On the use of temporal formal logic to model gene regulatory networks

Gilles Bernot and Jean-Paul Comet

Laboratoire I3S, UMR 6070 UNS-CNRS
Algorithmes-Euclide-B
2000, route des Lucioles
B.P. 121
F-06903 Sophia Antipolis CEDEX
{bernot,comet}@unice.fr

Abstract. Modelling activities in molecular biology face the difficulty of prediction to link molecular knowledge with cell phenotypes. Even when the interaction graph between molecules is known, the deduction of the cellular dynamics from this graph remains a strong corner stone of the modelling activity, in particular one has to face the parameter identification problem. This article is devoted to convince the reader that computers can be used not only to simulate a model of the studied biological system but also to deduce the sets of parameter values that lead to a behaviour compatible with the biological knowledge (or hypotheses) about dynamics. This approach is based on formal logic. It is illustrated in the discrete modelling framework of genetic regulatory networks due to René Thomas.

1 Introduction: modelling gene regulatory networks

Since the advent of molecular biology, biologists have to face increasing difficulties of prediction to link molecular knowledge with cell phenotypes. The belief that the sequencing of genomes would rapidly open the door to a personalized medicine has been confronted at first to the necessity of annotating finely genomes, then to the difficulty to deduce the structure(s) of proteins, then to the huge inventory of interactions that constitute biological networks, and so on. In the same way, we have to face now the fact that the knowledge of an interaction graph does not make it possible to deduce the cellular dynamics. Indeed, interaction graphs are of static nature in the same way as genetic sequences, and it turns out that a large number of parameters, which are unknown and not easily measurable, control the dynamics of interactions.

Moreover, *combined* interactions (and especially feedback circuits in an interaction graph) result in several possible behaviours of the system, qualitatively very different. Even with only two genes the situation is far from simple. Let us consider for example the interaction graph of Figure 1.

This simple graph contains 2 circuits, whose intersection is the gene x. The left hand side circuit is said *positive*:

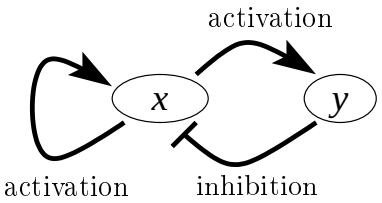

Fig. 1. A simple interaction graph containing a positive circuits and a negative one.

- If, from an external stress, the concentration level of the protein coded by gene x grows up, then, at a certain threshold, it will favour the expression of x, producing in turn more x proteins even if the external stress has disappeared.
- On the contrary, if the concentration level of the protein of gene x is low, then it will not favour the expression of x, and the concentration level of the x protein can stay at a low level.

More generally, a positive circuit in a gene interaction network is a circuit that contains an even number of inhibitions, and a positive circuit favours the existence of 2 stable states [28, 19, 24, 10, 27, 21] (respectively high level and low level of expression of x).

The right hand side circuit is said *negative* because it contains an odd number of inhibitions:

- If the concentration level of the protein coded by gene x grows up, then it will favour the expression of gene y, which will in turn inhibit gene x, resulting in a decreasing of the x protein concentration.
- Conversely, a low concentration level of the x protein shall decrease the expression level of gene y, which shall become unable to inhibit x, resulting in a higher expression level of $x\ldots$ and the process will start again.

More generally, a negative circuit favours *homeostasy*: oscillations which can either be damped towards a unique stable state or sustained towards a limit cycle surrounding a unique unstable equilibrium state of the biological system.

The two circuits of Figure 1 are consequently competitors: *do we get one or two stable states? shall we observe oscillations?* and if so, *do we get oscillations around a unique stable state or around two stable states?* These predictions entirely depend on parameters that control the strength of the activation or inhibition arrows of the interaction graph (see Section 2.3).

Most of the time, mathematical models are used to perform simulations using computers. Biological knowledge is encoded into ordinary differential equations (ODE) for instance, and many parameters of the system of ODEs are *a priori* unknown, a few of them being approximately known. Many and many simulations are performed, with different values for the parameters, and the behaviour observed *in silico* is compared with the known *in vivo* behaviour. This process, by trial and error, makes it possible to propose a robust set of parameters

(among others) that is compatible with the biological observations. Then, several additional simulations that simulate novel situations can predict interesting behaviours, and suggest new biological experiments.

The goal of this article is to convince the reader that "brute force simulations" are not the only way to use a computer for gene regulatory networks. A computer can do more than computations. A computer manipulates symbols. Consequently, using deduction rules, a computer can perform *proofs* within adequate *logics*. One of the main advantages of logics is that they exhaustively manipulate sets of models, and exhaustively manage the subset of all models that satisfy a given set of properties. More precisely, a logic provides three well established concepts:

- a *syntax* that defines the properties that can be expressed and manipulated (these properties are called formulas),
- a *semantics* that defines the models under consideration and the meaning of each formula according to these models,
- *deduction rules* or *model checking* from which we get algorithms in order to prove if a given model (or a set of models) satisfy a formula (or a finite set of formulas).

Logic can thus be used to manipulate, in a computer aided manner, the set of *all* the models that satisfy a given set of known properties. Such an exhaustive approach avoids focusing on one model, which can be "tuned" *ad libitum* because it has many parameters and few equations. Consequently it avoids focusing on a model which is non predictive. Logic also helps studying the ability to refute a set of models with the current experimental capabilities, it also brings useful concepts such as observationally equivalent models, and so on. More generally formal methods are helpful to assist biologists in their reasonnings [4, 7, 22, 1, 18].

In this article, we provide a survey of the "SMBioNet method" whose purpose is to elucidate the behaviour of a gene interaction graph, to find all the possible parameter values (if any), and to suggest suitable experiments to validate or to refute a given biological hypothesis. In the next section, we remind the approach of René Thomas [29] to obtain discrete models (with finite sets of possible states) for gene regulatory networks. We take as a pedagogical example the well known lactose operon in *E. coli*. In Section 3, we show how temporal logic and more precisely CTL can be used to properly encode biological properties. In Section 4, we show how temporal logic can be used to guide the process of gene network elucidation.

2 Discrete framework for gene regulatory networks

2.1 A classical example

In this article, we consider the system of the lac operon which plays a crucial role in the transport and metabolism of lactose in *Escherichia coli* and some

other enteric bacteria [11]. Lactose is a sugar which can be used as a source of carbon mandatory for mitosis. This system allows to switch on the production of enzymes allowing the metabolism of carbon only if lactose is available and no other more readily-available energy sources are available (e.g. glucose).

The lactose operon and its associated biological system. The operon consists of three adjacent structural genes, a promoter, a terminator, and an operator. The lac operon is regulated by several factors including the availability of glucose or of lactose. When lactose is absent in the current environment, a repressor protein maintains the expression of the operon at its basal level. In presence of lactose, it enters into the cell thanks to a protein named permease which is coded by the operon itself. The lactose proteins have affinity to the repressor proteins, form complexes with them leading first to a decreasing of the concentration of free repressor and thus to the activation of the operon. Consequently, the permease concentration increases, the lactose enters more efficiently into the cell, maintaining the low concentration of free repressor. These interactions form then a positive feedback loop on the intracellular lactose (left part of Figure 2).

Another protein coded by the operon plays also a role in the carbon metabolism: the enzyme galactosidase. It is responsible for the degradation of the lactose in order to transform the lactose into carbon. Thus the increasing of the intracellular lactose leads to an increasing of the galactosidase, then to the decreasing of the intracellular lactose. These interactions form then a negative feedback loop on the intracellular lactose (right part of Figure 2).

Moreover, when glucose is present, it inhibits indirectly the transcription of the operon (*via* an indirect inhibition of cAMP (cyclic Adenosine MonoPhospate) which forms with CAP (Catabolite gene Activator Proteins) the complex responsible for the transcription of the operon. Thus this alternative pathway of the carbon metabolism is inhibited.

To summarize, the intracellular lactose is subject to two influences which are contradictory one to another. The positive feedback loop attempts to keep the high concentration of intracellular lactose whereas the negative feedback loop attempts to decrease this concentration, as shown in Figure 2.

Biological questions drive modelling abstractions. The modelling of a biological system often means to construct a mathematical objet that mimics the behaviours of the considered biological system. The elaborated model is often built according to a particular knowledge on the system (interactions, structuration, published behaviours, hypotheses...), and this knowledge is often not complete: for example one may improperly focus on a given subsystem. Thus the modelling process presented in this paper proposes to construct models according a particular point of view on the biological system. This point of view may turn out to be inconsistent and the modelling process should be able to point out inconsistencies leading to reconsider the model(s).

