Formal methods applied to gene network modelling

Gilles BERNOT, University Nice Sophia Antipolis, I3S laboratory, Sophia Antipolis, France
Jean-Paul COMET, University Nice Sophia Antipolis, I3S laboratory, Sophia Antipolis, France
El Houssine SNOUSSI, University Mohammed V - Souissi, Rabat, Maroc

Abstract: Logic and its counterpart in computer science, namely formal methods, offer both a solid, flexible ground for modelling in biology, and a methodological backbone for accompanying experimental strategy within a multidisciplinary research process. We thus consider a partial knowledge setting in which we cannot take the risk to study a single model that may become a bad model when the biological knowledge increases. We manage at each step of the process the set of all the possible models according to the current knowledge. This is precisely the reason why logic is a suitable tool: it describes sets of models by their properties and it is able to manipulate them. Among those maniputations, the formal validation activity is particularly appreciated by researchers in biology: it suggest new biological experiments in a computer aided manner in such a way that some kind of completeness can be reached.
The methodology proposed in this chapter is independant both of the biological object and of the underlying logic, but we illustrate its main phases in the context of discrete modelling of gene networks using a particular temporal logic.

Keywords: Systems biology, Complex systems modelling, Discrete models, Gene networks, Formal methods, Temporal logic, Hoare Logic.

1 Introduction

Biology is an experimental science where most of the studied objects are so called complex systems and, in addition, theoretical biology is lacking universal laws, contrarily to what we can find in physics for example. When studying a biological phenomenon, the amount of "universal properties" that govern its behaviour is very low when compared to the amount of *ad hoc* properties extracted from biological knowledge about the considered object(s). Biology is mostly a science where researchers have constantly to deal with "exceptions" (meaning *a priori* surprising experimental results). Finding the causality chain(s) that can explain such "exceptions" regularly opens great challenges for theoretical biology. As a consequence, the mathematical modelling of biological systems is a field of research that must combine a wide variety of properties into a model, and the modelling frameworks must be particularly versatile. So, theoretical biology has become a particularly rich and exciting multidisciplinary research area.

Also, biology is deeply an *experimental* science, where a discovery is most of the time precisely centered around experiments that lead to "exceptional" observations. The investment to find and perform those key experiments is currently by far much bigger that the investment to explain the phenomena, and validated experimental observations constitute the research results by themselves. Consequently, what researchers in biology call a *result* should almost always lead to novel experiments. The novel "wet" experiments inspired by a mathematical model of a biological phenomenon are of utmost importance. Indeed, this *experimental feedback* from a mathematical model often constitutes the main part of its value.

In this chapter, we try to convince the reader that *formal methods* and *logic* are the adequate tools to define modelling frameworks for biology, and that they offer a solid, flexible methodolog-

ical backbone for discovery (supporting verification and *experimental* validation of hypotheses) within a multidisciplinary research process.

- The first obvious idea is that formal methods are able to automatize a large part of logical reasonning that relieve the biologists of boring (and error prone) deductions. Although the basic reasonings are generally simple, the number of elements involved is nevertheless important, so that the combinatorial of the arguments deserves to be mechanized.
- The second idea comes from a now widely recognised vision of experimental sciences introduced by Karl Popper [Pop63]; in short *the ability to design experiments that may falsify a model is fundamental*. When constructing a mathematical model, one should not only be able to mimic the observed biological behaviour, one should also have sufficient experimental capabilities to design novel experiments potentially able to refute the model in all its details. Here also, logic and formal methods can help.

Both ideas may seem far from the usual simulation schema where the computer is mainly used for its computing power. Here, the computer is used to help reasoning and to accompany an experimental strategy. Nevertheless, simulations are obviously helpful as well; they allow the modeler to gain an empirical knowledge of the system, by trying several parameter values.

In this chapter, we have tried to isolate, in Section 3, the bulk of our methodology which is independant both of the biological object and of the underlying logic. Then in Section 4 we focus on discrete modelling of gene networks, intensively using ideas due to René Thomas in the 70's with a modern finishing touch using temporal logics. The method is formally described within this context in Section 5 (consistency of biological hypotheses) and in Section 6 (validation of biological hypotheses).
A few preliminaries about biology and how biologists make use of mathematical models come in the next section.

