Formal approaches to model gene regulatory networks

Gilles Bernot

University of Nice SOPHIA ANTIPOLIS, I3S laboratory, France

Acknowledgments:

Observability Group of the Epigenomics Project

Menu

1. Modelling biological regulatory networks
2. Discrete framework for biological regulatory networks
3. Temporal logic and Model Checking for biology
4. Computer aided elaboration of formal models
5. Pedagogical example: *Pseudomonas aeruginosa*
6. Some current research topics
7. An extension to delays

Mathematical Models and Simulation

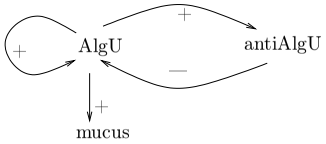
1. Rigorously encode sensible knowledge into ODEs for instance
2. - A few parameters are approximatively known
   - Some parameters are limited to some intervals
   - Many parameters are *a priori* unknown
3. Perform lot of simulations, compare results with known behaviours, and propose some credible values of the unknown parameters which produce acceptable behaviours
4. Perform additional simulations reflecting novel situations
5. If they predict interesting behaviours, propose new biological experiments
6. Simplify the model and try to go further

Static Graph *v.s.* Dynamic Behaviour

Difficulty to predict the result of combined regulations

Difficulty to measure the strength of a given regulation

Example of "competitor" circuits



Positive *v.s.* Negative circuits

Even *v.s.* Odd number of "—" signs

Multistationarity *v.s.* Homeostasy

René Thomas, Snoussi, . . . , Soulé, Richard

Functional circuits "pilot" the behaviour

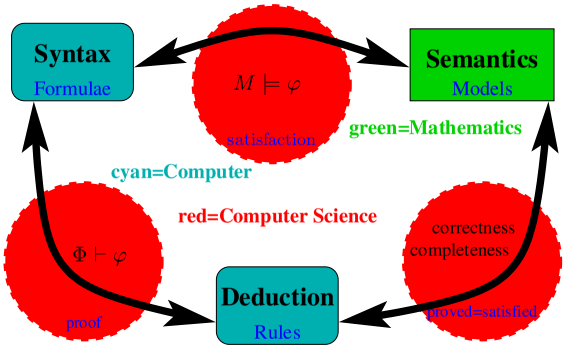
Mathematical Models and Validation

"Brute force" simulations are not the only way to use a computer.
We can offer computer aided environments which help:

- to avoid models that can be "tuned" *ad libitum*
- to validate models with a reasonable number of experiments
- to define only models that could be experimentally refuted
- to prove refutability w.r.t. experimental capabilities

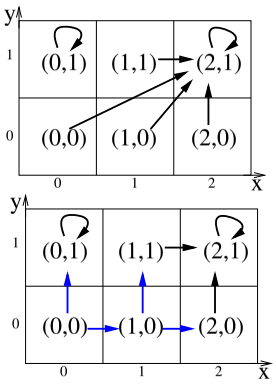
Observability issues:
Observability Group, Epigenomics Project.

Formal Logic: syntax/semantics/deduction



State Graphs

(x,y)	*Focal Point*
(0,0)	$(K_{x,\overline{y}}, K_y)$=(2,1)
(0,1)	(K_x, K_y)=(0,1)
(1,0)	$(K_{x,x\overline{y}}, K_y)$=(2,1)
(1,1)	$(K_{x,x}, K_y)$=(2,1)
(2,0)	$(K_{x,x\overline{y}}, K_{y,x})$=(2,1)
(2,1)	$(K_{x,x}, K_{y,x})$=(2,1)



"desynchronization" ⟶
by units of Manhattan distance

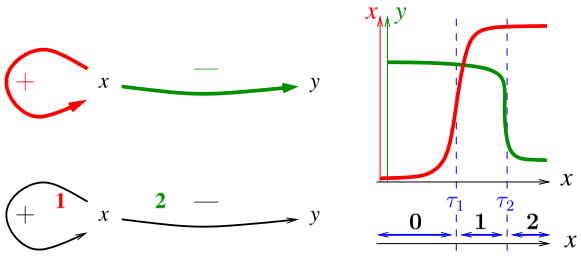
Menu

1. Modelling biological regulatory networks
2. Discrete framework for biological regulatory networks
3. Temporal logic and Model Checking for biology
4. Computer aided elaboration of formal models
5. Pedagogical example: *Pseudomonas aeruginosa*
6. Some current research topics
7. An extension to delays