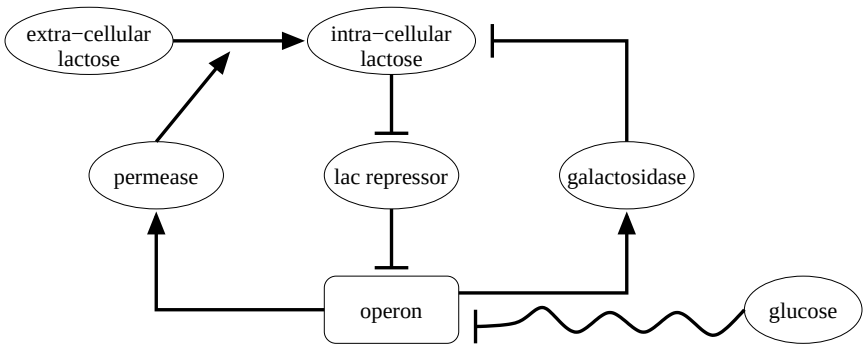

Fig. 2. Schematic representation of the lac operon system in *Escherichia coli*

The construction of a model aims at studying a particular behaviour of the system. Each facet of the system corresponds to a particular partial model. When this facet of the system is understood, the modeller can throw away the current model in order to go further in the understanding of the system. In other words, the point of view of the modeller evolves during the modelling process, leading to refinements or re-buildings of the model.

In such a perspective, as the construction of the model is based on a set of biological facts and hypotheses, it becomes possible to construct a model only to test different hypotheses, to apprehend the consequences of such hypotheses and possibly to refute some of them. Last but not the least, these models can be used in order to suggest some experiments.

For example, if biologists want to put the spot on the use of lactose, then we can adopt a modelling point of view that implicitly assumes the absence of glucose. Then glucose does not belong to the model any more. Moreover, even if it may seem surprising, the lacI repressor can be suppressed because the inhibition of cellular lactose on lacI and the inhibition of the lacI repressor on the operon are known to be always functional. These successive repressions can consequently be abstracted by a unique direct *activation* from cellular lactose to the operon.

Depending on the studied hypotheses, some additional simplifications of the model are possible:

- If the hypothesis does not refer explicitly to the galactosidase, then we can abstract galactosidase in a similar manner than for the repressor, see Figure 3.
- If the hypothesis does not refer explicitly to the permease, then we can abstract permease as in Figure 4.

In the sequel of this article we will consider the interaction schema of Figure 4.

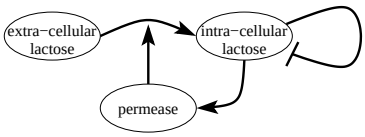

Fig. 3. Abstraction of the lac operon system when focusing on intra-cellular lactose and permease.

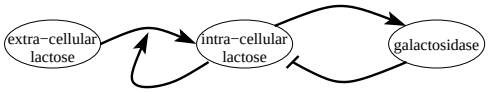

Fig. 4. Abstraction of the lac operon system when focusing on intra-cellular lactose and galactosidase.

2.2 Gene interaction networks

Discretization. When modelling gene interactions, threshold phenomena observed in biology [16] constitute one of the key points for the comprehension of the behaviour of the system. Combined with the additional *in vivo* phenomenon of macromolecule degradation, the interaction curves get a sigmoidal shape (e.g. Hill functions) [6], see Figure 5. Then, it becomes clear that for each interaction, two qualitative situations have to be considered: the regulation is effective if the concentration of the regulator is above the threshold of the sigmoid, and conversely, it is ineffective if the concentration of the regulator is below the threshold.

When the product of a gene regulates more than one target, more than two situations have to be considered. For example, Figure 5 assumes that u is a gene product which acts positively on v and negatively on w; each curve being the concentration of v (resp. w) with respect to the concentration of u; after a sufficient delay for u to act on v (resp. w). Obviously, three regions are relevant in the different levels of concentration of u:

- In the first region u acts neither on v nor on w,
- In the second region, u acts on v but it still does not act on w:
- In the last region, u acts both on v and w:

The sigmoid nature of the interactions shown in Fig. 5 is almost always verified and it justifies this discretization of the concentration levels of u: three abstract levels (0, 1 and 2) emerge corresponding to the three previous regions and constitute the only relevant information from a qualitative point of view[1].

The generalization is straightforward: if a gene acts on n targets, at most $n + 1$ abstract regions are considered (from 0 to n). Less abstract levels are possible when two thresholds for two different targets are equal.

[1] Atypic behaviours on the thresholds can also be studied, see for example [8].

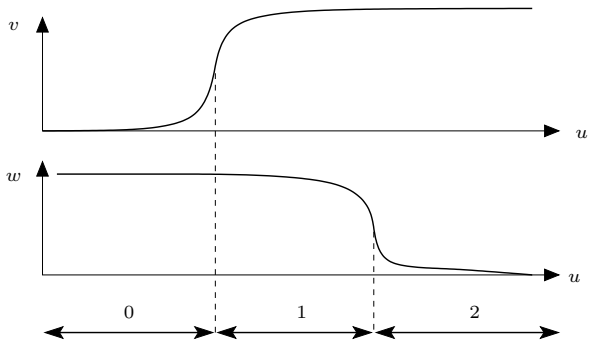

Fig. 5. Discretization of concentration of a regulator with 2 targets.

Gene regulatory graphs. A *biological regulatory graph* is defined as a labelled directed graph. A vertex represents a variable (which can abstract a gene and its protein for instance) and has a boundary which is the maximal value of its discrete concentration level. Each directed edge $u \to v$ represents an action of u on v. The corresponding sigmoid can be increasing or decreasing (Figure 5), leading respectively to an activation or an inhibition. Thus each directed edge $u \to v$ is labelled with an integer threshold belonging to $[0, b_u]$ and a sign: $+$ for an activation of v and $-$ for an inhibition of v.

Definition 1. *A* biological regulatory graph *is a labelled directed graph* $G = (V, E)$ *where:*

- *each vertex* v *of* V*, called variable, is provided with a boundary* $b_v \in \mathbb{N}^*$ *less or equal to the out-degree of* v *in* G*; except when the out-degree is 0 where* $b_v = 1$*;*
- *each edge* $u \to v$ *of* E *is labelled with a couple* $(t; \epsilon)$ *where* t*, called threshold, is an integer between 1 and* b_u *and* $\epsilon \in \{-, +\}$*.*

The schematic figure 4 is too informal to represent a *biological regulatory graph*: the edge modelling the auto-regulation of the intra-cellular lactose does not point on a variable but on an edge and, moreover, the thresholds are missing.

To construct from the figure 4 a *biological regulatory graph*, we modify the edge modelling the auto-regulation of the intra-cellular lactose: its target becomes directly the intra-cellular lactose, see Figure 6. Moreover, three different rankings of thresholds can be considered : cases A, B or C.

Gene regulatory networks. The discretization step allows one to consider only situations which are qualitatively different: if an abstract level changes, there exists at least one interaction which becomes effective or ineffective. To go further, one has to define what are the possible evolutions of each variable under some effective regulations. Assuming that $u_1 \dots u_n$ have an influence on v (entering

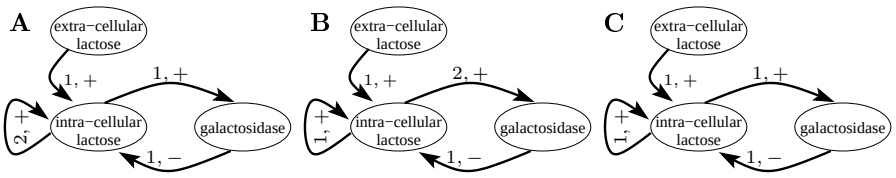

Fig. 6. Three different biological regulatory graphs of the operon lactose system (note the different values of interaction thresholds).

arrows $u_i \to v$), toward which concentration level is v attracted? This level depends on the set of active regulators, which evolves with time: at a given time, only some of them pass the threshold.

For example in Figure 6-A, the level toward which intra-cellular lactose is attracted only depends on the presence of extra-cellular lactose (*i.e.* has a level greater or equal to 1), the presence of itself (*i.e.* has a level greater or equal to 2) and the *absence* of galactosidase (*i.e.* has a level strictly less than 1). Indeed the absence of an inhibitor is equivalent to the presence of an activator, from the symmetry of sigmoids.

