

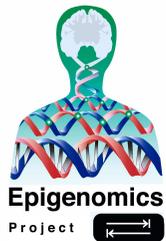
Applications des méthodes formelles aux réseaux de régulation biologique

Gilles Bernot

Programme ÉPIGÉNOMIQUE et LAMI

(Laboratoire de Méthodes Informatiques)

Genopole[®] et UMR 8042 C.N.R.S.–Université d'Évry



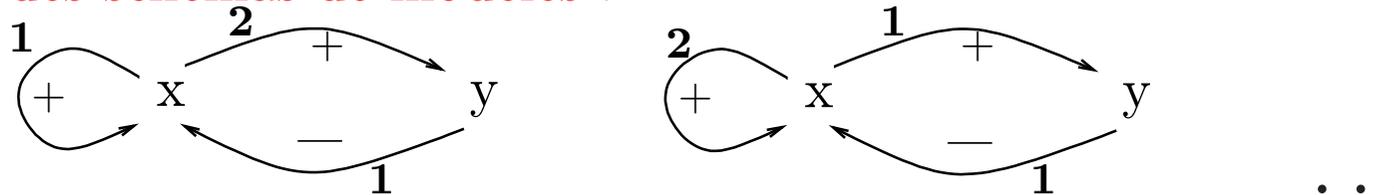
Plan

1. Modèles & logiques *formels*, cohérence, validation
2. Formalisation des réseaux de régulation biologique
3. Logique temporelle et Model Checking
4. Exemple (J.Guespin) : *Pseudomonas aeruginosa*
 - Cohérence par Model Checking
 - Extraction automatique de plans d'expériences
 - Propriétés expérimentalement observables
5. Questions d'actualité

Modèles *vs.* Formules

Des connaissances et/ou des hypothèses biologiques on extrait :

– des schémas de modèles :



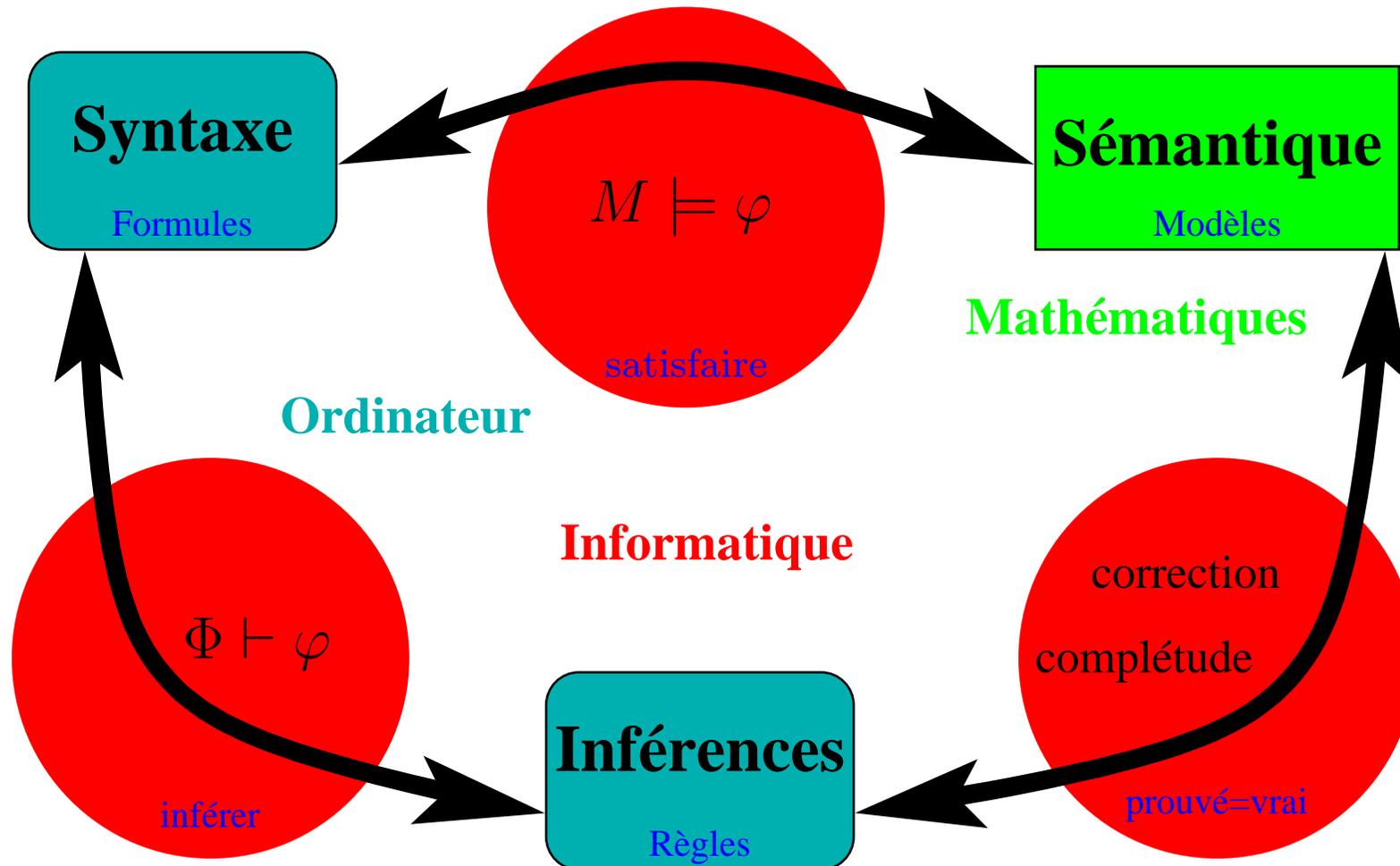
– des propriétés :

« *Sans stimuli, si x est à son niveau basal, il y reste* ».

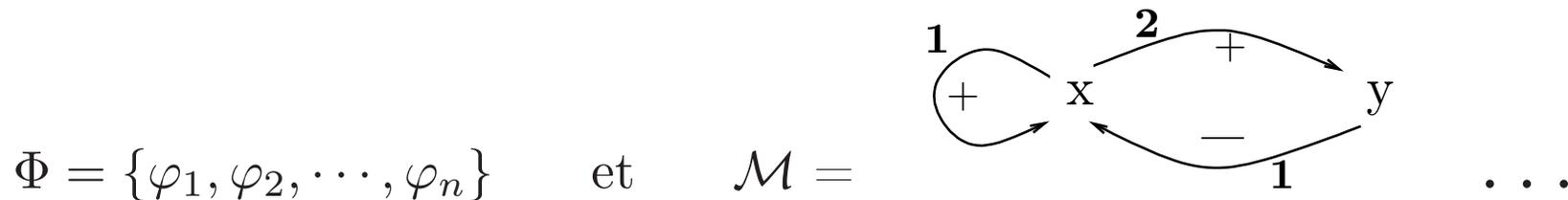
Vérifier ces connaissances et/ou ces hypothèses de manière assistée par ordinateur \implies formaliser les modèles et les propriétés.