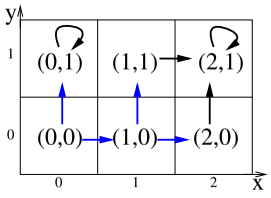
Menu

1. Modelling biological regulatory networks
2. Discrete framework for biological regulatory networks
3. Temporal logic and Model Checking for biology
4. Computer Aided elaboration Of Formal models
5. Pedagogical example: *Pseudomonas aeruginosa*
6. Some current research topics
7. An extension to delays

Multivalued Regulatory Graphs

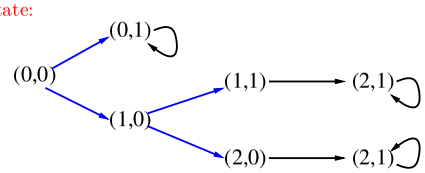


Time has a tree structure

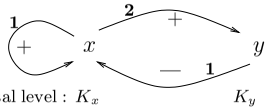


As many possible state graphs
as possible parameter sets...
(huge number)

From an initial state:



Regulatory Networks (R. Thomas)



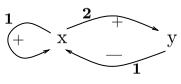
Basal level : K_x K_y
x helps : $K_{x,x}$ $K_{y,x}$
Absent y helps : $K_{x,\overline{y}}$
Both : $K_{x,x\overline{y}}$

(x,y)	*Focal Point*
(0,0)	$(K_{x,\overline{y}}, K_y)$
(0,1)	(K_x, K_y)
(1,0)	$(K_{x,x\overline{y}}, K_y)$
(1,1)	$(K_{x,x}, K_y)$
(2,0)	$(K_{x,x\overline{y}}, K_{y,x})$
(2,1)	$(K_{x,x}, K_{y,x})$

CTL to encode Biological Properties

Common properties:

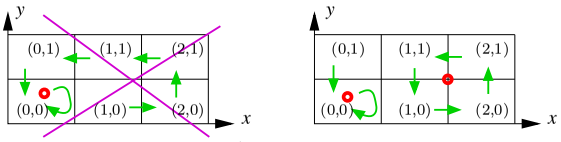
"functionality" of a sub-graph



Special role of "feedback loops"

– positive: *multistationnarity* (even number of —)

– negative: *homeostasy* (odd number of —)



Characteristic properties: $\begin{cases} (x=2) \Longrightarrow AG(\neg(x=0)) \\ (x=0) \Longrightarrow AG(\neg(x=2)) \end{cases}$

They express *"the positive feedback loop is functional"*

(satisfaction of these formulae relies on the parameters $K_{...}$)

CTL = Computation Tree Logic

Atoms = comparaisons : (x=2) (y>0) ...

Logical connectives: $(\varphi_1 \wedge \varphi_2)$ $(\varphi_1 \Longrightarrow \varphi_2)$ ...

Temporal modalities: made of 2 characters

first character	second character
A = for **A**ll path choices	X = ne**X**t state
	F = for some **F**uture state
E = there **E**xist a choice	G = for all future states (**G**lobally)
	U = **U**ntil

$AX(y=1)$: the concentration level of y belongs to the interval 1 in all states directly following the considered initial state.

$EG(x=0)$: there exists at least one path from the considered initial state where x always belongs to its lower interval.

Model Checking

Efficiently computes all the states of a state graph which satisfy a given formula: $\{\ \eta\ \mid\ M\ \models_\eta\ \varphi\ \}$.

Efficiently select the models which globally satisfy a given formula.