These "target" concentration levels are defined by parameters, denoted by $k_{v,\omega}$, where ω is a subset of regulators. *Biological regulatory networks* are biological regulatory graph (Definition 1) together with these parameters $k_{v,\omega}$.

Definition 2. *A* biological regulatory network *is a couple* $\mathcal{R} = (G, \mathcal{K})$ *where* $G = (V, E)$ *is a biological regulatory graph, and* $\mathcal{K} = \{k_{v,\omega}\}$ *is a family of integers such that*

- v *belongs to* V,
- ω *is a subset of* $G^{-1}(v)$*, the set of predecessors of* v *in the graph* G*, and will be called a set of resources of* v,
- $0 \leq k_{v,\omega} \leq b_v$

Intuitivelly, the parameter $k_{v,\omega}$ describes the behaviour of variable v when all variables of ω act as a resource of v (a *resource* being the presence of an activator or the absence of an inhibitor).

Most of the time, we consider an additional monotony condition called the Snoussi condition [25]:

$$\forall v \in V, \forall \omega, \omega' \in G^{-1}(v), \omega \subset \omega' \Rightarrow k_{v,\omega} \leq k_{v,\omega'}$$

In other words, values of parameters never contradict the quantity of resources.

For the running example, the variable *intra-cellular lactose*, noted *intra* in the sequel, is regulated by two activators, and by one inhibitor. This variable can thus be regulated by $2^3 = 8$ different subsets of its inhibitors/activators. In the same way, 2 parameters have to be given for the variable *galactosidase*, noted g in the sequel. To sum up, 10 parameters have to be given

$$\mathcal{K} = \left\{ \begin{array}{c} k_{g,\emptyset}, k_{g,\{intra\}}, k_{intra,\emptyset}, k_{intra,\{extra\}}, k_{intra,\{g\}}, k_{intra,\{extra,g\}}, \\ k_{intra,\{intra\}}, k_{intra,\{intra,extra\}}, k_{intra,\{intra,g\}}, k_{intra,\{intra,extra,g\}} \end{array} \right\}.$$

These parameters control the dynamics of the model since they define how targets evolve according to their current sets of ressources.

2.3 Dynamics of gene networks: State graphs.

At a given time, each variable of a regulatory network has a unique concentration level. The *state* of the biological regulatory network is the vector of concentration levels (the coordinate associated with any variable u is an integer belonging to the interval from 0 to the boundary b_v).

According to a given state, the *resources* of a variable are the regulators that help the variable to be expressed. The set of resources of a variable is constituted by all activators whose level is above the threshold of activation and all the inhibitors whose level is below the threshold.

Resources are used to determine the evolution of the system. At a given state, each variable is attracted by the corresponding parameters $k_{v,\omega}$ where ω is its set of resources. The function that associates with each state the vector formed by the corresponding $k_{v,\omega}$ is an endomorphism of the state space. Table 1 defines the endomorphism for the example of Figure 6-A when extra-cellular lactose is present (parameter values are given in the caption).

$extra$	$intra$	g	$\omega(intra)$	$\omega(g)$	$k_{intra,\omega(intra)}$	$k_{g,\omega(g)}$
1	0	0	$\{extra, g\}$	$\{\}$	2	0
1	0	1	$\{extra\}$	$\{\}$	0	0
1	1	0	$\{extra, g\}$	$\{intra\}$	2	1
1	1	1	$\{extra\}$	$\{intra\}$	0	1
1	2	0	$\{extra, intra, g\}$	$\{intra\}$	2	1
1	2	1	$\{extra, intra\}$	$\{intra\}$	1	1

Table 1. Partial state table for the figure 6-A. Here only state with extra-cellular lactose are considered. Values of parameters are : $k_{intra,\{extra\}} = 0$, $k_{intra,\{extra,intra\}} = 1$, $k_{intra,\{extra,g\}} = 2$, $k_{intra,\{extra,intra,g\}} = 2$, $k_{g,\emptyset} = 0$ and $k_{g,\{intra\}} = 1$.

Such a table can be represented by a *asynchronous state graph* in which each state has a unique successor: the state towards which the system is attracted, see the left part of Figure 7 where *extra* is supposed to be equal to 1. In this example, when the system is in the state $(1, 1, 0)$, it is attracted towards the state $(1, 2, 1)$. Variables *intra* and g are both attracted toward different values. The probability that both variables pass through their respective thresholds at the same time is negligible *in vivo*, but we do not know which one will be passed first. Accordingly we replace such a diagonal transition by the collection of the transitions which modify only one of the involved variables at a time. For example, transition $(1, 1, 0) \rightarrow (1, 2, 1)$ is replaced by the transitions $(1, 1, 0) \rightarrow (1, 2, 0)$ and $(1, 1, 0) \rightarrow (1, 1, 1)$: the first one corresponds to the case where the

variable *intra* evolves first whereas the second one corresponds to the case where the variable g evolves first, see the right part of Figure 7.

An arrow of length greater or equal to 2 would imply a variable which increases its concentration level abruptly and jumps several thresholds. For our example, when the system is in the state $(1,0,0)$, it is attracted towards the state $(1,2,0)$. Since the concentration varies continuously, independently of whether it varies rapidly or not, real transitions should only address neighbor states. Thus, transition $(1,0,0) \rightarrow (1,2,0)$ is replaced by the transition $(1,0,0) \rightarrow (1,1,0)$, see the right part of Figure 7.

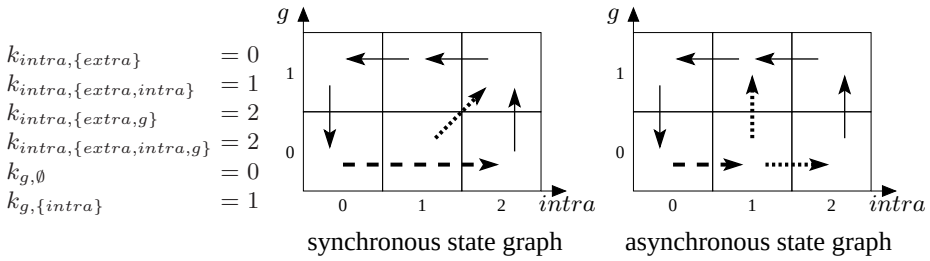

Fig. 7. From the values of parameters to the asynchronous state graph.

.

The value of the parameter $k_{v,\omega}$ (where ω is the set of resources of v at the current state η), indicates how the expression level of v can evolve from the state η. It can increase (respectively decrease) if the parameter value is greater (respectively smaller) than the current level of the variable v. The expression level must stay constant if both values are equal. Formally:

Definition 3. *The* asynchronous state graph *of the biologigical regulatory network $\mathcal{R} = (G, \mathcal{K})$ is defined as follow:*

- *the set of vertices is the set of states $\Pi_{v \in V}[0, b_v]$*
- *there is a transition from the state $n = (n_1, \ldots, n_{|V|})$ to $m = (m_1, \ldots, m_{|V|})$ iff*

$$\begin{cases} \exists! i \text{ such that } m_i \neq n_i \\ m_i = (n_i \triangleright k_{i,\omega_i(n)}) \end{cases} \quad or \quad \begin{cases} m = n \\ \forall i \in [1, |V|], n_i = (n_i \triangleright k_{i;\omega_i(n)}) \end{cases}$$

where $\omega_v(n)$ represents the set of resources of variable v at state n and where $(a \triangleright b) = a + 1$ if $b > a$, $(a \triangleright b) = a - 1$ if $b < a$ and $(a \triangleright b) = a$ if $b = a$.

Transitions have some biological interpretations. For the current example, horizontal left-to-right transitions correspond to the entering of extra-cellular lactose into the cell whereas horizontal right-to-left transitions correspond to the the breakdown of lactose into glucose.

Unfortunatelly, this asynchronous state graph has been built from the knowledge of the different parameters. In fact, usualy no information about these parameters is available, it is necessary to consider all possible values.

The gene regulatory graphs of Figure 6-A and 6-B, give rise to $2^3 = 8$ parameters for intra-cellular lactose ($intra$) and $2^1 = 2$ parameters for galactosidase (g). The $intra$-parameters can get 3 possible values (from 0 to 2), the g-parameters can get 2 possible values (0 or 1). So, there are $3^{2^3} \times 2^{2^1} = 26244$ different regulatory networks associated to the regulatory graph of the figures 6-A or 6-B. More generally each gene of a regulatoty graph contributes for $(out+1)^{2^{in}}$ different parameter combinations, where in and out are its in- and out-degrees in the graph, and gene contributions are multiplicative... The total number of differents regulatory networks denoted by Figure 6 is thus $26244 + 26244 + 2^{2^3} \times 2^{2^1} = 53512$ because Figure 6-C assumes that the two outgoing arrows of intra-cellular lactose share the same threshold.