2 From gene interactions to gene network modelling

2.1 Gene regulations and regulatory genes

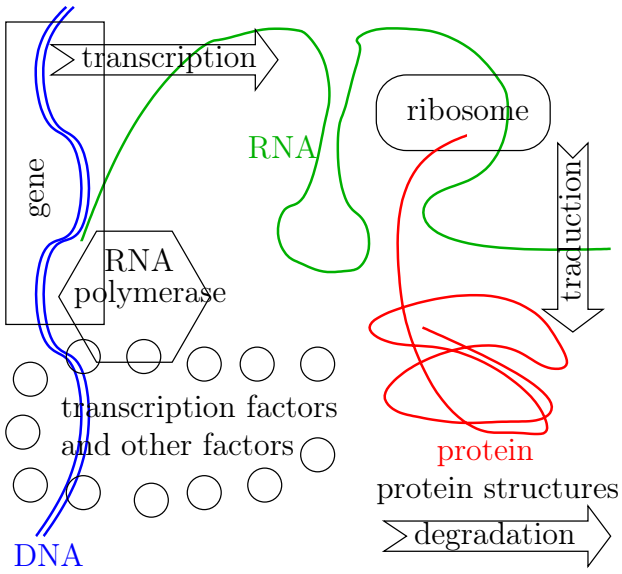
Very few knowledge from molecular cell biology is necessary to understand what gene networks are (but full details can be found for example in [LBK+08]). A gene can be considered as a portion of a DNA strand, belonging to the genetic material of a cell, and producing (when *expressed*) biomolecules that participate to the cell activity. See Figure 1.

To become expressed, a gene needs to be *transcribed*, that is to be "read" by a RNA polymerase that produces a RNA sequence reflecting the gene sequence. The RNA polymerase, in turn, is controled by *transcription factors* that can promote or inhibit its recruitment for this gene.

A transcription factor, possibly with the help of coactivators or corepressors, is able to recognise its target gene *via* some specific DNA sequence (situated near the gene on the DNA strand) called promoter or activator of the gene. Once fixed, a transcription factor can favour or, on the contrary, it can block the transcription of the gene. By the way, a transcription factor can also be blocked by certain other factors, so that an activator can be in fact an inhibitor of an inhibitor (and we may imagine any logical combination of factors).

Depending on the gene under consideration, the effective biomolecules that it produces can be

Figure 1: Gene expression: transcription and translation



Gene transcription is effective under complex conditions on the presence or the absence of some "factors", including transcription factors. The traduction, if any, can result on different protein structures. Regulatory genes produce "factors" that can modify the production rate of other genes, modify their protein structure or degradation rate, *etc.*

directly the RNA molecules (including microARN molecules) or proteins obtained after *translation* of the transcribed RNA. *Ribosomes* perform the translation and they are themselves composed of several RNA and proteins (a *complex* of biomolecules). Also, transcribed RNA can be modified in several manners before being translated (alternative splicing), and translated proteins can change their structure depending on the cellular context.

Lastly, proteins and RNA are subject to certain degradation rates in the cell. In particular, even if a gene constantly produces proteins, the degradation rate imposes an equilibrium so that the total number of proteins is bounded.

All the factors mentioned here (RNA polymerase, transcription factors, coactivators, corepressors, spliceosomes, ribosomes, and so on) are themselves RNA, proteins, or complexes made of RNA and proteins. Consequently they are themselves produced by other genes: each gene expression is controled by the expression of other genes. The genes that control the expression of other genes (possibly indirectly, e.g., by modifying the degradation rate of a protein) are called *regulatory genes*. Obviously, the presence or the absence of the products of the regulatory genes (or more generally their concentration level) control the behaviour of a cell.

Most of the time, researches in molecular cell biology study some "biological functions" of a cell. It consists in a particular behaviour of the cell in response to certain environments or to a "stress" and the question is to understand which genes participate to the response and how. By comparison between stress and non-stress conditions, the set of genes participating to the biological function is often rapidly known, so that one has to understand the dynamics of the

interactions between these genes.

Biologists may spend decades to establish precisely every single molecular interaction that participates to the expression or the inhibition of a gene. So, one can take benefit of this knowledge in order to inventory some potential gene interactions (e.g., the protein of gene a can repress the transcription of gene b) and some needed cooperations (e.g., proteins of genes a and b are both required to activate gene c because they have to form a complex before acting on c). Due to the cost and difficulty of such precise knowledge, gene interaction are also often simply observed at the cellular level (e.g., every time gene a is activated, so does gene b after a short delay) and systematic experiments are done in order to inventory such interactions.

So, when it comes to the study of the global behaviour of a given biological function, not only the set of genes that should participate to this function is known, but also the mutual interactions between genes are almost completely known. By "almost completely", one should understand that the ability of a gene to apply a regulation on another gene is often known but we do not know the strength of the regulation, nor the succession of gene variations. A few interactions can be missing and a few interactions, although molecularly established, can have a insignificant strength with respect to the biological function under consideration. The set of involved genes can also be, in some rare cases, only approximately known.

In terms of a mathematical model, it means that we almost entirely know the graph of possible interactions between genes. This graph is static (it contains no rule to evolve along time) and it does not furnish valuable indications about the strength of the interactions. *A fortiori* it does not allow to deduce the dynamic behaviour of the system. Lot of additional parameters will be needed.