Temporal Connectives of CTL

neXt state:

$EX\varphi$: φ can be satisfied in a next state

$AX\varphi$: φ is always satisfied in the next states

eventually in the Future:

$EF\varphi$: φ can be satisfied in the future

$AF\varphi$: φ will be satisfied at some state in the future

Globally:

$EG\varphi$: φ can be an invariant in the future

$AG\varphi$: φ is necessarily an invariant in the future

Until:

$E[\psi U\varphi]$: there exist a path where ψ is satisfied until a state where φ is satisfied

$A[\psi U\varphi]$: ψ is always satisfied until some state where φ is satisfied

Model Checking for CTL

Computes all the states of a theoretical model which satisfy a given formula: $\{\ \eta\ \mid\ M\ \models_\eta\ \varphi\ \}$.

Idea 1: work on the state graph instead of the path trees.

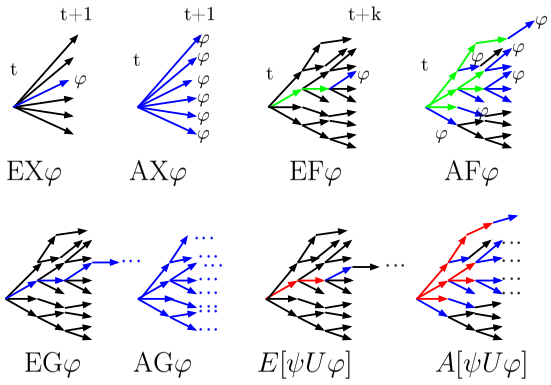
Idea 2: check first the atoms of φ and then check the connectives of φ with a bottom-up computation strategy.

Idea 3: (computational optimization) group some cases together using BDDs (Binary Decision Diagrams).

Example: $(x=0) \Longrightarrow AG(\neg(x=2))$

Obsession: *travel the state graph as less as possible*

Semantics of Temporal Connectives



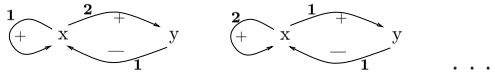
Computer Aided Elaboration of Models

From biological knowledge and/or biological hypotheses, it comes:

- properties:

  "Without stimulus, if gene x has its basal expression level, then it remains at this level."

- model schemas:

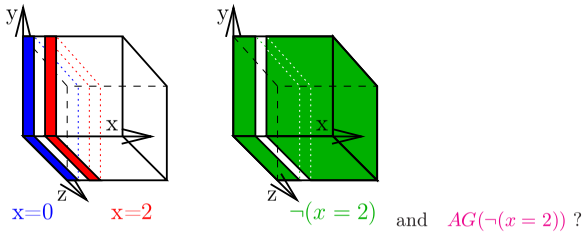


. . .

Formal logic and formal models allow us to:

- verify hypotheses and check consistency
- elaborate more precise models incrementally
- suggest new biological experiments to efficiently reduce the number of potential models

$(x = 0) \implies AG(\neg(x = 2))$



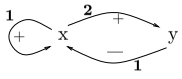
and $AG(\neg(x = 2))$?

. . . one should **travel <u>all</u>** the paths from any green box and check if successive boxes are green: *too many boxes to visit.*

Trick: $AG(\neg(x = 2))$ is equivalent to $\neg EF(x = 2)$
start from the red boxes and follow the transitions backward.

The Two Questions

$\Phi = \{\varphi_1, \varphi_2, \cdots, \varphi_n\}$ and $\mathcal{M} =$



. . .

1. Is it possible that Φ and $\mathcal{M}$?
   Consistency of knowledge and hypotheses. Means to select models belonging to the schemas that satisfy Φ.
   $(\exists?\ M \in \mathcal{M}\ \mid\ M \models \varphi)$
2. If so, is it true *in vivo* that Φ and $\mathcal{M}$?
   Compatibility of one of the selected models with the biological object. Require to propose experiments to **validate** or **refute** the selected model(s).