Utiliser une *logique formelle*, avec des ensembles de formules Φ

Logiques formelles & modèles



Les 2 questions à traiter



1. Est-il possible que Φ et \mathcal{M} ?

Cohérence des connaissances et des hypothèses.

Construire des modèles issus des schémas précédents satisfaisant Φ . ($\exists? M \in \mathcal{M} \mid M \models \varphi$)

2. Si oui, est-il vrai *in vivo* que Φ et \mathcal{M} ?

Compatibilité d'un des modèles trouvés avec l'objet biologique.

Proposer des plans d'expériences pour le **valider**/tester.

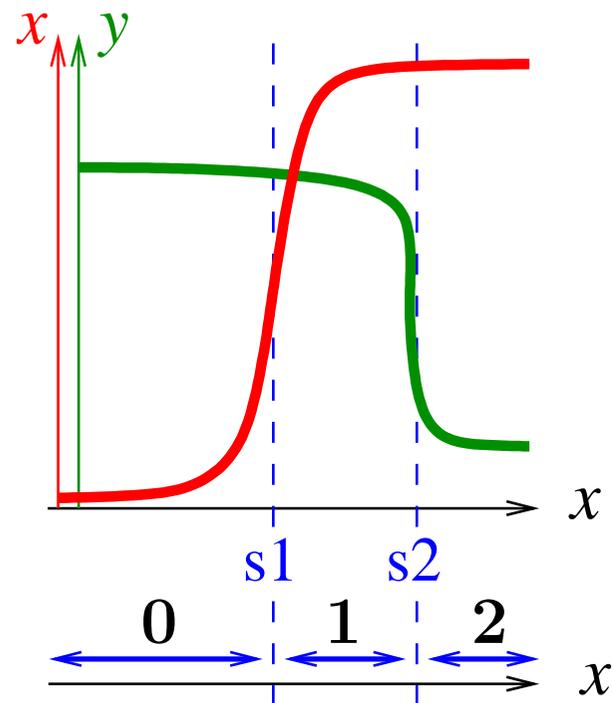
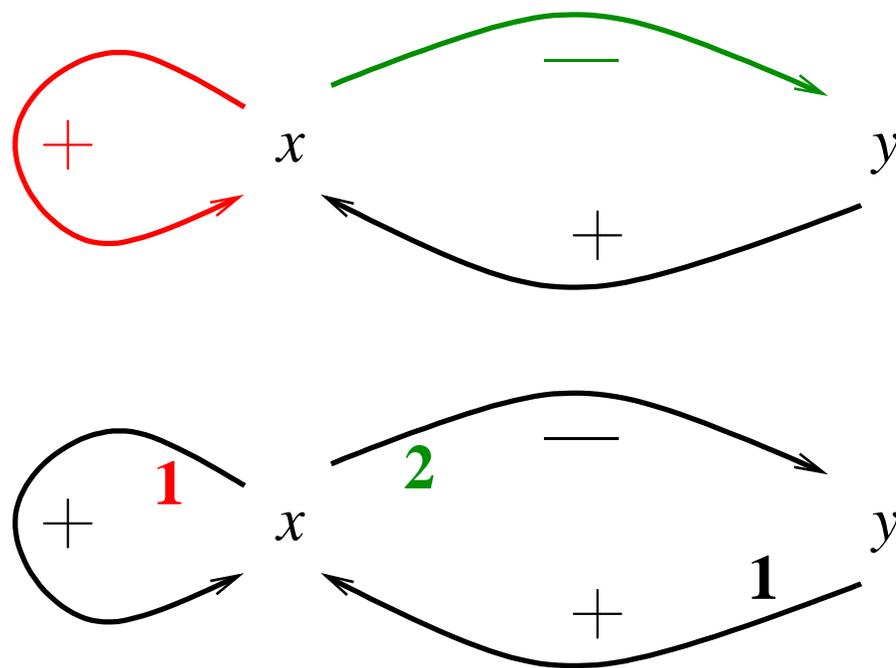
→ *Preuve et Validation* \implies peuvent être assistées par ordinateur

Plan

1. Modèles & logiques *formels*, cohérence, validation
2. **Formalisation des réseaux de régulation biologique**
3. Logique temporelle et Model Checking
4. Exemple (J.Guespin) : *Pseudomonas aeruginosa*
 - Cohérence par Model Checking
 - Extraction automatique de plans d'expériences
 - Propriétés expérimentalement observables
5. Questions d'actualité

Graphes de régulation *multivaluée*

Abstraire par une variable : un gène, l'ARN
et la protéine pour laquelle il code



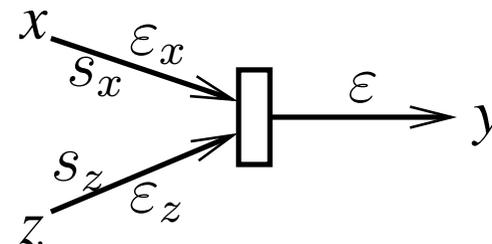
Formalisation des graphes de régulation

Grphe à seuils : couple $(\mathcal{N}, \mathcal{A})$

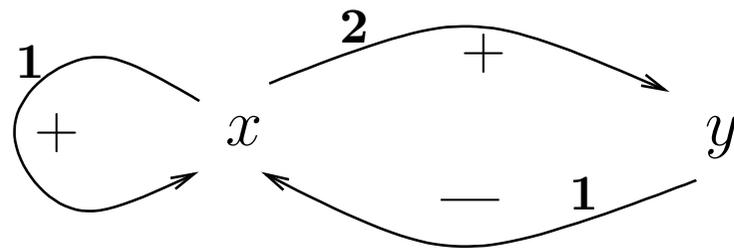
- Nœud = variable x munie d'une $borne_x \in \mathbb{N}$,
inférieure au degré sortant de x .
- Arc $x \rightarrow y$ étiqueté par $\varepsilon \in \{+, -\}$ et $s \in [0 \dots borne_x]$.