2.2 Reverse engineering

In practice, there are at least four visions about the role of mathematical models for biology:

- *Models as a tools to store complete knowledge.* If the biological function is known in almost all its molecular details and if a large number of behavioural properties have been experienced and properly understood, then biologists may consider that a mathematical model reflecting both the static knowledge and the dynamics of the system is a good tool to store all their knowledge, better than a huge collection of scientific articles. It begins to be the case for well studied species such as yeast (e.g., [CCC+04]) or *Escherichia coli* (e.g., [EIP01]).
  There is a thin line between "fairly complete mathematical models" and "faithful simulations." Sometimes, the model is (too) rapidly asked to "replace" the biological object by itself. We observe an increasing number of such applications in chemistry, in pharmacology or for cosmetics design, due to legal requirements such as REACH in Europe for example, where both the innocuity of new products for human health must be established and animal testing is restricted [RH09]. Then, intensive *in silico* simulations are performed, reflecting many different situations, and the accumulation of results play the role of "evidence" to corroborate some assertion. The number of "wet" experiments is then reduced to its minimum and similarities with other already validated assertions are also intensively used.

- *Models as design plans.* Synthetic biology is a growing application domain for modelling. It mainly aims at making biological systems synthetise useful molecules, or more generally exhibit useful behaviours [Kep13]. Currently, synthetic biology is mostly implementing

the following idea: life is too complex to build "useful" living cells *ab initio*, so let us add into a common cell (such as *Escherichia coli*) a small set of genes whose products do not interact with the ones of the hosting cell. Then, this small set of new genes can be designed independantly in order to get the intended behaviour: an approach that helps a *modular design*, as it is the case for usual manufactured products [PW09]. Although such an independance between genes asks for a huge knowledge about the hosting cells and for the design of a fairly complete library of synthetic genes and proteins (including their interaction properties), this allows us to manage subnetworks of size reachable for the current know-how.

- *Models as tools to understand causality.* Research teams in molecular cell biology often work on a biological function which is only partly understood, with a lot of established facts but also a lots of gray areas. A collection of different hypothetic causality chains emerge, that are *a priori* able to explain observed behaviours... and the PI of the research team has often (her)his prefered assumption, which (s)he confront to experiments. As in most scientific questions about complex systems, many of these assumptions are inconsistent by themselves but these inconsistencies are not easy to uncover, arising from intricated feedbacks within the system. In fact, human reasonning is of little help to navigate between possible assumptions, and biologists have often the feeling to observe contradictory behaviours between different experiments. In such cases, mathematical models, and logic applied to these models, are of great help to accompany fundamental researches in biology [FH07], see Section 5.

- *Models as dashboard for the experimental strategy.* Researchers in biology say "a mathematical model must be predictive." Behind this somewhat vague vocabulary they do not mean that, interrogating the model, one must get a *prediction* of what the system will do, as for a meteorological model. What is behind is rather the following idea: a model that simply mimics already known behaviours of the biological object is *useless*. The motivation of a biologist to waste time making a mathematical model is to get a "predictive" feedback suggesting new ideas of experiments (remember that a research result in biology is usually a set of experimental observations). So, what is behind is indeed a Popper [Pop63] approach: trying to falsify the model in a rigorous manner is a way to drive the choice of experiments. A similar situation arises for software Verification and Validation (for the Validation activity to be more precise). Extracting test cases from a software specification involves heuristic formal techniques which are based on logical refutation of formulae [BGM91]. They appear to be applicable to biology, see Section 6.

As already mentioned in Section 1, our aim is to help researchers in molecular cell biology when they try to elucidate the mode of operation of a biological behaviour, when they have to face apparently contradictory experimental observations and when they have hypotheses to validate. We accompany the discovery process, helping researchers in biology to understand causality chains and to design an experimental strategy in a enlightened manner. Consequently we focus on the two last visions about modelling, using mathematical frameworks able to handle the dynamics of complex systems. In this context, the first vision of modelling is not under consideration yet because biological knowledge is far too fragmentary, and the second vision of modelling is dedicated to modifying the biological object under study, which is not the goal here.