Let us nevertheless note that the number of different asynchronous state graphs can be less than the number of parameterizations (two different parameterizations can lead to the same state graph). For example, the parameterization deduced from the one of Figure 7 by replacing the value of $k_{intra,\{extra,intra\}}$ by 0, leads to the same dynamics.

Anyway, the number of parameterizations depends of the number of interactions pointing on each variables following a double exponential.

2.4 Introducing multiplexes

In order to decrease the number of parameterizations, other structural knowledge can be useful. For example, let us consider a variable c that has 2 different activators a and b. Without information, we have to consider $2^2 = 4$ different parameters associated with the four situations : what is the evolution of c when both regulators are absent, when only a is present, when only b is present and when both are present.
Sometimes additional structural knowledge can be derived from molecular biology: for example, it could be known that the regulation takes place only when both are present, because the effective regulator is in fact the complex of a and b. In such a case, the four parameters account for only two parameters: what is the evolution when the complex is present (both regulators are present) and when the complex is absent (at least one of the regulator is absent). Such information reduces the number of parameters, and drastically decreases the number of parameterizations to consider.

Multiplexes allows the introduction of such information [2]. They provide a slight extension of the R. Thomas' modelling, with explicit information about cooperative, concurrent or more complex molecular interactions [13, 14]. Intuitivelly, a regulatory graph with multiplexes is a graph with two kinds of vertices: variables account for vertices (they constitute the set V of nodes, V for variables) and moreover each information about cooperative concurrent or more complex molecular interactions also gives rise to a vertex (they constitute the set M of nodes, M for multiplexes). Information about molecular interactions is coded

into a logical formula that explains when the interaction takes place. In the previous example of complexation, the interaction takes place only when both regulators are present that is when $(a \geq s_a) \wedge (b \geq s_b)$.

Definition 4. *A* gene regulatory graph with multiplexes*, is a tuple* $G = (V, M, E_V, E_M)$ *such that:*

1. $(V \cup M, E_V \cup E_M)$ *constitutes a (labelled) directed graph whose set of nodes is* $V \cup M$ *and set of edges is* $E_V \cup E_M$*, with* $E_V \subset V \times \mathbb{N} \times M$ *and* $E_M \subset M \times (V \cup M)$.
2. V *and* M *are disjoint finite sets. Nodes of* V *are called* variables *and nodes of* M *are called* multiplexes*. An edge* (v, s, m) *of* E_V *is denoted* $(v \xrightarrow{s} m)$ *where* s *is called the threshold.*
3. *Each variable* v *of* V *is labelled with a positive integer* b_v *called the bound of* v.
4. *Each multiplex* m *of* M *is labelled with a formula belonging to the language* L_m *inductively defined by:*
   - *If* $(v \xrightarrow{s} m) \in E_V$*, then* v_s *is an atom of* L_m*, and if* $(m' \to m) \in E_M$ *then* m' *is an atom of* L_m.
   - *If* ϕ *and* ψ *belong to* L_m *then* $\neg\phi$*,* $(\phi \wedge \psi)$*,* $(\phi \vee \psi)$ *and* $(\phi \Rightarrow \Psi)$ *also belong to* L_m.
5. *All cycles of the underlying graph* $(V \cup M, E_V \cup E_M)$ *contain at least one node belonging to* V [2].

Let us remark that the point 1 of the previous definition separates two sets of edges. On one hand the first set is made of edges starting from a variable: their targets are multiplexes, they are labelled by a threshold that determine the atoms used in the target multiplexes. On the other hand, the second set is made of edges starting from a multiplex: their targets can be either a variable (the target of the complex interaction) or a multiplex (the expressionlogical formula of the source multiplex plays the role of an atom in the language of the target multiplex).

We define now the regulatory network with multiplexes as a regulatory graph with multiplexes provided with a familly of parameters which define the evolutions of the system according to the subset of predecessors (which are now multiplexes instead of variables).

Definition 5. *A* gene regulatory network with multiplexes *is a couple* $(G, \mathcal{K})$ *where*

- $G = (V, M, E_V, E_M)$ *is a regulatory graph with multiplexes.*
- $\mathcal{K} = \{k_{v,\omega}\}$ *is a family of parameters indexed by* $v \in V$ *and* $\omega \subset G^{-1}(v)$ *such that all* $k_{v,\omega}$ *are integers and* $0 \leq k_{v,\omega} \leq b_v$.

As in the classical framework, the parameters $k_{v,\omega}$ define how evolves the variable v when set of effective interactions on v is $\omega \subset G^{-1}(v)$. This set of effective interactions, named set of *resources* is defined inductivelly for each variable v and each state η :

[2] this condition is mandatory for the definition of dynamics (Definition 6).

Definition 6. *Given a regulatory graph with multiplex $G = (V, M, E_V, E_M)$ and a state η of G, the set of resources of a variable $v \in V$ for the state η is the set of multiplexes m of $G^{-1}(v)$ such that the formula φ_m of the multiplex m is satisfied. The interpretation of φ_m in m is inductively defined by:*

- *If φ_m is reduced to an atom v_s of $G^{-1}(m)$ then φ_m is satisfied iff $v \geq s$ according to the state η.*
- *If φ_m is reduced to an atom $m' \in M$ of $G^{-1}(m)$ then φ_m is satisfied iff $\varphi_{m'}$ of m' is satisfied.*
- *If $\varphi_m \equiv \psi_1 \wedge \psi_2$ then φ_m is satisfied if ψ_1 and ψ_2 are satisfied; and we proceed similarly for all other connectives.*

We note $\rho(v, \eta)$ the set of resources of v for the state η.

Definition 3 of the asynchronous state graph remains valid for gene regulatory graphs with multiplexes: the set of vertices does not change, nor the definition of transitions, the only difference resids in the definition of the state of resources: $\omega_i(n)$ has to be replaced by $\rho(v, n)$.

The contribution of multiplexes is thus simply to decrease the number of parameters. Introducing a multiplex corresponds to specify how the predecessors of the multiplex cooperate, and allows one to associate a single parameter whatever the number of predecessors. For example, if the cooperation of three regulators on a common target is well know, without multiplexes, one needs $2^3 = 8$ parameters to describe the evolutions of the targer in each situation whereas when considering multiplexes, only 2 are mandatory: one to describe the evolution of the target when cooperation of regulators takes places, and another to describe the evolution of the target when the cooperation does not take place.

3 Temporal logic and Model Checking for biology

Since the parameters are generally not measurable *in vivo*, finding a suitable classes of parameters constitutes a major issue of the modelling activity. This *reverse engineering* problem is a *parameter identification problem* since the structure of the interactions is supposed known, see for example [15] for such a problem in the differential framework.

In our discrete framework, this problem is simpler because of the finite number of parameterizations to consider. Nevertheless this number is so enormous that a computer aided method is needed to help biologists to go further in the comprehension of the biological system under study. Moreover, when studying a system, the biological knowledge arrives in an incremental manner. It would be apreciable to apprehend the problem in such a way that when a new knowledge has to be taken into account, previous work is not put into question. In other words, one would like to handle not only a possible model of the system, but the exhaustive set of models which are, at a given time, acceptable according to the current knowledge.

Biological knowledge, extracted from the litterature, about the behaviour of the system, can be seen as constraints on the set of possible dynamics: a model

is satisfactory only if its dynamics are compatible with the biological behaviours reported in the litterature. In a similar way, when a new *wet* experiment leads to new knowledge about the behaviour of the system, it can also be used as a new constraint which filter the set of possible models: only the subset of models whose dynamics is compatible with this new experiment have to be conserved for further investigation.

It becomes straightforward that a computer aided approach has to be developped in order to manipulate the different knowledge about the dynamics of the system and to make use of them to automatically handle the set of compatible parameterizations. Logic precisely allows the definition of the sets of these models. In constitutes a suited approach for addressing such a combinatorial problem.

3.1 The Computation Tree Logic (CTL)

Computation tree logic (CTL) is a branching-time logic, in which the time structure is like a tree: the future is not determined; there are different paths in the future, in other words, at some points of the time, it is possible to choose among a set of different evolutions.