→ Computer aided *proofs* and *validations*

Theoretical Models ↔ Experiments

CTL formulae are satisfied (or refuted) w.r.t. a set of paths from a given initial state

- They can be tested against the possible paths of the theoretical models ($M \models_{Model\ Checking} \varphi$)
- They can be tested against the biological experiments ($Biological\_Object \models_{Experiment} \varphi$)

<u>CTL formulae link theoretical models and biological objects together</u>

Question 1 = Consistency

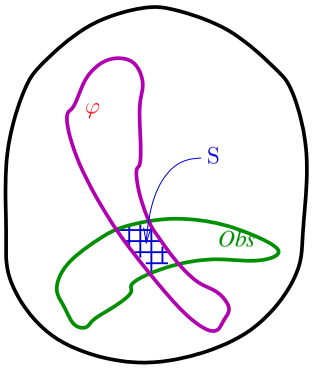
1. Draw all the sensible regulatory graphs with all the sensible threshold allocations. It defines $\mathcal{M}$.
2. Express in CTL the known behavioural properties as well as the considered biological hypotheses. It defines Φ.
3. Automatically generate all the possible regulatory networks derived from $\mathcal{M}$ according to all possible parameters $K_{...}$.
   Our software plateform SMBioNet handles this automatically.
4. Check each of these models against Φ.
   SMBioNet uses model checking to perform this step.
5. If no model survive to the previous step, then reconsider the hypotheses and perhaps extend model schemas. . .
6. If at least one model survives, then the biological hypotheses are consistent. Possible parameters $K_{...}$ have been indirectly established. Now Question 2 has to be addressed.

Menu

Generation of biological experiments (4)

Set of all the formulae:

φ = hypothesis
Obs = possible experiments
$Th(\varphi)$ = φ inferences
S = sensible experiments

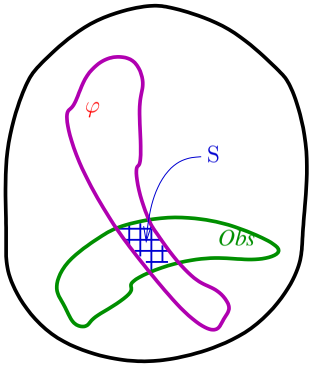


Generation of biological experiments (5)

Set of all the formulae:

φ = hypothesis
Obs = possible experiments
$Th(\varphi)$ = φ inferences
S = sensible experiments

Refutability:
S $\Longrightarrow$ φ ?



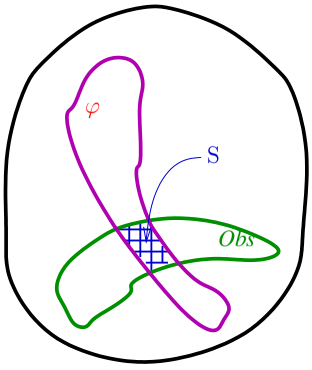
Generation of biological experiments

Set of all the formulae:

φ = hypothesis
Obs = possible experiments
$Th(\varphi)$ = φ inferences
S = sensible experiments

Refutability:
S $\Longrightarrow$ φ ?

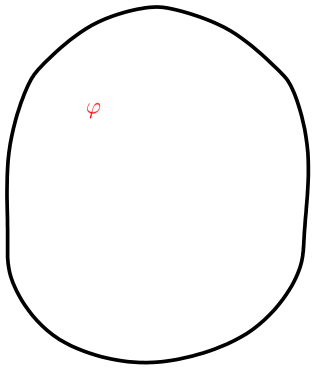
Best refutations:
Choice of experiments in S ?
. . . optimisations



Generation of biological experiments (1)

Set of all the formulae:

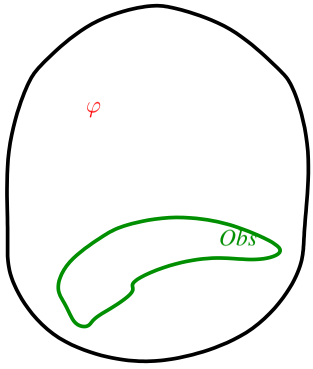
φ = hypothesis



Generation of biological experiments (2)

Set of all the formulae:

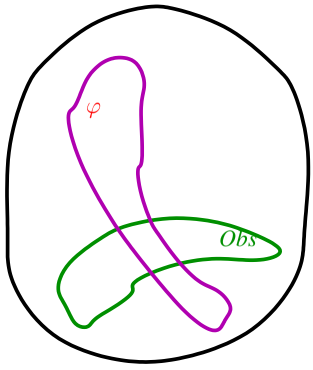
φ = hypothesis
Obs = possible experiments