From the modeler point of view, the problem is consequently a *reverse engineering* problem: the biological object is given, with limited knowledge about its functioning and limited experimental capabilities, so that many parameters of the system cannot be directly measured. The problem of *identification* of parameters is, as usual in complex systems, the main problem but

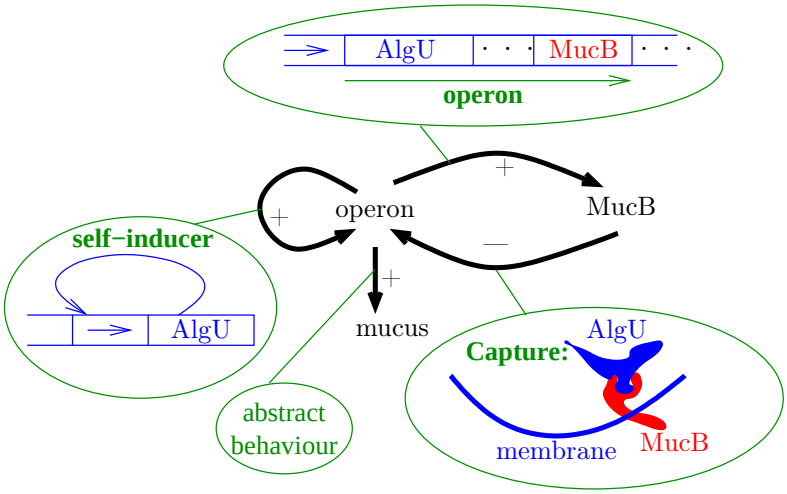
it is regarded under a special viewpoint: biologists have assumptions they want to validate and *neither the exact model nor the parameter values are the goal; only the validity of the assumptions matters*. We show in the next section that this slightly modified vision of the identification problem brings logic and formal methods in first place to accompany this reverse engineering activity for biology.

3 Logic: a tool for multidisciplinarity with experimental sciences

3.1 What we expect from models

Most of the time, when biologists ask the help of mathematicians or computer scientists, they have a collection of different credible interpretations of their experimental results, with several possible hypotheses, and these hypotheses are usually not mutually compatible. Modelling is then the only hope to sort out these different assumptions without a huge, costly, and in fact unthinkable, number of delicate experiments. It does not mean that the system itself (or at least its model) is huge: there are many complex systems with very few elements [Bar97] (often, feedbacks and non linear interactions create complexity). To give a simplistic example with two genes, let us consider the interaction graph of Figure 2.

Figure 2: Mucus production of *P. aeruginosa*



An operon activates several genes: AlgU, MucB and some other ones that participate to mucus production. The protein of AlgU is a transcription factor that activates the operon, so the operon activates itself, which we represents by an arc on the left part of the figure, where "+" means "activation." The protein of MucB is a membrane protein which is able to capture the protein of AlgU at the membrane, far from the operon, thus inhibiting the activation of the operon. So, the operon also inhibits itself *via* MucB (successive "+" and "−" arcs on the right part of the figure, where "−" means "inhibition").

Looking at the left (operon→operon) activation, if we totally ignore the right part of the

figure, one would predict that the behaviour could exhibit two attraction basins:

- a basin where the operon is expressed, so that it produces the protein of AlgU that pushes it to stay constantly expressed,
- and another basin where the operon is not expressed and it stays unexpressed for lack of AlgU.

Knowing that mucus is produced when the operon is expressed (down arrow with a "+" sign in the figure), the first behavioural basin of attraction should make *P. aeruginosa* produce mucus whilst the second one would show no mucus. What we call an *epigenetic switch*.

Now looking at the right part of the figure, the negative feedback (operon→MucB→operon) would induce apparently contradictory predictions:

- if the operon is expressed, then it produces the protein of MucB, so that it should be repressed after a certain delay,
- if the operon is not expressed, then the degradation of MucB will lower its repression on the operon, so that the operon could increase its expression level again.
- It could also depend on the relative degradation speeds of MucB and AlgU. If the degradation of AlgU is higher than the one of MucB, then the operon may stay unexpressed after a certain time.

So, according to the right part of the figure, and totally ignoring the left self activation, the production of the operon should exhibit oscillations, possibly damped oscillations, with only one attraction basin. What we call a *homeostasis*. Consequently, the left and right parts of the interaction graph suggest somehow contradictory predictions. Full details on this example can be found in [BCRG04] and [FMB+06].

This small example shows the typical questions, with possibly contradictory answers, that motivate biologists to use mathematical models. Most of the time, the dynamics of the system is the main question to solve and, when biologists come to such kind of precise questions, the static structure (i.e., the drawing of Figure 2) is fairly known. In practice, the design of the interaction graph mainly comes from the litterature (the difficult step of extracting the possible interactions from experimental data has been performed before). It may be necessary to consider a few possible variants of the interaction graph but the main parts are known. Some interactions can be hypothetic. For example, if they come from considerations on the sequence of genes or proteins then the actual interaction may be missing, due to a different folding of the proteins or due to different compartments in the cell, and so on.