Variante : graphe bipartite

- formation de complexes protéiques
- inhibition d'une influence
- environnement, flux, système ouvert



Réseaux de régulation (R. Thomas)



niveau basal : K_x
 x aide : $K_{x,x}$
 y absent aide : $K_{x,\bar{y}}$
 les deux : $K_{x,x\bar{y}}$

K_y
 $K_{y,x}$

(x,y)	<i>Attracteur</i>
(0,0)	$(K_{x,\bar{y}}, K_y)$
(0,1)	(K_x, K_y)
(1,0)	$(K_{x,x\bar{y}}, K_y)$
(1,1)	$(K_{x,x}, K_y)$
(2,0)	$(K_{x,x\bar{y}}, K_{y,x})$
(2,1)	$(K_{x,x}, K_{y,x})$

Formalisation des réseaux de régulation

Marquage de niveaux de concentration :

$\eta : \mathcal{N} \rightarrow \mathbb{N}$ (= vecteur de valeurs entières, inférieures à la borne pour chaque nœud). $\eta(x) = \text{niveau de concentration de } x$

Variante : état *singulier* (certaines valeurs sont les seuils s_1, s_2, \dots)

Arc franchissable :

niveau d'entrée supérieur (resp. inférieur) à s si $\varepsilon = +$ (resp. $\varepsilon = -$)

Paramètres de transitions :

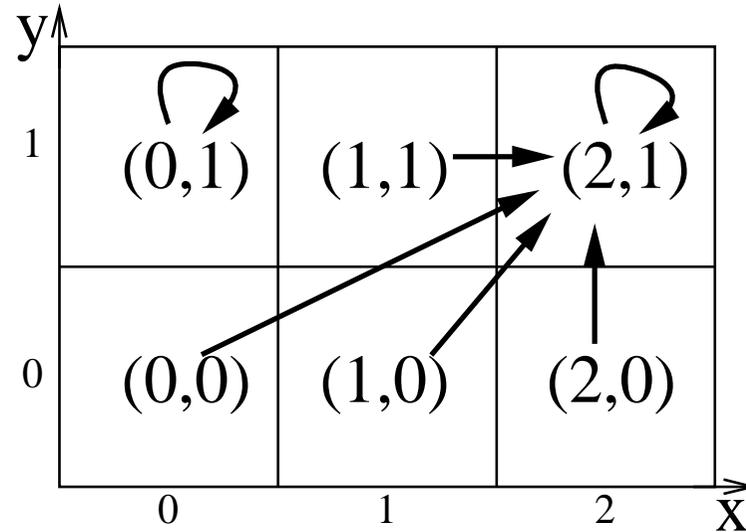
fonction partielle $K : \mathcal{N} \times \mathcal{P}(\mathcal{N}) \rightarrow \mathbb{N}$

Vecteur attracteur :

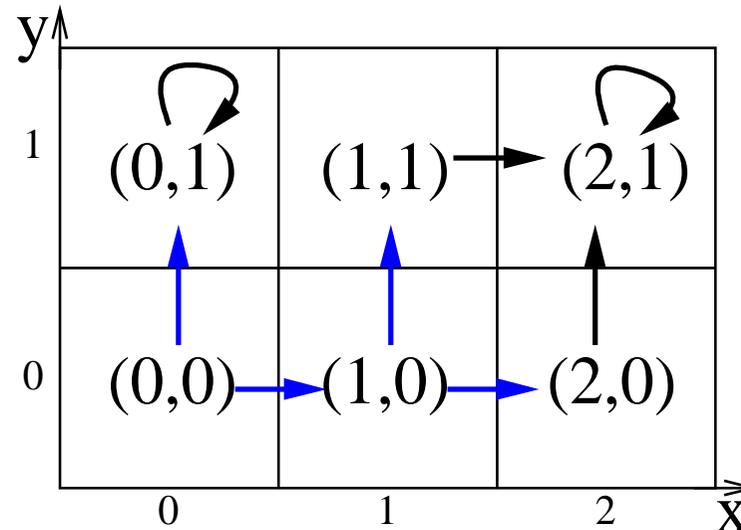
état (régulier) « attracteur » donné par les $K(x, \omega)$

Réseau de régulation → Graphe d'états

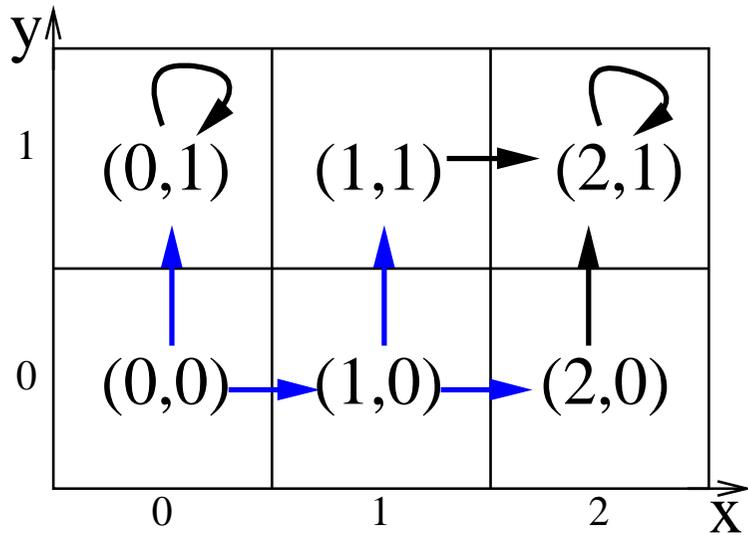
(x,y)	<u>Attracteur</u>
(0,0)	$(K_{x,\bar{y}}, K_y)=(2,1)$
(0,1)	$(K_x, K_y)=(0,1)$
(1,0)	$(K_{x,x\bar{y}}, K_y)=(2,1)$
(1,1)	$(K_{x,x}, K_y)=(2,1)$
(2,0)	$(K_{x,x\bar{y}}, K_{y,x})=(2,1)$
(2,1)	$(K_{x,x}, K_{y,x})=(2,1)$



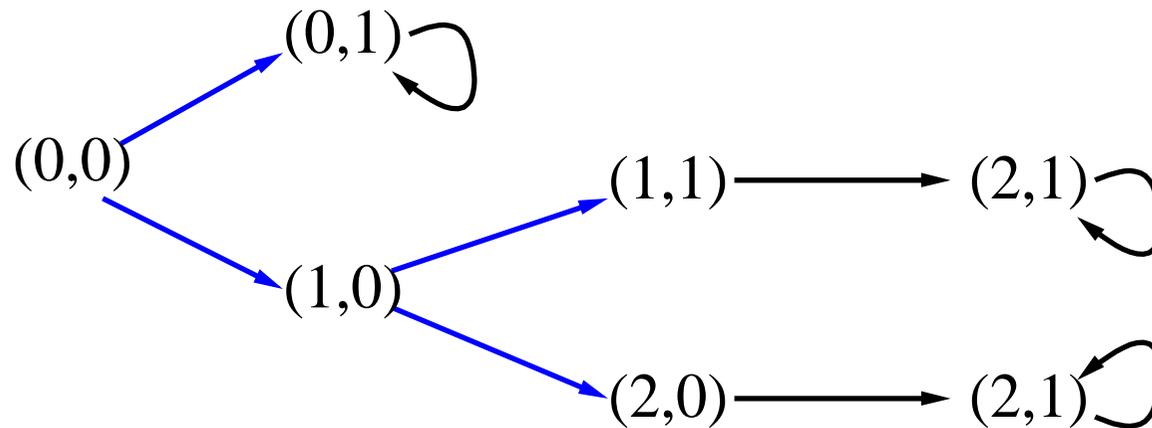
« désynchronisation » →
par unités de distance Manhattan



Arbre des traces



à partir d'un état initial :



Plan

1. Modèles & logiques *formels*, cohérence, validation
2. Formalisation des réseaux de régulation biologique
3. **Logique temporelle et Model Checking**
4. Exemple (J.Guespin) : *Pseudomonas aeruginosa*
 - Cohérence par Model Checking
 - Extraction automatique de plans d'expériences
 - Propriétés expérimentalement observables
5. Questions d'actualité

CTL = Computation Tree Logic

Atomes = les comparaisons : $(x=2)$ $(y>0)$...