CTL is generally used in formal verification of software or hardware, specially when the artificial system is supposed to control systems where consequences of a bug can lead to tragedies (public transportation, nuclear power plants, ...). For such a goal, software applications known as model checkers are useful: they determine if a given model of a system satisfies or not a given temporal property which is written in CTL.

Syntax of CTL. The language of well-formed CTL Formulae is generated by the following recursive definition :

$$\phi ::= \begin{cases} \bot \mid \top \mid p \mid & \text{atoms} \\ (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \Rightarrow \phi) \mid (\phi \Leftrightarrow \phi) \mid & \text{usual connectives} \\ AX\phi \mid EX\phi \mid AF\phi \mid EF\phi \mid & \text{temporal connectives} \\ AG\phi \mid EG\phi \mid A[\phi U \phi] \mid E[\phi U \phi] & \text{temporal connectives} \end{cases}$$

where $\bot$ and $\top$ codes for `False` and `True`, p denotes a particular atomic formula, and ϕ is another well formed CTL formula. In the context of R. Thomas theory for genetic regulatory networks, the atoms can be of the form $(a \propto n)$ where

- a is a variable of the system,
- $\propto$ is a operator in $\{<, \leq, >, \geq\}$
- n is an integer belonging to the interval $[0, b_a]$.

Semantics of CTL. This definition uses usual connectives ($\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ as well as *temporal modalities* which are pairs of symbols: the first element of the pair is A or E and the second belongs to $\{X, F, G, U\}$ whose meanings are given in the next table.

First letter		Second letter	
A	for All paths choices	X	neXt state
E	for at least one path choice (Exist)	F	some Future state
		G	all future states (Globally)
		U	Until

Figure 8 illustrates each *temporal modality* :

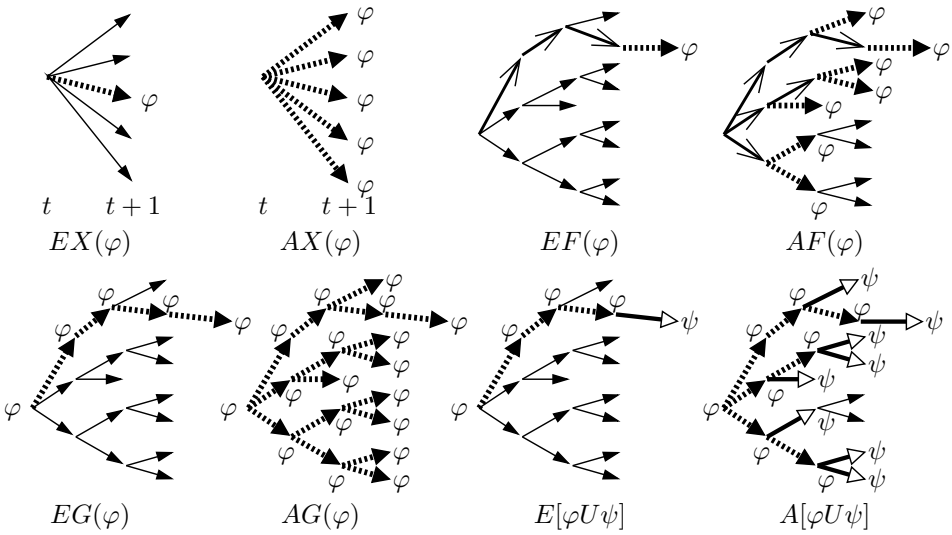

Fig. 8. Semantic of CTL formula. Dashed arrows (resp. arrows with empty head) point on states where φ (resp. ψ) is satisfied.

- $EX(\varphi)$ is true at the current state if there exists a successor state where φ is true.
- $AX(\varphi)$ is true at the current state if in all successor states, φ is true.
- $EF(\varphi)$ is true at the current state if there exists a path leading to a state where φ is true.
- $AF(\varphi)$ is true at the current state if all paths lead to a state where φ is true.
- $EG(\varphi)$ is true at the current state if there exists a path starting from the current state whose all states satisfy the formula φ.
- $AG(\varphi)$ is true at the current state if all states of all paths starting from the current state, satisfy the formula φ.
- $E[\varphi U \psi]$ is true at the current state if there exists a path starting from the current state leading to a state where ψ is true, and passing only through states satisfying φ.
- $A[\varphi U \psi]$ is true at the current state if all paths starting from the current state lead to a state where ψ is true, pass only through states satisfying φ.

For example $AX(intra \geq 1)$ means that in all next states accessible from the current state in the asynchronous state graph, the concentration level of *intra* is greater or equal to 1. Note that this last formula is false in the asynchronous state graph of figure 7 if the initial state is $(1,1)$ or $(0,1)$ and is is true for all other initial states. Formula $EG(g = 0)$ means that there exists at least one path starting from the current state where the concentration of g is constantly equal to 0. In Fig. 7 no state satisfies this formula, because from each state, all paths will return to a state where the concentration of g is equal to 0.

We say that a model or a state graph satisfies a CTL formula, if each state satisfies the formula.

3.2 CTL to encode biological properties

CTL formulas are useful to express temporal properties of the biological system. Once such properties have been elaborated, a model of the biological system will be acceptable only if its state graph satisfies the CTL formulas, otherwise, it is not considered anymore.

The first temporal property focuses on the functionality of the entering of lactose into the cell. When external lactose is constantly present in the environment, after a sufficently long time, intra-cellular lactose will increase and will at least cross its first threshold. This first property can then be written in CTL by the formula φ_1 as follows :

$$\varphi_1 \equiv AG(extra = 1) \Longrightarrow AF(intra > 0)$$

This CTL formula is not satisfied by the state graph of Figure 9 (where only the plane $extra = 1$ is drawn) because from the state $(0,0)$, it is not possible to increase the abstract level of *intra*. Let us remark that the state graph of Figure 7 does satisfy it because each path leads to a state where *intra* is present.

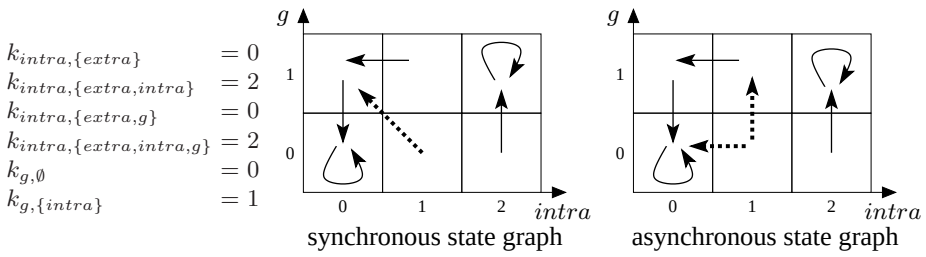

Fig. 9. From others values of parameters to the asynchronous state graph.

.

The second temporal property focuses on the production of galactosidase. When external lactose is constantly present in the environment, after a sufficently

long time, β-galactosidase will be sufficiently produced to degrade lactose, and then it will stay present forever. The translation of this property into CTL is:

$$\varphi_2 \equiv AG(extra = 1) \Longrightarrow AF(AG(g = 1))$$

This CTL formula is not satisfied by the states graphs of figures 7 and 9: In the first case, each path leads to a state where $g = 0$ whereas in the second case, from the state $(0, 0)$, it is not possible to increase the abstract level of g.

Similarly, when external lactose is constantly absent in the environment, after a sufficently long time, β-galactosidase will disappear. The CTL formula expressing this property is:

$$\varphi_3 \equiv AG(extra = 0) \Longrightarrow AF(g = 0)$$

The fourth formula we consider, states that when environment is rich in lactose, the degradation of intra-cellular lactose to produce carbon, is not sufficient in order to entirely consume the intra-cellular lactose. In other words, the permease allows the lactose to enter sufficiently rapidly in order to balance the consumption of intra-cellular lactose. To express this property, we focus on the case where extra-cellular lactose is present, intra-cellular lactose and galactosidase are present. In such a configuration, the intra-cellular lactose will never reach the basal level equal to 0. The CTL formula coding for this property is then written as follow :

$$\varphi_4 \equiv (AG(extra = 1) \wedge (intra > 0) \wedge (g = 1)) \Longrightarrow AG\,(intra > 0)$$

This CTL formula is not satisfied by the state graph of figure 7 because from each state, there is a path that leads to the total degradation of *intra*. The state graph of figure 9 does not satisfy it because each path starting from a state where $intra = 1$ leads to a total degradation of *intra*.