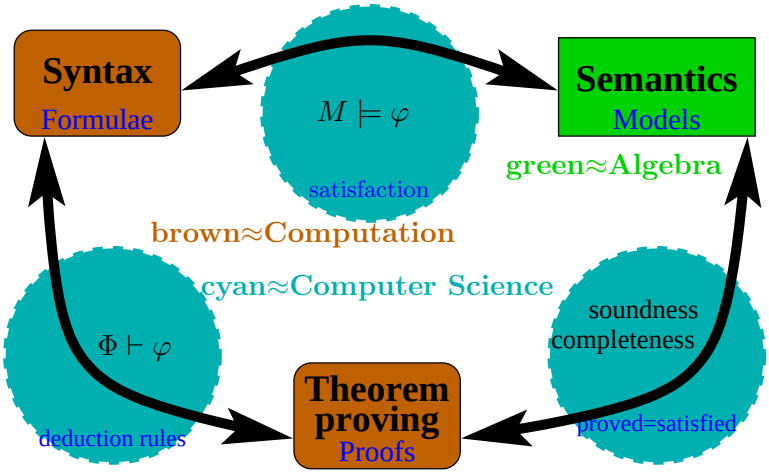
The main problem remains the identification of parameters, that can distinguish between the possible dynamics of the system. Notice however that only some consequences of the dynamics motivate the biologists, not all of them. For example the biological question may be only the presence or absence of mucus, whatever the behaviour of AlgU and MucB:

- As a consequence, *if* one can prove that, after a *partial* identification procedure, all remaining models exhibit our two previous attraction basins, *then* it becomes useless to continue the identification process until we get a single set of parameter values.

- Nevertheless, conversely, it is useless as well to exhibit a *single* set of parameter values compatible with the current biological knowledge and that exhibits two attraction basins... *if* one does not prove that *all* possible models do so.

Our aim (to accompany researchers in biology in order to accelerate the discovery process) implies that we are in a frame where the biological knowledge is growing along the process. Consequently, contrarily to a modelling process used to store complete knowledge with well established universal laws (as it is often the case in physics for example), we navigate in a partial knowledge setting and we cannot take the risk to study a *single* model that may become a bad model when the biological knowledge increases. We must *manage* at each step of the process *the set of all* the possible models according to the current knowledge. This is precisely the reason why *logic* is a suitable tool.

Figure 3: General logics at a glance



Syntax, semantics and theorem proving are the three first components of a general logic. The syntax defines the set of well formed formulas (usually *via* an inductive definition). The semantics mathematically defines the set of mathematical models within which one checks whether a formula is satisfied (usually models are sets that additionnaly carry internal operators, as in universal algebra). The theorem proving defines precisely what is a formal proof (mostly *via* proof trees).
Links between them are important: "$\models$" makes the link between syntax and semantics; "$M \models \varphi$" defines mathematically whether a model M of the semantics satisfies a formula φ of the syntax. "$\vdash$" makes the link between syntax and theorem proving; "$\Phi \vdash \varphi$" defines under which conditions there exists a deduction that proves φ under the set of hypotheses Φ (mostly *via* deduction rules that can be verified by a computer). Lastly soundness and completeness make the link between theorem proving and semantics: a provable theorem must be true semantically (soundness) and one greatly appreciates when all semantically satisfied properties are provable (completeness).

Figure 3 gives a good global picture of what a logic is. For sake of simplicity it leaves implicit the notion of *signature*. When using a logic to describe some properties about models, we use symbols. Some of the symbols come from the definition of the logic itself (such as the

conjunction "$\wedge$" or a universal quantifier "$\forall$" for example) and some additional symbols are required to describe the phenomena under consideration (such as the names of the considered biomolecules, MucB for example). The set of symbols that are specific to the problem under consideration, together with the conventions that limit their usage in formulae (e.g., the arity of each symbol) is called a signature.

Following [Mes89] and a huge corpus of subsequent works with several variants, a *general logic* is defined by five entities :

- A category of signatures Sig that defines the set of all conceivable signatures for the logic under consideration, as well as what is considered an inclusion of signatures or a renaming between signatures[1].
- A function[2] $For : Sig \to Set$ that, for each signature $\Sigma \in Sig$, defines the set $For(\Sigma)$ of all well formed formulae using this signature. So, For defines the *syntax* of the logic in Figure 3.
- A function[3] $Mod : Sig \to Cat$ that, for each signature Σ, defines the set $Mod(\Sigma)$ of all models[4] that can make sense with respect to this signature. That is, Mod defines the *semantics* of the logic.
- Binary relations $\vdash_\Sigma \subset \mathcal{P}(For(\Sigma)) \times For(\Sigma)$ such that $\Phi \vdash_\Sigma \varphi$ belongs to the relation if and only if there exists a proof of the formula φ under the set of hypotheses Φ. "$\Phi \vdash_\Sigma \varphi$" is read "$\Phi$ entails φ". So, the family $\vdash$ of binary relations $\{\vdash_\Sigma\}_{\Sigma \in Sig}$ defines the theorem proving part of Figure 3.
  Most of the time, a proof is defined as a tree whose nodes are elementary *deduction rules*, so that the binary relation $\vdash$, *via* its inductive definition, also covers the link between syntax and theorem proving in Figure 3.
- Binary relations $\models_\Sigma \subset Mod(\Sigma) \times For(\Sigma)$ such that $M \models_\Sigma \varphi$ belongs to the relation if and only if the model M satisfies the formula φ. "$M \models_\Sigma \varphi$" is read "M satisfies φ". So, the family $\models$ of binary relations $\{\models_\Sigma\}_{\Sigma \in Sig}$ defines the link between syntax and semantics in Figure 3.