Connecteurs : $(\varphi_1 \wedge \varphi_2)$ $(\varphi_1 \implies \varphi_2)$...

Modalités : composées de 2 lettres

<u>première lettre</u>	<u>seconde lettre</u>
A = for A ll path choices	X = ne X t state
E = there E xist a choice	F = for some F uture state
	G = for all future states (G lobally)
	U = U ntil

$AX(y = 1)$: le niveau de concentration de y vaut 1 dans tous les états suivant l'état initial.

$EG(x = 0)$: il existe au moins un chemin partant de l'état initial où x est identiquement nul.

Modalités de CTL

Atomes=comparaisons $(x=2)$, $(y>0)$... + **Connecteurs** standard

Prochain état :

$EX\varphi$: φ peut être vraie à l'étape d'après

$AX\varphi$: φ toujours vraie à l'étape d'après

Un jour dans le futur :

$EF\varphi$: φ peut être vraie plus tard

$AF\varphi$: φ toujours vraie plus tard

Modalités

Globalement :

$EG\varphi$: φ peut être un invariant à l'avenir

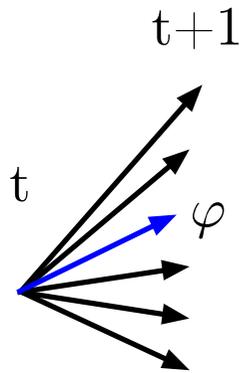
$AG\varphi$: φ toujours un invariant à l'avenir

Jusqu'à :

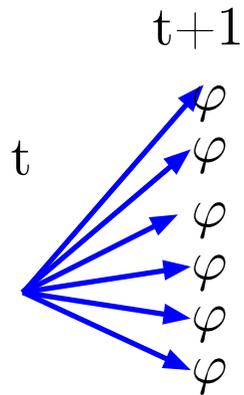
$E[\psi U \varphi]$ il existe un chemin avec ψ vrai jusqu'à ce que φ le soit