The two last temporal properties focus on the functionality of the lactose pathway, even when environment is no more rich in lactose. On one hand, when intra-cellular lactose is present at level 1 but extra-cellular lactose is absent, the pathway leads to a state where intra-cellular lactose has been entirely consummed, without passing through a state where *intra* is at its highest level (the only source of intra-cellular lactose is the extra-cellular one). Moreover, when this state is reached, there is no way to increase the concentration of intra-cellular lactose: its level then remains to 0.

$$\varphi_5 \equiv AG(extra = 0) \wedge (intra = 1) \Longrightarrow A\,[(intra = 1) U AG(intra = 0)]$$

On the other hand, when intra-cellular lactose is present at level 2 but extra-cellular lactose absent, the pathway leads to a state where intra-cellular lactose has decreased to level 1.

$$\varphi_6 \equiv AG(extra = 0) \wedge (intra = 2) \Longrightarrow AF\,(intra = 1)$$

These two previous CTL formulas cannot be checked in Figures 7 and 9 because the formulas concern the dynamics of the system when *extra* is absent whereas the figures focus on the dynamics when *extra* is present.

The formal language CTL has been developped in a computer science framework, and then is not dedicated to gene regulatory networks. For example, it is not possible to express in CTL that the dynamic system presents n different stable states. Nevertheless if we know a frontier between two stable behaviours, it becomes possible to express it in CTL. Let us consider a system where if variable a is at a level less than 2, the system converges to a stable state, and if variable a is at a level greater than 2, the system converges to another stable state. This property can be translated into the formula :

$$((a < 2) \Rightarrow AG(a < 2)) \quad \wedge \quad ((a >= 2) \Rightarrow AG(a >= 2))$$

Even if in some cases the translation of a property is tricky, in practice, CTL is sufficient to express the majority of biological properties useful for gene regulatory networks.

Let us now emphasize that the CTL language makes the link between the biological experiments and the models that are supposed to represent the behaviours of the studied biological system. Indeed,

- a CTL formula can be confronted against a model: the traces of the dynamics of the model verify the temporeal properties expressed through the CTL formula,
- a CTL formula can also be confronted against traces observed through an wet experiment: either the wet experiment is a realisation of the temporal property coded by the CTL formula, or it accounts for a counter example.

So, the modelling activity has to focus on the manner to select models that satisfy CTL formulas representing dynamic knowledge, extracted from experiments, about the system.

4 Computer aided elaboration of formal models

4.1 The landscape

The subject of this article is to present our computer aided *method* to accompany the *process of discovery* in biology, by using *formal modelling* in order to make valuable *predictions*.

According to this point of view, the "ultimate model", which would perfectly mimic the *in vivo* behaviour, is *not* our object of interest. It may be surprising for computer scientists, but this model is in fact rarely a subject of interest for biologists. Indeed the "ultimate model" would be untractable. The majority of valuable results, for a researcher in biology, comes from well chosen wet experiments and contributions to biology "are" wet experiments. The theoretical models are only intermediate objects, which reflect intermediate hypotheses that facilitate a good choice of experiments.

Consequently, a computer aided process of discovery requires to formally manage these models. It implies a formal expression of the sensible knowledge about the biological function under interest. It also implies a formal expression of

the set of (possibly successive) biological hypotheses that motivate the biological research. So, our method manages automatically the set of all possible models and we take benefit of this in order to guide a sensible choice of wet experiments.

There are two kind of knowledge:

- *Structural* knowledge, that inventories the set of relevant genes as well as the possible gene interactions. This knowledge can come from static analysis of gene sequences or protein sequences; it can come from dynamic data, e.g. transcriptomic data, *via* machine learning techniques; it can also come from the literature. This kind of knowledge can be formalized by one or several putative *regulatory graphs*.
- *Behavioural* knowledge, that reflects the dynamic properties of the biological system, such as the response to a given stress, some possible stationnary states, known oscillations, etc. This kind of knowledge can be formalized by a set of *temporal formulas*.

They give rise to several formal objects of different nature:

- The set $\mathcal{M}$ of all the structurally possible regulatory networks (for each putative regulatory graph, we consider all possible values of all the parameters). So, $\mathcal{M}$ can be seen as the set of all possible *models* according to the terminology of formal logic, since each regulatory network defines a unique state graph.
  For example, once the decision to make permease implicit is taken, the possible regulatory graphs are drawn in Figure 6, where all possible threshold distributions are considered. Remember that it gives rise to 53512 different parameterizations (section 2.3).
- The set Φ of the CTL formulas that formalize the dynamic properties.
  For example, according to Section 3.2, the set Φ can contain the 6 formulas from φ_1 to φ_6.
- Moreover, as already mentioned, the biological research is usually motivated by a *biological hypothesis*, that can be formalized *via* a set of CTL formulas H.
  For example, let us consider the following hypothesis: *"If extra-cellular lactose is constantly present then the positive circuit on intra-cellular lactose is functional."* It would mean that when extra-cellular lactose is constantly present, there is a multi-stationnarity on *intra*, which is separated by its auto-induction threshold 2 (according to the notion of characteristic state [26]). It can be formalized with $H = \{\psi_1, \psi_2\}$ as follows:
  $$\psi_1 \equiv \quad AG(extra = 1) \implies (intra = 2 \implies AG(intra = 2))$$
  $$\psi_2 \equiv \quad AG(extra = 1) \implies (intra < 2 \implies AG(intra < 2))$$

Of course, if "the ultimate model" were known and properly defined (*i.e.* a regulatory network with known values for all its parameters), it would satisfy exactly the set of behavioural properties that are true *in vivo*, and thus $\mathcal{M}$ would be reduced to a singleton (the ultimate model), Φ would be useless and H would be decided, by simply checking if H is satisfied by $\mathcal{M}$. The difficulty comes from the

uncertainty of the model structure and parameters, the incompleteness of the behavioural knowledge and the complexity of the systems which makes intuitive reasoning almost useless when studying hypotheses.

Fortunately, once the formalization step is performed, formal logic and formal models allow us to test hypotheses, to check consistency, to elaborate more precise models incrementally, and to suggest new biological experiments.
The set of potential models $\mathcal{M}$ and the set of properties $\Phi \cup H$ being given, two obvious scientific questions naturally arise:

1. Is it possible that $\Phi \cup H$ and $\mathcal{M}$? In other words: does it exist at least one model of $\mathcal{M}$ that satisfies all the formulas in $\Phi \cup H$? In the remainder, this question will be referred to as *the consistency question* of knowledge and hypotheses.
2. And if so, is it true *in vivo* that $\Phi \cup H$ and $\mathcal{M}$? As a matter of fact, the existence of a mathematical model satisfying the hypotheses is not sufficient. We must verify that the model reflecting the real *in vivo* behaviour belongs to the set of models that satisfy $\Phi \cup H$. It implies to propose experiments in order to validate or refute H (assuming that the knowledge Φ is validated).

For both questions, we can take benefit of computer aided proofs and computer optimized validation schemas can be proposed. More precisely, a CTL property can be confronted to traces and traces can be either generated by wet experiments or extracted from a state graph. Consequently a logic such as CTL establishes a bridge between *in vivo* experiments and mathematical models.

4.2 Consistency

In practice, when actually working with researchers in biology, there is an obvious method to check consistency:

1. Draw all the sensible regulatory graphs according to biological knowledge, with all the sensible, possible threshold allocations. It formalizes the structural knowledge.
2. From the discussions with the biologists, express in CTL the known behavioural properties as well as the considered biological hypotheses. It defines Φ and H.
3. Then, automatically generate, for each possible regulatory graph, all the possible values for all Thomas'parameters: we get all the possible regulatory networks. For all of them, generate the corresponding state graph: it defines $\mathcal{M}$. Our software platform SMBioNet handles this automatically.
4. Check each of these models against $\Phi \cup H$. SMBioNet intensively uses the model checker called NuSMV [5] to perform this step automatically.

If no model survive to the fourth step, then reconsider the hypotheses and perhaps extend model schemas...
On the contrary, if at least one model survives, then the biological hypotheses are consistent. Even better: the possible parameter sets $K_{v,\omega}$ have been exhaustively identified.

If we consider for example the set of models $\mathcal{M}$ characterized by Figure 6-A, there are 19 parameter settings leading to a dynamics compatible with the set of properties $\Phi \cup H$ proposed above. Among the 8+2=10 parameters that govern the dynamics, 6 of them are completely identified (*i.e.*, shared by the 19 parameter settings): $K_{intra} = 0$, $K_{intra,g} = 0$, $K_{intra,extra} = 1$, $K_{intra,extra\ g} = 1$, $K_g = 0$ and $K_{g,intra} = 1$. The 4 other parameters are those were *intra* is a resource of itself.