General logics ask for a lot of "reasonable" properties about the binary relations $\vdash$ and $\models$, which may depend on the authors and the variants of this framework. Among them, let us mention reflexivity, transitivity or monotonicity for $\vdash$, as well as the so called satisfaction condition for $\models$. Our figure also highlights the soundness (also called correctness) and the compleness of a logic. They make a link between $\vdash$ and $\models$:

- *Soundness* is an unavoidable property that ensures a meaning to all this stuff. It requires that if one can prove $\Phi \vdash \varphi$ then it is semantically satisfied. In other words, if $\Phi \vdash \varphi$ then for all models M that satisfy all the formulae in Φ, we have $M \models \varphi$.
- *Completeness* is a (sometimes too) strong property. It requires that if φ is a consequence of Φ in all models then it is provable. In other words, if we have $M \models \varphi$ for all models M

[1] The biological models being essentially non-modular [BT09], the signature morphisms for biological systems are usually reduced to renaming morphisms.

[2] In fact a *functor* in order to transport inclusion and renaming from signatures to formulae. *Set* is the category of sets.

[3] In fact a *contravariant functor* in order to transport inclusion and renaming from signatures to models.

[4] In fact the *category of models*, so that models can be treated as in universal algebra. *Cat* is the category of categories.

that satisfy all the formulae in Φ, then $\Phi \vdash \varphi$.
Completeness is often unreachable in computer science because many domains of computer science involve recursivity or structural induction, consequently covering axioms as powerful as the Peano axioms [Bou06] and the Gödel's incompleteness theorem [God31] implies that completeness is impossible. This is one of the reasons why *domain specific frameworks* constitute an important research area in formal methods: by reducing and by specializing the expressiveness of the logic one tries to recover completeness.

Back to the example given in Figure 2, the signature Σ to handle the biological network could be the set of nodes of the graph (operon, MucB and mucus). A model of $Mod(\Sigma)$ could be for example a set of trajectories in $[0,1]^3$ where a triplet (x,y,z) represents the "state" of the network at a given moment, x, y and z being respectively the levels of expression for operon, MucB and mucus. A formula of $For(\Sigma)$ could be for example a first order logic formula whose atoms are comparisons between derivatives of x, y and z and real values, allowing formulae such as $(x' > 0 \Longrightarrow y' > 0)$, and so on. In fact, we shall see from Section 4 that temporal logics are well suited for biological regulatory networks.

So, by construction, a logic allows one to consider sets of models and, if the biological knowledge is formalized under a set of formulae Φ, then we can consider the subset $Mod(\Sigma,\Phi)$ of $Mod(\Sigma)$ whose elements are the models $M \in Mod(\Sigma)$ that satisfy Φ. If the considered logic is monotonic, then, by definition, adding a formula ψ when the knowledge increases will result in decreasing the considered set of models: $Mod(\Sigma,\Phi \cup \{\psi\}) \subset Mod(\Sigma,\Phi)$. Under the point of view of logic, it becomes obvious that a modelling process that offers a single model M_i at a time is risky: when the additional knowledge ψ comes to the surface, if $M_i \not\models \psi$ then one has to redo the job and offer a new model M_{i+1}. It is much more satisfying to extract information from the set $Mod(\Sigma,\Phi)$ as a whole, and to manage its evolution along the research process with biologists, according to the evolution of the biological knowledge.

3.2 A logical multidisciplinary research process

As shown in Section 2, when researchers in biology come to modelling, they know practically all the biological components involved in the model(s) and their potential interactions. In other words, they know the signature Σ of the problem. In practice, a few hypothetical variants of the signature are to be considered, so that a small number of signatures Σ_i is to be considered. Consequently, once the formal framework to manage the models is chosen, we know a first overapproximation of the set of possible models: $\mathcal{M} = \bigcup_i Mod(\Sigma_i)$. The set $\mathcal{M}$ (or by notation abuse $\Sigma = \bigcup_i \Sigma_i$) reflects intuitively the "*static knowledge*" of the biologists about the studied biological function (the set of components and their possible interactions is a knowledge which is not dynamically modified, although some interactions can become transitorily ineffective).