$A[\psi U \varphi]$ ψ reste toujours vrai jusqu'à ce que φ le soit

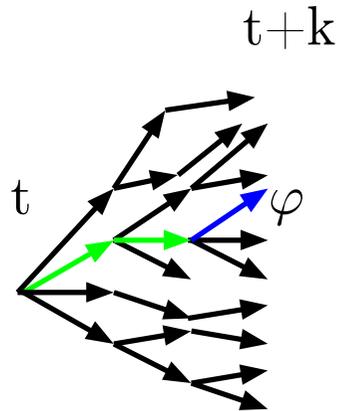
Sémantique des modalités sur les traces



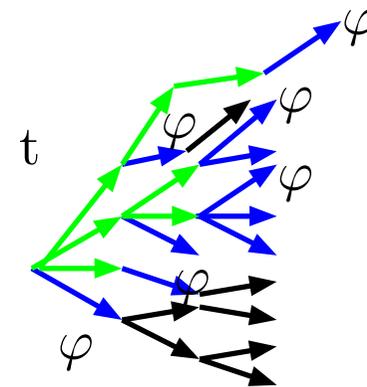
$EX\varphi$



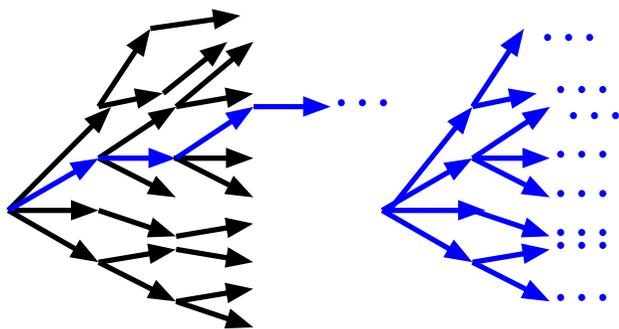
$AX\varphi$



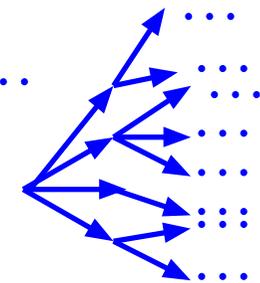
$EF\varphi$



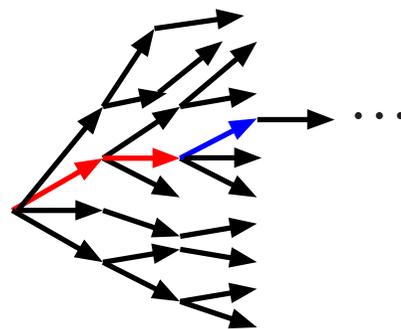
$AF\varphi$



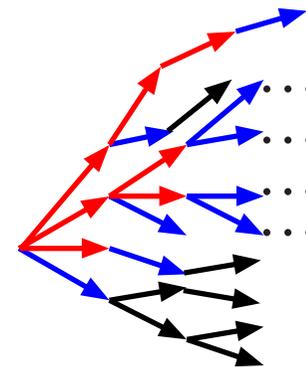
$EG\varphi$



$AG\varphi$



$E[\psi U \varphi]$

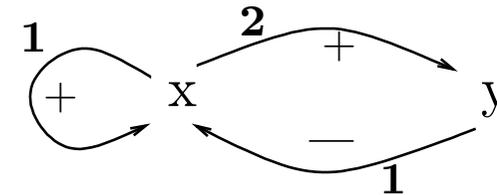


$A[\psi U \varphi]$

Formalisation de propriétés biologiques

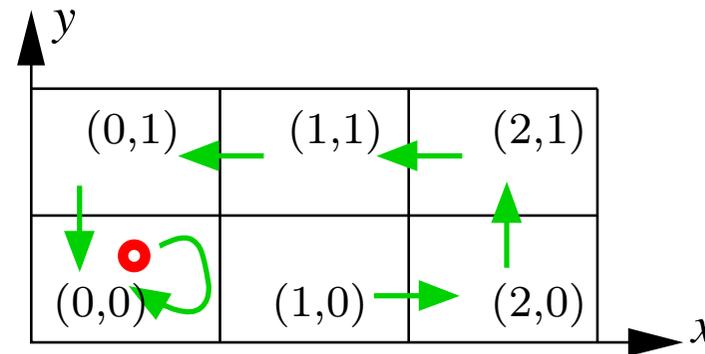
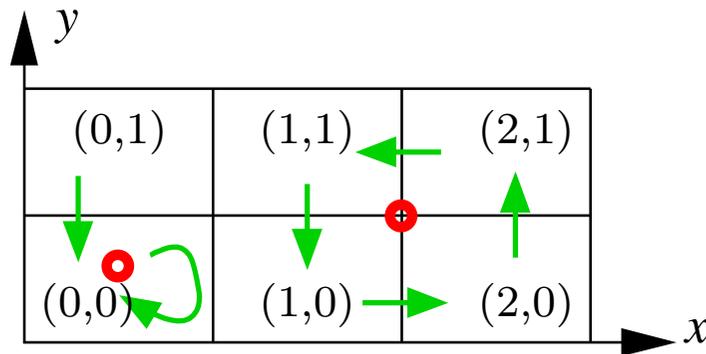
Propriétés souvent rencontrées :

« fonctionnalité » d'un sous-graphe



Rôle majeur des boucles de rétroaction

- positive : *bistationnarité* (nombre pair de —)
- négative : *homéostasie* (nombre impair de —)



Propriétés distinctives :

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

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

traduisent *la boucle positive est fonctionnelle* (dépendent des $K...$)

Formule = Lien modèles–expériences

Les formules ainsi construites sont valides ou invalides par rapport à un ensemble de traces donné partant d'un état donné

Elles peuvent être confrontées à l'ensemble des traces possibles du modèle théorique

Elles peuvent être confrontées à l'ensembles des expériences connues

Elles font donc le lien entre modèles et objets biologiques

Model Checking

Objet = calcule tous les états d'un graphe d'états qui satisfont une formule donnée : $\{ \eta \mid M \models_{\eta} \varphi \}$.

Idée n°1 : travailler sur le graphe d'états plutôt que sur les traces.

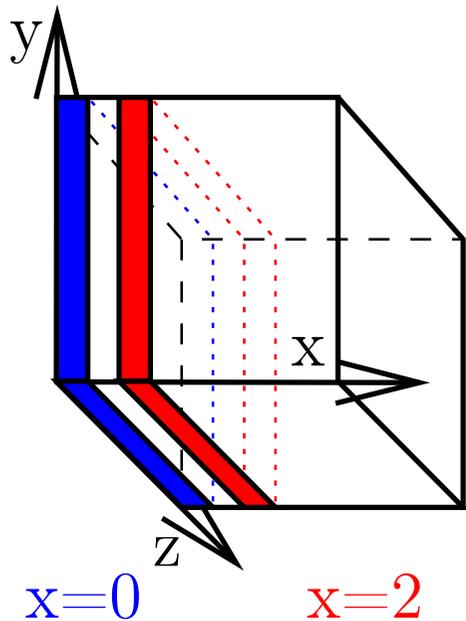
Idée n°2 : traiter les atomes de φ puis traiter « en remontant » les connecteurs et modalités.

Idée n°3 : (optimisation technique) factoriser les traitements *via* des BDD.

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

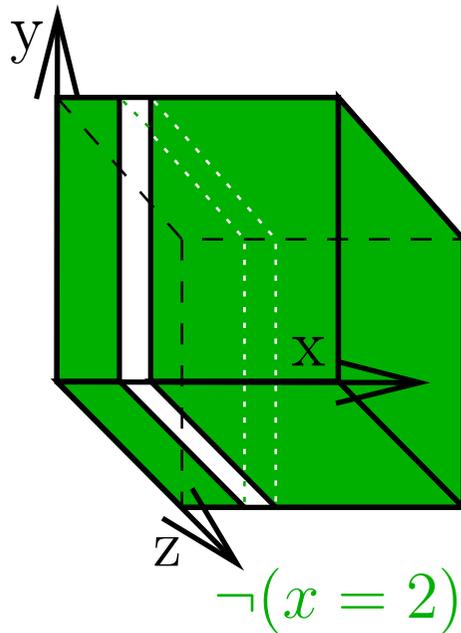
Obsession : *parcourir la matrice des états le moins souvent possible*

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



$x=0$

$x=2$



$\neg(x = 2)$

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

il faudrait **parcourir** tous les chemins partant d'une case verte et voir s'ils restent verts : visites multiples de trop de cases.

Astuce : $AG(\neg(x = 2))$ équivaut à $\neg EF(x = 2)$

on part des cases rouges et on remonte les transitions.

Méthode par Model Checking et test

- Tracer tous les réseaux de régulation avec toutes les combinaisons de seuils crédibles
- Formaliser en CTL les propriétés comportementales connues et les hypothèses (boucles fonctionnelles, etc.)
- Engendrer automatiquement tous les graphes de régulation selon toutes les valeurs possibles des paramètres ($K...$)
N.B. : \exists réductions du domaine par analyse statique
- Les confronter par Model-Checking, *retenir les modèles rescapés*
- S'il n'en reste aucun alors remettre en cause les hypothèses

S'il en reste passer à la question 2. : chercher les conséquences révélatrices qui peuvent donner lieu à des expériences humides

Actuellement la détermination des paramètres $K...$ est un goulot d'étranglement de la construction de modèles

Plan

1. Modèles & logiques *formels*, cohérence, validation
2. Formalisation des réseaux de régulation biologique
3. Logique temporelle et Model Checking
4. Exemple (J.Guespin) : *Pseudomonas aeruginosa*
 - Cohérence par Model Checking
 - Extraction automatique de plans d'expériences
 - Propriétés expérimentalement observables
5. Questions d'actualité

Un exemple « simple »