With respect to the classical ODE simulation method where some possible parameters are identified by trial and error, this method has the obvious advantage to compute the *exhaustive* set of possible models according to the current biological knowledge. It has also the crucial advantage of facilitating the refutation of models in a systematic manner.

The four steps described before, as such, replace a "brute force simulation method" by a "brute force refutation method" based on formal logic. In fact, the method is considerably more sophisticated in order to avoid the brute force enumeration of all the possible models. For example, in SMBioNet, when several regulatory networks share the same state graph, only one of them is considered: even better, it is not necessary to generate common state graphs as they can be identified *a priori*. In [9, 17], logic programming and constraint solving are integrated in this method and they almost entirely avoid the enumeration of models in order to establish consistency. In [23] model checking is replaced by proof techniques based on products of automata. Moreover, in practice, the use of multiplexes considerably reduces the number of different networks to be considered. Anyway, all these clever approaches considerably improve the algorithmic treatment, but the global method remains the same.

4.3 Selection of biological experiments

Once the first question (consistency) is positively answered, the second question (validation) has to be addressed: we aim at proposing "wet" experiment plans in order to validate or refute H (assuming that the knowledge Φ is validated). Here, we will address this question from a point of view entirely based on formal logic.

The global shape. Our framework is based on CTL whose atoms allow the comparison of the discrete expression level of a gene with a given integer value. So, a regulatory graph being given, we can consider the set $CTLsyntax$ of all the formulas that can be written about the regulatory graph, according to the CTL language. Notice that this set is not restricted to valid formulas: for example if φ belongs to $CTLsyntax$ then its negation $\neg\varphi$ also belongs to $CTLsyntax$.

- The set of hypotheses H is a subset of $CTLsyntax$. In Figure 10, we delimit the set $CTLsyntax$ with a bold black line and H is drawn in a circle.
- Unfortunately, it usually does not exist a single feasible wet experiment that can decide if H is valid or not *in vivo* (except for trivial hypotheses, which do not deserve a research campaign). A given wet experiment reveals a small set

of CTL properties, which are usually elementary properties. So, feasible wet experiments define a subset of $CTLsyntax$: the set of all properties that can be decided, without any ambiguity, from a unique feasible wet experiment. Such properties are often called *observable* properties and we note Obs this subset of $CTLsyntax$. In Figure 10, Obs is represented by the vertically hatched set.

- H is usually disjoint from Obs, however we can consider the set $Th_\Phi(H)$ of all the consequences of H (assuming the knowledge Φ). In Figure 10, $Th_\Phi(H)$ is represented by the horizontally hatched set.
- Let $E = Th_\Phi(H) \cap Obs$ be the intersection of $Th_\Phi(H)$ and Obs. It denotes the set of all the consequences of the hypotheses that can be verified experimentally. If ψ is a formula belonging to E then there exists a wet experiment e after which the validity of ψ is decided without any ambiguity:
  - If the experiment e "fails" then ψ is false *in vivo* and the hypothesis H is refuted.
  - If on the contrary the experiment e "succeeds" then ψ is true *in vivo*... Of course it usually does not imply H.

  In Figure 10, the intersection E defines the set of relevant experiments with respect to the hypothesis H.

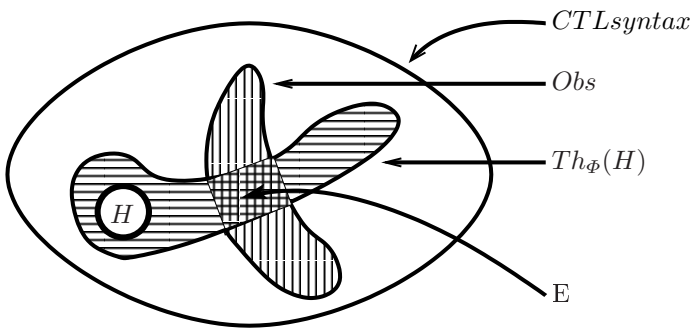

Fig. 10. Sets of CTL formulas involved in the computer aided selection of biological experiments

Observability and refutability. According to Popper [20], we should only consider hypotheses H for which we can propose experiments able to refute H: if H is false *in vivo*, there must exist a wet experiment e that fails. In other words: $\neg H \implies (\exists \psi \in E \mid \neg\psi)$, which is equivalent to: $E \Longrightarrow H$.

Consequently, refutability depends on the "power" of Obs with respect to H because $E = Th_\Phi(H) \cap Obs$. If Obs is "big enough" then it increases the capability of E to refute H. If E does not imply H (assuming Φ) it may result

that experimental capabilities in biology are unsufficient, so H is out of the scope of current know-how and current knowledge. It may also be possible that the wet laboratory has not enough fundings, or that the experimental cost would be disproportionate with regard to the problem under consideration.

We have shown so far that, for the modelling process, properties (Φ and H) are as much important as models ($\mathcal{M}$). Refutability issues prove that "experimental observability" (Obs) constitutes the third support of the process.

Often, observable formulas are of the form $(\rho \Longrightarrow AF(\omega))$ or $(\rho \Longrightarrow EF(\omega))$ where ρ characterizes some initial states that biologists can impose to a population of cells at the beginning of the experiment, and ω is deducible without any ambiguity from what can be observed at the end of the experiment. We use AF when all repeated experiments give the same result, and EF when we suspect that some additional conditions are imposed by the chosen experimental protocol during the experiments.

According to our small running example, we may consider that only external lactose can be controled, and that the flux of entering lactose can be roughly estimated. So, ρ can be of the form $extra = 0$ or $extra = 1$, possibly prefixed by a modality such as AG. Moreover, if the flux is considered "high", it denotes the presence of many permease proteins, and consequently implies that $intra$ has reached the threshold 2 (according to Figure 6-A). So, we may imagine for example that ω can be of the form $intra = 2$ or its negation $intra < 2$, possibly prefixed by modalities such as AG or AU. This defines Obs.
We will see later on that this observability is a rough underestimation, and how formal proofs can help improving it.

Selection of experimental schemas. Unfortunately, E is infinite in general, so, the art of choosing "good" wet experiments can be formalized by heuritics to select a finite (small) subset of formulas in E that has "good" chances to refute H if one of the corresponding experiments fails.

Classical testing frameworks from computer science [3,12] aim at selecting such subsets. However the subsets selected by the corresponding software testing tools are always huge because running lot of tests on a computer costs almost nothing. Nevertheless, the main idea of these frameworks can still be suitably applied to regulatory networks. Tests are selected incrementally and *completeness to the limit* is the main preoccupation: if H is not valid then the incremental selection process must be able to provide a counter-example after a certain number of iterations. It formally means that each possible reason for H to be false is tested after a finite (possibly large) amount of selection time.

Let us illustrate this completeness criteria on a simple case: according to our example, H is made of 2 formulas ψ_1 and ψ_2. Spending a lot of money to refute only ψ_1 would be a bad idea: this strategy would be incomplete because if H is false because ψ_2 is false, then this strategy will never refute H. Thus, one must try to refute both formulas.
Refutation of ψ_1: $AG(extra = 1) \implies (intra = 2 \Longrightarrow AG(intra = 2))$
It is well known that the truth table of "$\Longrightarrow$" is always true when the precondition

is false. Consequently, any wet experiment that does not ensure $AG(extra = 1)$ has no chance to refute the hypothesis. So, Popper tells us that any experiment associated with ψ_1 must constantly have external lactose (and remind that, from the context, glucose is always absent).
For the same reason, one must start with a population of bacteria with $intra = 2$ as initial state... and unfortunately the precondition $\rho \equiv AG(extra = 1) \wedge (intra = 2)$ is not reachable according to our description of observable formulas. Moreover, our knowledge (φ_1 to φ_6) never concludes on the atom $intra = 2$, so CTL cannot propose a sufficient condition to reach this initial state. Let us postpone this problem for a short time.
Refutation of ψ_2: $AG(extra = 1) \implies (intra < 2 \implies AG(intra < 2))$
The same reasonning applies: one must start with a population of bacteria with $intra < 2$ as initial state, and ensure $AG(extra = 1)$. Here, the formal definition of "$<$" allows to transform ψ_2 into the conjunction of two formulas, using classical *unfolding techniques*:

$$\begin{aligned} \psi_3 \equiv &\quad AG(extra = 1) \implies (intra = 0 \implies AG(intra < 2)) \\ \psi_4 \equiv &\quad AG(extra = 1) \implies (intra = 1 \implies AG(intra < 2)) \end{aligned}$$

From now on, we have to refute 3 hypotheses (ψ_1, ψ_3 and ψ_4) and again, the completeness of the method imposes to treat all of them. More precisely, the completeness of the unfolding technique, commonly used in PROLOG for example, ensures the completeness of our method to select wet experiments: in practice, after several unfoldings, the set of formulas under consideration gives a complete panel of the qualitatively different cases that deserve to be experimented in the wet laboratory. Unfolding steps make the cases more and more precise. This example gives the 3 obvious cases; for more elaborated examples, the exhaustive inventory of the relevant cases is far less obvious for a human, and the technique has proved useful.