Biologists have also a "*dynamic knowledge*" about the behaviour of the biological system. For example, "If the operon is highly expressed then, after a certain delay, a mucus production is observed" or possibly "If the operon is not expressed (and no stress is applied) then it will never be expressed". These are properties that readily reduce the set of models to consider. Of course we must use a logic allowing to express these dynamic properties, so that biological knowledge of that kind can be collected into a set of formulae $\Phi \subset For(\Sigma)$. Then $Mod(\Sigma,\Phi)$ is the set of models that we have to manage. So, the signature Σ reflects the static knowledge about the biological system whilst the formulae of Φ reflect the known dynamic properties of the system.

In addition, according to Popper [Pop63], researchers in biology spend time, efforts and money

to perform biological experiments because they have scientific *hypotheses* to validate/falsify. These are most of the time hypotheses about the dynamics of the biological system (indeed, the "static hypotheses" are already embedded into the collection of possible signatures Σ_i). Let us denote H this set of "dynamic hypotheses."

Having $\mathcal{M}$ in the one hand, and $\Phi \cup H$ on the other hand, the modelling requirements become fully obvious:

- Firstly, one needs to prove that $Mod(\Sigma, \Phi \cup H)$ is non-empty. At first glance, it means to establish the **consistency** of the hypotheses.
  - In practice, the biological properties are often very subtle, context-dependent and subject to exceptions, so that even the formalized knowledge Φ can be inconsistent. Consequently in practive, we always begin by proving the consistency of the knowledge (i.e., $Mod(\Sigma, \Phi)$ is non-empty). This preliminary step is empirically very useful and highly instructive because it forces to make explicit any restriction or exceptional cases.
  - The consistency of $\Phi \cup H$ comes after this preliminary step.
- Then, one needs to **validate or refute** the hypotheses because the fact that $Mod(\Sigma, \Phi \cup H)$ is non-empty does not imply that "the true" biological model belongs to $Mod(\Sigma, \Phi \cup H)$. This model belongs *a priori* to $Mod(\Sigma, \Phi)$ but there is no certitude that it satisfies H. As already mentioned, assuming that the biological object satisfies the knowledge Φ, the Popper's approach turns this step into the design of "wet" biological experiments whose results maximize the chance to refute H.

So, it follows that the methodology to accompany biology with formal methods becomes also fully obvious:

- One has first to inventory the *static knowledge* about the biological system under study. This gives the signature Σ of the underlying general logic (possibly several variants Σ_i).
  In practice, this first step has usually been prepared by biologists before asking the help of mathematicians or computer scientists. However, it often catalyses the first multidisciplinary discussions, mainly giving a finishing touch due to simple predictions issued from domain specific theorems that already roughly link behaviour and global interactions (e.g., a homeostasis requires a negative cycle, and so on).
- The next step is to inventory the *dynamic knowledge* about the biological system. This gives a set of formulae Φ (and consequently this defines the set of *a priori* possible mathematical models $Mod(\Sigma, \Phi)$).
  In practice, we formally encode the main properties established in the litterature, of course with the help of biologists who gather relevant information and who help the modellers in abstracting it properly, thanks to a lot of multidisciplinary discussions. At the beginning of this step, it is often efficient to pick up arbitrarily a small number of models from $Mod(\Sigma, \Phi)$ and to discuss their biological meaning. At the beginning of the process, this requires biologists to explicitly describe why some models are not possible for "obvious reasons," leading to additional formulae in Φ which would otherwise stay implicit for a much longer time...
- An important step if to extract formally the *biological hypotheses* that motivate the biologists to conduct research on this subject precisely. This gives the set of hypotheses H (and

if we admit that there is one model M_{bio} in $Mod(\Sigma, \Phi)$ that represents the behaviour of the true biological object[5], then the main question is "does M_{bio} belong to $Mod(\Sigma, \Phi \cup H)$?").
In practice, biologists are at first apprehensive of this step, due to lack of familiarity with abstraction. We often have to remind that only behavioural properties are desired, we do not need detailed hypotheses about molecular affinities for instance.
The choice of the underlying formal framework is important for these first steps. Section 4 describes a discrete modelling approach based on ideas due to René Thomas, combined with the temporal logic CTL. This framework has proven very effective.