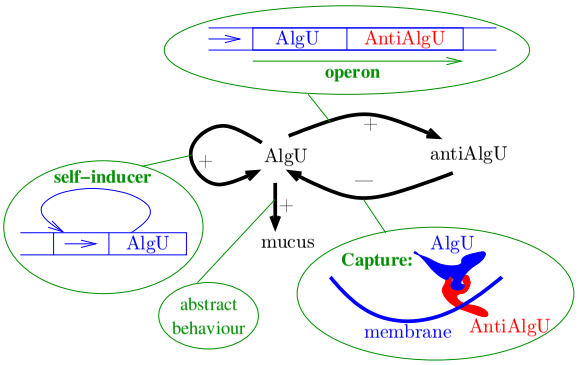
Generation of biological experiments (3)

Set of all the formulae:

φ = hypothesis
Obs = possible experiments
$Th(\varphi)$ = φ inferences



Mucus Production in *P. aeruginosa*



Question 2 = Validation

1. Among all possible formulae, some are "observable" i.e., they express a possible result of a possible biological experiment. Let *Obs* be the set of all observable formulae.
2. Let Λ be the set of theorems of Φ and $\mathcal{M}$. $\Lambda \cap Obs$ is the set of experiments able to validate the survivors of Question 1. Unfortunately it is infinite in general.
3. Testing frameworks from computer science aim at selecting a finite subsets of these observable formulae, which maximize the chance to refute the survivors.
4. These subsets are often too big, nevertheless these testing frameworks can be suitably applied to regulatory networks.

   It has been the case of the mucus production of *P.aeruginosa*.

Parameters & thresholds: unknown

Thresholds for AlgU in *P.aeruginosa* are unknown:

and parameters are unknown:

$3^4 \times 2^2$ $3^4 \times 2^2$ $2^4 \times 2^2$

712 possible models

One CTL formula for each stable state:

$$(\text{AlgU} = 2) \Longrightarrow AXAF(\text{AlgU} = 2)$$
$$(\text{AlgU} = 0) \Longrightarrow AG(\neg(\text{AlgU} = 2))$$

Question 1, consistency: proved by *Model Checking*
→ 10 models among the 712 models are extracted by SMBioNet

Menu

Validation of the epigenetic hypothesis

Question 2 = to validate bistationnarity *in vivo*

Non mucoid state: $(\text{AlgU} = 0) \Longrightarrow AG(\neg(\text{AlgU} = 2))$
P. aeruginosa, with a basal level for AlgU *does not produce mucus spontaneously*: actually validated

Mucoid state: $(\text{AlgU} = 2) \Longrightarrow AXAF(\text{AlgU} = 2)$

Experimental limitation:
AlgU can be saturated but it cannot be measured.

Experiment:

to pulse AlgU *and then to test if mucus production remains*

($\Longleftrightarrow$ to verify a hysteresis)

This experiment can be generated automatically

Mutation, Epigenesis, Adaptation

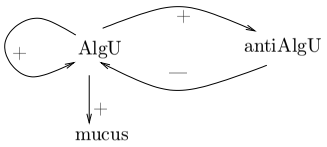
Terminology about phenotype modification:

genetic modification: inheritable and not reversible (mutation)
epigenetic modification: inheritable and reversible
adaptation: not inheritable and reversible

The biological question (Janine Guespin):
is **mucus production** in *Pseudomonas aeruginosa* due to an epigenetic switch ? $\Longrightarrow$ New possible therapy
[→ cystic fibrosis]

To test (AlgU=2)$\Longrightarrow AXAF$(AlgU=2)

AlgU = 2 cannot be directly verified but $mucus = 1$ can be verified.



Lemma: $AXAF(\text{AlgU} = 2) \Longleftrightarrow AXAF(mucus = 1)$
(. . . formal proof by computer . . .)

