing large network dynamics with process hitting. *Logical Modeling of Biological Systems*, 125-166.
- [RMI+15a] Ribeiro, T., Magnin, M., Inoue, K., & Sakama, C. (2015). Learning delayed influences of biological systems. *Frontiers in bioengineering and biotechnology*, *2*.
- [RMI+15b] Ribeiro, T., Magnin, M., Inoue, K., & Sakama, C. (2014). Learning Multi-valued Biological Models with Delayed Influence from Time-Series Observations. ICMLA 2015: 25-31

Références

- [DB08] Davidich, M. I., & Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PloS one*, *3*(2), e1672.
- [FNL+09] Fauré, A., Naldi, A., Lopez, F., Chaouiya, C., Ciliberto, A., & Thieffry, D. (2009). Modular logical modelling of the budding yeast cell cycle. Molecular BioSystems, 5(12), 1787-1796.
- [FHS07] François, P., Hakim, V., & Siggia, E. D. (2007). Deriving structure from evolution: metazoan segmentation. *Molecular systems biology*, *3*(1), 154.
- [IRS14] Inoue, K., Ribeiro, T., & Sakama, C. (2014). Learning from interpretation transition. Machine learning, 94(1), 51-79.
- [KSL+06] Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L., & Gilles, E. D. (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC bioinformatics*, *7*(1), 56.
- [LLL+04] Li, F., Long, T., Lu, Y., Ouyang, Q., & Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(14), 4781-4786.

Références

- [PJA+11] Prill, R. J., Saez-Rodriguez, J., Alexopoulos, L. G., Sorger, P. K., & Stolovitzky, G. (2011). Crowdsourcing network inference: the DREAM predictive signaling network challenge. Science signaling, 4(189), mr7.
- [SFL+09] Sahin, Ö., Fröhlich, H., Löbke, C., Korf, U., Burmester, S., Majety, M., ... & Poustka, A. (2009). Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC systems biology*, *3*(1), 1.
- [SSA+09] Samaga, R., Saez-Rodriguez, J., Alexopoulos, L. G., Sorger, P. K., & Klamt, S. (2009). The logic of EGFR/ErbB signaling: theoretical properties and analysis of high-throughput data. *PLoS Comput Biol*, *5*(8), e1000438.
- [SSL+07] Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., ... & Schraven, B. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput Biol*, *3*(8), e163.
- [TT95] Thieffry D, Thomas R. 1995. Dynamical behaviour of biological regulatory networks--II. Immunity control in bacteriophage lambda.. Bulletin of mathematical biology. 57(2):277-97.