Modification de phénotype, terminologie :

modification génétique : héritable et non réversible (mutation)

modification épigénétique : héritable mais réversible

adaptation : non héritable et réversible

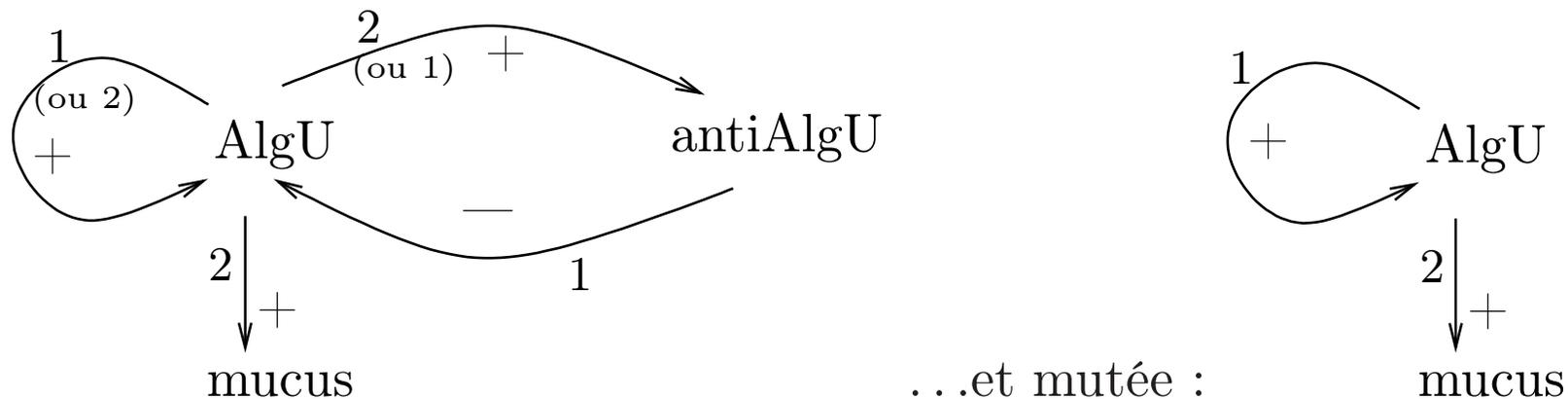
Les questions biologiques (Janine Guespin) :

la **mucoïdie** et/ou la **cytotoxicité** chez la bactérie *Pseudomonas aeruginosa* sont-elles de nature épigénétique ?

[→ mucoviscidose]

Mucoïdie chez *Pseudomonas aeruginosa*

Pseudomonas aeruginosa sauvage : (J.Guespin, M.Kaufman)



...et mutée :

Hypothèse épigénétique (i.e. sans mutation) =

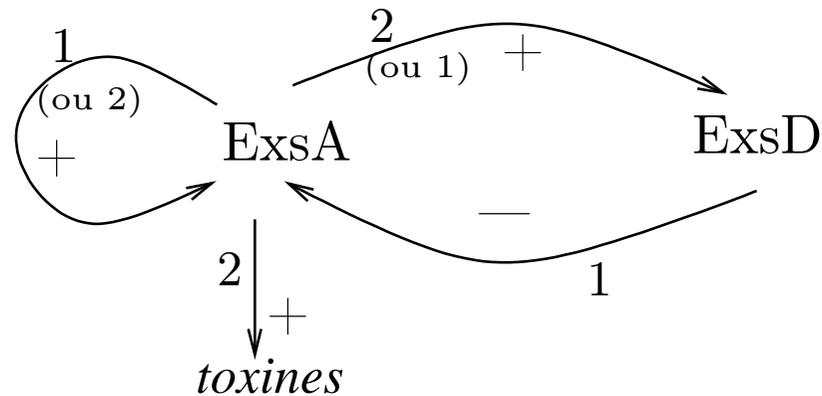
→ Le cycle positif est fonctionnel malgré le cycle négatif, avec un état non mucoïde et l'autre mucoïde.

→ Un signal extérieur (produit par le poumon malade) pourrait éventuellement faire passer AlgU de l'état bas à l'état haut.

→ La pression de sélection favorise *ultérieurement* les mutants en milieu muqueux. \implies Nouvelles perspectives de thérapie.

Cytotoxicité chez *P. aeruginosa*

(Janine Guespin)



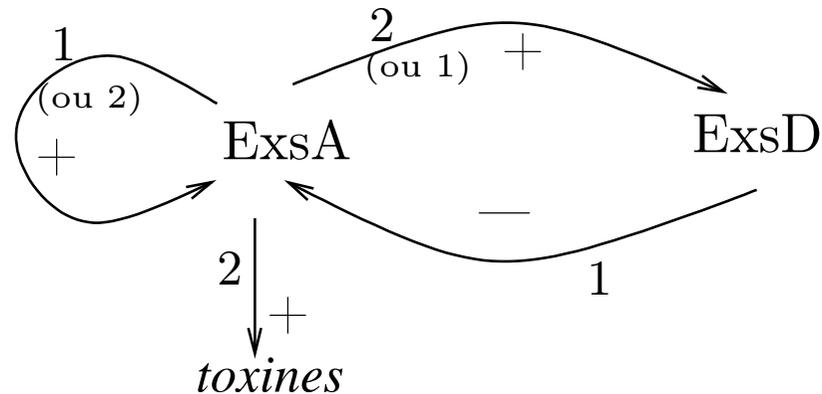
Hypothèse épigénétique =

→ Le cycle positif est fonctionnel malgré le cycle négatif, avec un état non cytotoxique et l'autre cytotoxique.

→ Un signal extérieur pourrait éventuellement faire passer ExsA de l'état bas à l'état haut.

⇒ Nouvelles perspectives de thérapie.

Cohérence de l'hypothèse épigénétique



Les 2 états stables :

$$(E_{xsA} = 2) \implies AXAF(E_{xsA} = 2)$$

$$(E_{xsA} = 0) \implies AG(\neg(E_{xsA} = 2))$$

Question 1, cohérence : prouvée par *Model Checking*

→ 8 modèles parmi 648, extraits par notre logiciel dédié SMBioNet

Question 2, et *in vivo*? ...

Validation de l'hypothèse épigénétique

Question 2 = valider *in vivo* la stabilité des deux états

État non cytotoxique : $(\text{ExsA} = 0) \implies AG(\neg(\text{ExsA} = 2))$

Une bactérie ayant son niveau basal de ExsA ne deviendra pas cytotoxique spontanément : validé au quotidien

État cytotoxique : $(\text{ExsA} = 2) \implies AXAF(\text{ExsA} = 2)$

Hypothèse de travail :

on peut porter ExsA à saturation, pas le mesurer.

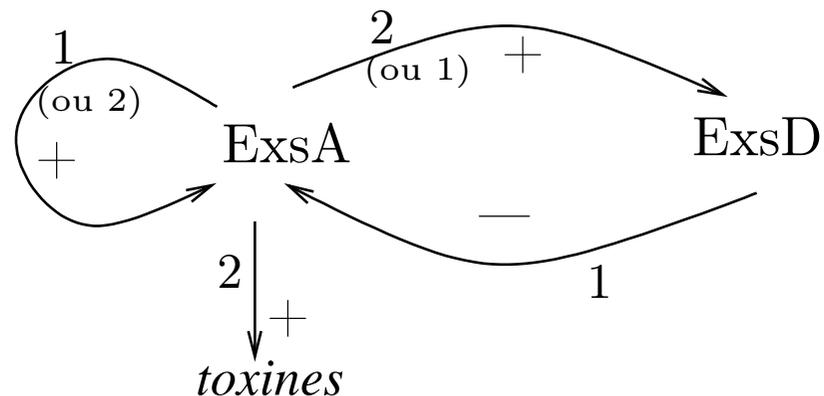
Plan d'expérience : *pulse sur ExsA puis après une phase transitoire tester si la production de toxines persiste*

(\iff vérifier une hysteresis)

On peut générer les plans d'expérience automatiquement

Expérimenter $(\text{ExsA}=2) \implies AXAF(\text{ExsA}=2)$

$\text{ExsA} = 2$ n'est pas directement vérifiable mais $\text{toxines} = 1$ l'est.