→ To test: $(\text{AlgU} = 2) \Longrightarrow AXAF(mucus = 1)$

(AlgU = 2) $\Longrightarrow AXAF(mucus = 1)$

Karl Popper:
to validate = to try to refute
thus A=false is useless
experiments must begin with a pulse

$A \Longrightarrow B$	*true*	*false*
true	true	false
false	true	true

The pulse forces the bacteria to reach the initial state AlgU = 2.
If the state were not directly controlable we had to prove lemmas:

(*something reachable*) $\Longrightarrow$ (AlgU = 2)

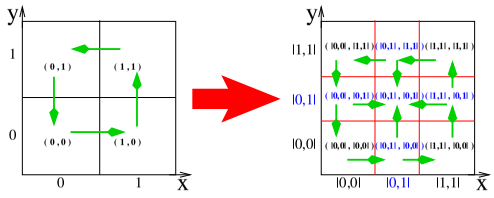
General form of a test:

(*something reachable*) $\Longrightarrow$ (*something observable*)

Menu

Research topics (1)

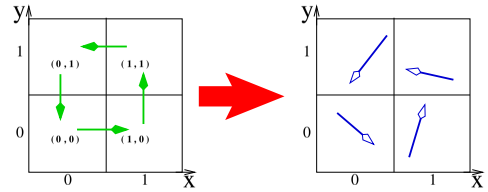
Explicit singular states:



e.g. to distinguish stable states from limit cycles

Research topics (2)

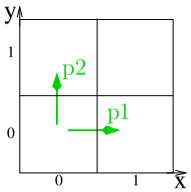
Hybrid approaches:
simplified trajectories which locally approximate differential equations



(e.g. linear)

Research topics (4)

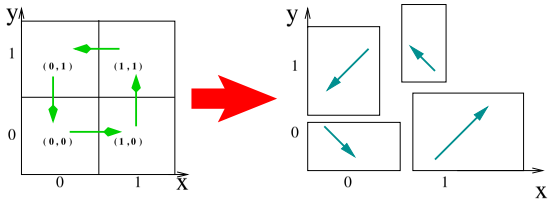
Stochastic approaches:



More or less dual to delays

Research topics (3)

Time delays:

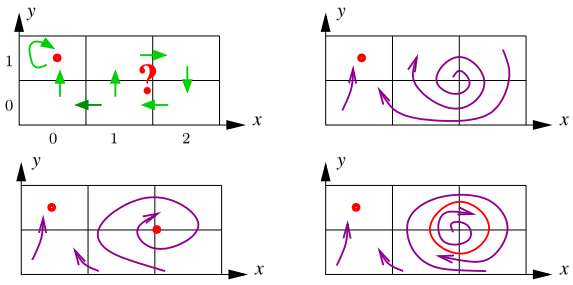


(size of rectangular areas = delays)
Requires constraint solving

Menu

1. Modelling biological regulatory networks
2. Discrete framework for biological regulatory networks
3. Temporal logic and Model Checking for biology
4. Computer aided elaboration of formal models
5. Pedagogical example: *Pseudomonas aeruginosa*
6. Some current research topics
7. An extension to delays

Ambiguous discrete models

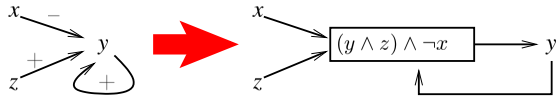


1 ou 2 attraction basins ?

It depends on the relative *delays* for x and y to cross each of the four domains.

Research topics (5)

Networks with multiplexes:



Explicit encoding of knowledge on cooperations

Research topics (6)

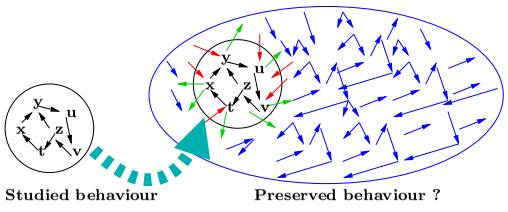
From static shapes to properties on dynamics:

- positive/negative cycles and epigenesis/homeostasis
- maximum number of attraction basins
- ...