On this example, we would have to ask the biologists if they can increase their experimental operability in order to control the value of $intra$ at the initial state of the experiments. Proof techniques can help. For example ψ_3 needs $intra = 0$ and there are formulas in Φ that conclude on $intra = 0$. In fact, it is not difficult to establish that Φ implies φ_7:

$$\varphi_7 \equiv \quad AG(extra = 0) \implies AF(AG(intra = 0))$$

The formula φ_7 indicates that the atom $intra = 0$ can be included in the preconditions ρ of observable formulas. This extension of *Obs* suggests the following experimental protocol: in order to ensure $intra = 0$, keep the cell population in an environment without external lactose for a sufficiently long time, and then, put external lactose in order to check ψ_3.

By the way, this ψ_3-experiment gives, after a sufficiently long time in the Petri dish, a high flux of entering lactose, which shows that $intra = 2$. This refutes the hypothesis *H in vivo*, although it was consistent.

5 Conclusion

We have shown that formal methods from computer science can help the discovery process in molecular biology. More precisely, we have proposed a formal modelling method for gene regulatory networks, based on the discrete approach of René Thomas, that we have enriched with CTL and model checking. We have also shown that, even if simulations are useful and easy to perform in this setting, the main subject to address is the logical identification of parameters. During the modelling process, we must face incomplete knowledge and, consequently, we must manage in parallel:

- a (possibly large) set of different potential models and parameter values,
- a set of known behavioural properties (whose consistency has to be managed),
- a set of biological hypotheses, that motivate the biological research (and whose consistency has to be managed),
- and a set of conceivable wet experiments (which appears sometimes to be unsufficient to identify the parameters or to refute some potential models).

Our framework is not only a method to identify the "good" model(s), it is rather a full modelling process to accompany the discovery process in molecular biology, from the elaboration of abstract models to the design of well chosen wet experiments.

At the beginning of the process, the hypotheses of interest being expressed, we can simplify the models under consideration provided that the simplifications do not modify the truth of the hypotheses. This simple idea allows to efficiently reduce the number of nodes in the regulatory graph.

Consistency check is the next step of our method and the formalization of biological properties into temporal logic is a key point. It facilitates the use of tools from formal logic, *e.g.* model checking or constraint solving, in order to establish the consistency of knowledge and hypotheses.

Lastly, a formal definition of the so called observable properties helps to suggest wet experiments in order to validate or refute the hypotheses *in vivo*. If the hypotheses are consistent, then formal manipulations of the syntax of the formulas, such as unfolding techniques and theorem proving, can produce observable consequences that describe wet experiment schemas.

All in all, formal models are not "universal" in biology: they are only a temporary intermediate that serve to validate or refute biological hypotheses.

References

1. J. Ahmad, J. Bourdon, J. Eveillard, D. Fromentin, O. Roux, and C. Sinoquet. Temporal constraints of a gene regulatory network: refining a qualitative simulation. *Biosystems*, 98(3):149–159, 2009.
2. G. Bernot, J.-P. Comet, and Z. Khalis. Gene regulatory networks with multiplexes. In *European Simulation and Modelling Conference Proceedings*, ISBN 978-90-77381-44-1, pages 423–432, France, October 27st-29th 2008.

3. G. Bernot, M.C. Gaudel, and B. Marre. Software testing based on formal specifications: A theory and a tool. *Software Engineering Journal*, 6(6):387–405, 1991.
4. L. Cardelli, E. Caron, P. Gardner, O. Kahramanogulları, and A. Phillips. A process model of rho gtp-binding proteins. *Theoretical Computer Science*, 410(33-34):3166–3185, 2009.
5. A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV Version 2: An OpenSource Tool for Symbolic Model Checking. In *Proc. International Conference on Computer-Aided Verification (CAV 2002)*, volume 2404 of *LNCS*, Copenhagen, Denmark, July 2002. Springer.
6. Julio Collado-Vides, Boris Magasanik, and Temple Smith. *Integrative approaches to molecular biology*. The MIT press, 1996.
7. M. Curti, P. Degano, C. Priami, and C.T. Baldari. Modelling biochemical pathways through enhanced π-calculus. *Theoretical Computer Science*, 325(1):111–140, 2004.
8. H. de Jong. Qualitative modeling and simulation of bacterial regulatory networks. In A. Uhrmacher and M. Heiner, editors, *Computational Methods in Systems Biology (CMSB-08)*, volume 5307 of *Lecture Notes in Bioinformatics*. Springer-Verlag, 2008.
9. E. Fanchon, F. Corblin, L. Trilling, B. Hermant, and D. Gulino. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In *CMSB*, pages 104–118, 2004.
10. J.-L. Gouzé. Positive and negative circuits in dynamical systems. *Journal of Biological Systems*, 6:11–15, 1998.
11. F. Jacob and Monod. J. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3:318–356, 1961.
12. C. Jard and T. Jéron. TGV: theory, principles and algorithms. a tool for the automatic synthesis of conformance test cases for non-deterministic reactive systems. *Software Tools for Technology Transfert*, 7(4):297–315, 2005.
13. Z. Khalis, G. Bernot, and J.-P. Comet. *Proc. of the Nice Spring school on Modelling and simulation of biological processes in the context of genomics*, chapter Gene Regulatory Networks: Introduction of multiplexes into R. Thomas' modelling, pages 139–151. EDP Science, ISBN : 978-2-7598-0437-5, 2009.
14. Z. Khalis, J.-P. Comet, A. Richard, and G. Bernot. The smbionet method for discovering models of gene regulatory networks. *Genes, Genomes and Genomics*, 2009.
15. P. Kügler, E. Gaubitzer, and S. Müller. Parameter identification for chemical reaction systems using sparsity enforcing regularization: A case study for the chlorite-iodide reaction. *Journal of Physical Chemistry A*, 113(12):2775–2785, 2009.
16. John W. Little. Threshold effects in gene regulation: When some is not enough. *PNAS*, 102(15):5310–5311, 2005.
17. D. Mateus, J.-P. Gallois, J.-P. Comet, and P. Le Gall. Symbolic modeling of genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, 5(2B):627–640, 2007.
18. A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya. A reduction of logical regulatory graphs preserving essential dynamical properties. In *Computational Methods in Systems Biology, CMSB'09*, volume 5688 of *LNBI (Lect. Notes in Bioinformatics)*, pages 266–280, 2009.
19. E. Plahte, T. Mestl, and S.W. Omholt. Feedback loops, stability and multistationarity in dynamical systems. *Journal Biological Systems*, 3:409–413, 1995.

20. Karl R. Popper. *Conjectures and refutations : the growth of scientific knowledge*. Routledge & Kegan Paul, 1965.
21. A. Richard and J.-P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics*, 155(18):2403–2413, 2007.
22. A. Rizk, G. Batt, F. Fages, and S. Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25(12):i169–178, 2009.
23. H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoretical Computer Science*, 391(3):258–275, 2008.
24. E. H. Snoussi. Necessary conditions for multistationarity and stable periodicity. *Journal of Biological Systems*, 6:3–9, 1998.
25. E.H Snoussi. Qualitative dynamics of a piecewise-linear differential equations : a discrete mapping approach. *Dynamics and stability of Systems*, 4:189–207, 1989.
26. E.H. Snoussi and R. Thomas. Logical identification of all steady states : the concept of feedback loop caracteristic states. *Bull. Math. Biol.*, 55(5):973–991, 1993.
27. C. Soulé. Graphical requirements for multistationarity. *ComPlexUs*, 1:123–133, 2003.
28. R. Thomas. On the relation between the logical structure of systems and their ability to generate multiple steady states and sustained oscillations. In *Series in Synergetics*, volume 9, pages 180–193. Springer, 1981.
29. R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.