Lemme : $AXAF(\text{ExsA} = 2) \iff AXAF(\text{toxines} = 1)$

(... preuve par ordinateur ...)

→ **Expérimenter : $(\text{ExsA} = 2) \implies AXAF(\text{toxines} = 1)$**

$$(ExsA = 2) \implies AXAF(toxines = 1)$$

Karl Popper :

valider = tenter de réfuter
donc A Faux ne sert à rien
 donc commencer par un pulse

$A \implies B$	<i>Vrai</i>	<i>Faux</i>
<i>Vrai</i>	Vrai	Faux
<i>Faux</i>	Vrai	Vrai

Le pulse permet d'atteindre l'état initial $ExsA = 2$.

Sinon, il aurait fallu établir un **lemme** :

$$(ExsA = 2) \iff (qqchose\ contr\hat{o}lable)$$

Forme générale d'un test :

$$(\underline{qqchose\ atteignable}) \implies (\underline{qqchose\ observable})$$

Plan

1. Modèles & logiques *formels*, cohérence, validation
2. Formalisation des réseaux de régulation biologique
3. Logique temporelle et Model Checking
4. Exemple (J.Guespin) : *Pseudomonas aeruginosa*
 - Cohérence par Model Checking
 - Extraction automatique de plans d'expériences
 - Propriétés expérimentalement observables
5. Questions d'actualité

Les techniques de test de logiciels

Problème similaire = un logiciel répond-il à sa spécification ?

Infinité de scénarii de test possibles, en sélectionner de révélateurs

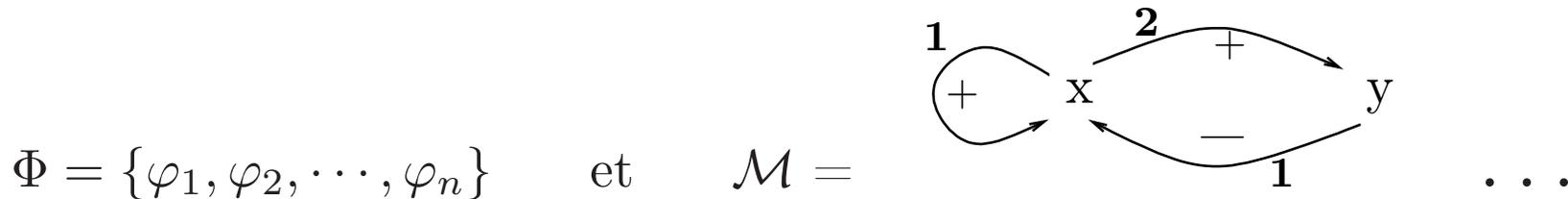
Solution = découper en domaines de scénarii

Comportements supposés « uniformes » au sein d'un domaine

- le *dépliage de formules sert à découper les domaines*
- approche **probabiliste** : on déplie peu,
peu de domaines (mais gros),
tirage probabiliste de nombreux tests dans chaque domaine
- approche **déterministe** : on déplie beaucoup,
domaines petits,
choix d'un seul test par domaine

Plus de 99% des bugs sont détectables automatiquement

Sélection de plans d'expériences



Formules observables Φ_{obs} :

$\{ \psi \mid \Phi, \mathcal{M} \models \psi \text{ et } \psi \text{ de la forme } \ll \textit{atteignable} \implies \textit{observable} \gg \}$

Problème : Φ_{obs} est infini

→ choisir des ψ « révélateurs » dans Φ_{obs}

P.aeruginosa : coup de chance, il existe 2 formules observables

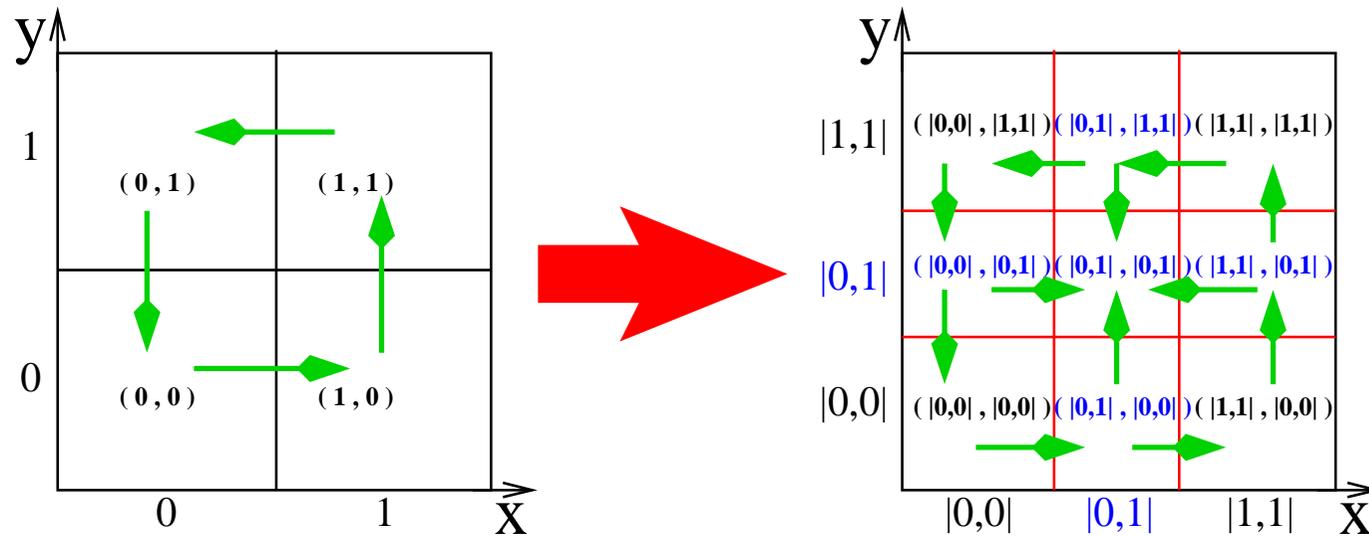
$$\psi_1, \psi_2 \in \Phi_{obs} \text{ telles que } \{\psi_1, \psi_2\} \models \Phi_{obs}$$