Mathematical proofs similar to the ones for cellular automaton

Research topics (7)

Embeddings of Regulatory Networks:



Necessary and sufficient condition on the *local* dynamics of the "input frontier"

Offers a methodology to identify interesting sub-networks

Regulatory Network with Delays

$\mathcal{N} = (V, E, K, D)$

– *As usual :* bounded variables (V),
edges with sign and threshold (E), family of parameters (K)

– *Production and degradation delays :* $D = D^+ \cup D^-$

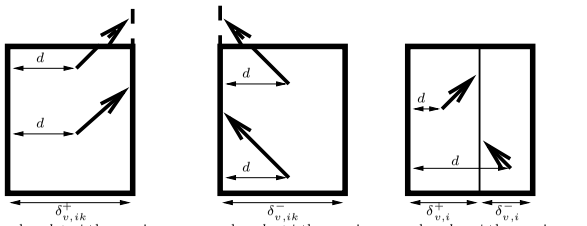
— $D^+ = \{\delta^+_{v,i\omega}\}_{v\in V, i\in[0,K_{v,\omega}],\omega\subset G^{-1}(v)}$

— $D^- = \{\delta^-_{v,i\omega}\}_{v\in V, i\in[K_{v,\omega},b_v],\omega\subset G^{-1}(v)}$

Delays vary in $\mathbb{R}^+$ according to the current state (i) and the resources (ω) of a variable v

Dynamics within a unique domain

A state : $\eta = (l, d)$ where $l : V \to \mathbb{N}$ is a discrete state as usual
and $d : V \to \mathbb{R}^+$ satisfies : $d(v) \leq \delta^+_{v,l(v)\omega_l(v)} + \delta^-_{v,l(v)\omega_l(v)}$



when $k > i$ the moving delay is $\delta^+_{v,ik} - d$ in the direction of v (horizontal)

when $k < i$ the moving delay is d in the direction of v (horizontal)

when $k = i$ the moving delay is $|\delta^+_{v,ik} - d|$ in the direction of v (horizontal)

Moving delay : $\mu_\eta(v) = |\delta^+_{v,l(v)\omega_l(v)} - d(v)|$

Dynamics

$\eta' = (l', d')$ is a successor of $\eta = (l, d)$ iff $\exists u \in V$ s.t.:

– $\forall v \in V,\ \mu_\eta(u) \leq \mu_\eta(v)$

– $l'(u) = l(u) + \kappa_\eta(u)$ with $\kappa_\eta(u) \in \{-1, 0, 1\}$ as usual

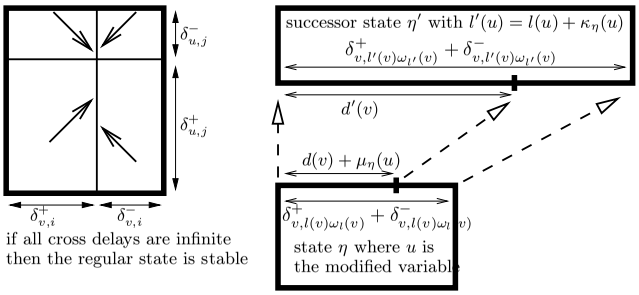
– $\forall v \in V,\quad u \neq v \Longrightarrow l'(v) = l(v)$

– $\kappa_\eta(u) = 0 \Longrightarrow (\forall v \in V,\ d'(v) = \delta^+_{v,l(v)\omega_l(v)})$

– $\kappa_\eta(u) \neq 0 \Longrightarrow d'(u) = 0$

– $\kappa_\eta(u) \neq 0 \implies (\ \forall v \in V,\quad u \neq v \Longrightarrow$
$d'(v) = \frac{(d(v)+\mu_\eta(u)\ \times\ (\delta^+_{v,l'(v)\omega_{l'}(v)} + \delta^-_{v,l'(v)\omega_{l'}(v)})}{\delta^+_{v,l(v)\omega_l(v)} + \delta^-_{v,l(v)\omega_l(v)}}\ )$

Dynamics = Thales in the space of delays



if all cross delays are infinite
then the regular state is stable

Concluding Comments

Models to encode already elucidated biological models *v.s.* modelling methods to help discovery in biology...

Behavioural *properties* (Φ) are as much important as *models* $(\mathcal{M})$

Modelling is significant only with respect to the considered experimental *reachability* and *observability* (Obs)

Formal proofs can suggest wet experiments

But...

even very simple approaches for delays are unreachable for model checking : we currently explore constraint programming methods.