Solution générale informatique : les techniques de dépliage

(\approx raisonnement par cas) devraient permettre d'expliciter les

hypothèses faites lorsqu'on se limite à un nombre fixé d'expériences

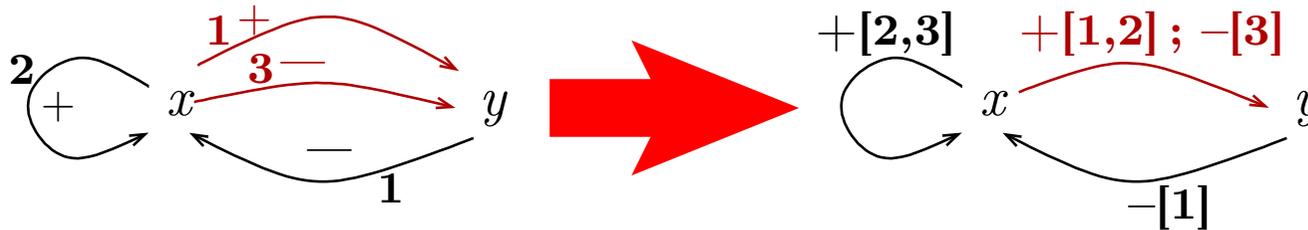
Explicit singular states : SMBioNet



Main idea : no new parameters $K_{x,\omega}$, but the *image* of a variable owning « singular resources » is redefined accordingly ; desynchronization as usual.

SMBioNet computes all the sets of parameters which fulfill the specified properties (functional loop, known steady state or any given CTL property) *via* the SMV model checker.

Semantics with intervals and neutral states



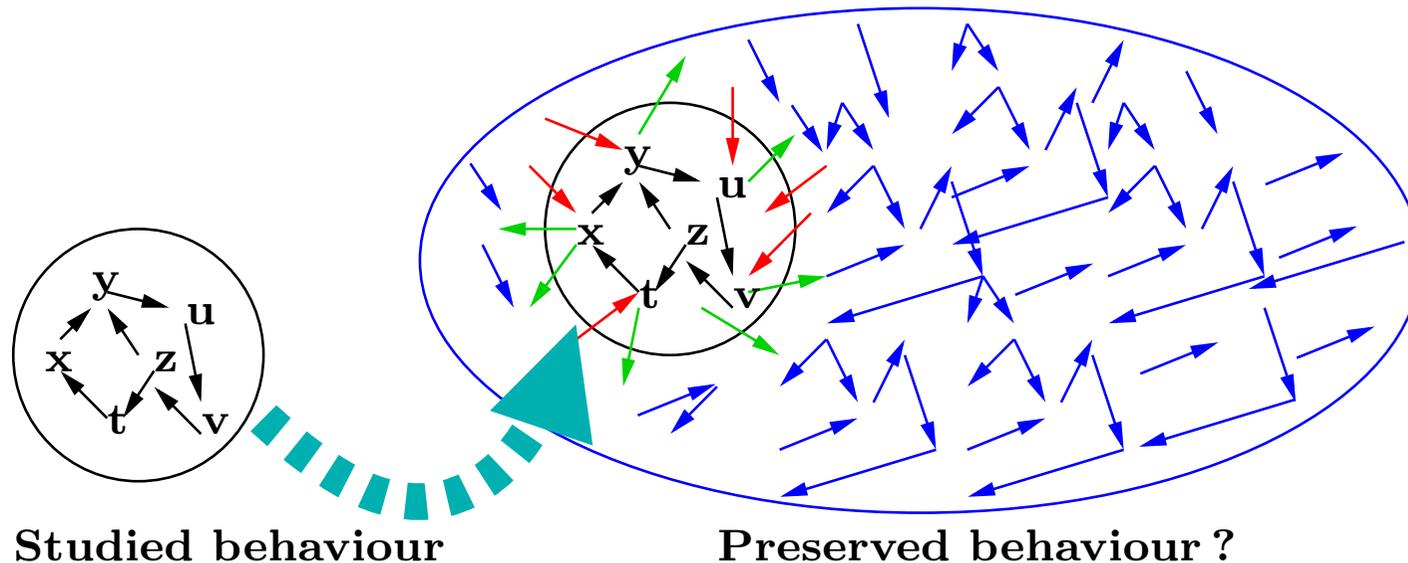
Some states can be neutral (e.g., 0) \implies modifies the formalism

Inhibition is not the negation of *Induction*

Parameters $K_{y,\omega,\sigma}$ $\left\{ \begin{array}{l} \omega : \text{set of active inducers} \\ \sigma : \text{set of active inhibitors} \end{array} \right.$

The BIOFOR prototype computes the state graph of a gene regulatory network and graphically highlights configurations of interests such as steady states, functional loops, ...

Embeddings of Regulatory Networks



→ require state folding technics to translate behaviour

→ Main theorem : necessary and sufficient condition on the *local* dynamics of the « input frontier »

Offers a methodology to identify interesting subnetworks

Modélisation multi-niveaux

- Plusieurs techniques de modélisation
- Plusieurs vues d'un même objet biologique
- Plusieurs niveaux d'organisation biologique
- Plusieurs niveaux de raffinement des modèles
- Connaissances « à trous »
 - ⇒ faire feu de tous bois
 - ⇒ approches bottom-up ou top-down impossibles
- Nombreux phénomènes d'émergence
- Création dynamique d'objets biologiques nouveaux
- Forte limitation de « l'observabilité » biologique
- Problème des artefacts induits par la technique de modélisation

Modélisation v.s. Simulation

- Ergonomie de la visualisation
- Rendre les modèles exécutables
- Quels résultats présenter ?
- Sous quelle forme ?
- Dangers des implicites faux, induits par l'interface
- Bien distinguer et montrer
 - . les connaissances biologiques
 - . les hypothèses dûes au modèle
 - . les déductions « sûres »
 - . les hypothèses dûes à la simulation
(e.g. choix « aléatoires » lors de l'exécution)
 - . etc.

Quelques slogans

- Les *propriétés* comportementales (Φ) sont aussi importantes que les modèles (\mathcal{M})
- La modélisation n'a de sens qu'accompagnée de l'*atteignabilité* et de l'*observabilité* considérées
- Plus elles permettent d'explorer le modèle, plus le risque de *réfutation* est grand et plus la modélisation a d'intérêt (Popper)
- Rasoir d'*Occam* : en conséquence plus le modèle est simple (peu de paramètres non observables), plus il a d'intérêt

Les **méthodes formelles** (*syntaxe/sémantique/preuves*) facilitent l'*abstraction* et donc la simplification des modèles

- Elles assurent la *cohérence* de la modélisation
- Elles permettent d'instrumenter la *validation* des modèles
- Elles bénéficient d'un corpus de 30 ans de recherches en informatique