- Proving the *consistency* (i.e., $Mod(\Sigma, \Phi \cup H) \neq \emptyset$) is already a long process by itself because, in practice, the first versions of Φ and H are actually inconsistent! Nevertheless, this step is very interesting because the size and, more importantly, the complexity of the models under consideration do not allow a human brain to dominate the models. The inconsistencies come from very special cases that only a computer can notice owing to an exhaustive exploration of the possible cases.
  Often, the first versions of Φ itself are inconsistent because when encoding the behavioural properties, context-dependent restrictions are sometimes left implicit by biologists. Moreover, we manipulate most of the time *ad hoc* logics with restricted expressive power (in order to preserve completeness as already mentioned) and it can be sometimes partly misleading for a subtle behaviour. All in all, after the consistency step, biologists, mathematicians and computer scientists get a deeper concrete understanding of the model. It is not rare that biologists slightly change their perspective and their hypotheses after this step, without any additional experiment.
  Section 5 describes the technical aspects of this step for gene networks.

- Lastly, when both knowledge Φ and hypotheses H have stabilized, comes the *validation* step. One should design biological experiments that could potentially exhibit a behaviour that is incompatible with the hypotheses H. There is a striking similarity with software testing: in both cases the question is to put the [software/biological system] in carefully chosen contexts where they exercise some aspects of the [specification/hypotheses]. Noticeably, an important difference is the number of experiments that can be performed.
  In practice, the ability of formal methods to exhaustively consider the possible cases in a causality chain or within a reasoning strategy, allows us to produce shapes of experiments that biologists consider very interesting. Again, the complexity of the models under consideration do not allow a human brain to inventory all the revealing contexts, but theorem proving algorithms do. Nevertheless, finding good heuristics to choose the best experiments among the possible cases remain an active research subject.
  Section 6 describes in more details a formal frame that can help the design of correct heuristics. It is independent of the underlying general logic.

Logic has the priceless advantage that syntactic formulae bridge the gap between the mathematical models and the biological object: each experiment reveals an elementary formula (usually a closed formula) about the behavior of the biological object. In other words, there are formulae φ such that $M_{bio} \models \varphi$ can be decided *via* a wet experiment.

[5] Of course we do not know which is the model M_{bio}, indeed that is the question.

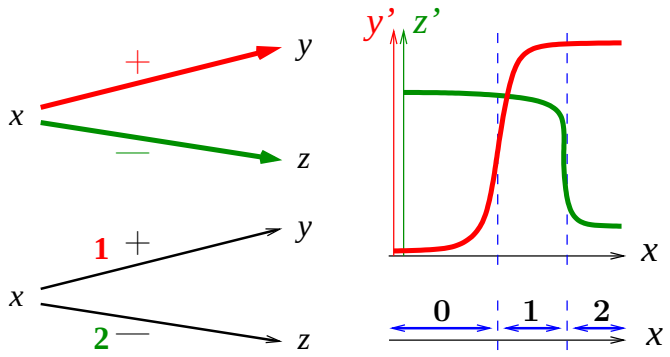
4 Thomas and Sifakis should have met

In this section we focus on gene regulatory networks which constitute a particular case of biological systems. In such systems, the non-linearities of the gene regulations are here largely exploited in order to discretize the continuous phase space of concentrations leading to a discrete phase space. The set of discrete states has additionnaly to reflect the topological relationships between concentrations (no "jump" is possible between distant concentration levels). This point of view has been initiated in the 70's by René Thomas who first proposed a framework to study the discrete aspects of gene regulations [Tho78, TdA90]. This discretization opens the door to simple formal methods since it transforms models into particular automata.

4.1 Thomas' multivalued approach

These discrete aspects of gene networks can be explained by slicing the concentration space of each chemical species into different intervals within which one can observe a uniform behaviour. More precisely, each chemical species which plays a role in a gene network has highly non-linear actions on its targets. These non linearities take place in a very narrow domain which allows a qualitative separation of the behaviour: when the concentration is below this non-linearity from its behaviour when it is above it. When a gene x activates a gene y, there is a threshold

Figure 4: Sigmoidal behaviour of gene responses



The contribution of a gene x to the synthesis speed of its targets depends on the concentration level of its product (x axis). This contribution is a increasing (resp. decreasing) sigmoid if x activates (resp. inhibits) its target. The inflexion point of the sigmoid depends solely on the considered target, thus slicing the possible expression levels of x into several intervals where its contributions are qualitatively uniform: 0 = no action, 1 = activation of y, 2 = both activation on y and inhibition on z.

such that x has a uniformly low action on y before the threshold, and it has a uniformly positive action after the threshold (see red curve in Figure 4). This threshold depends only on the target y and the curve (called a sigmoid) is almost a step function. When a gene x inhibits another gene z, the sigmoid is simply decreasing (see green curve in Figure 4). The y-threshold and the z-threshold have no reason to be equal, so, as shown in Figure 4, they slice the concentration space of x into 3 intervals, which we conventionnaly number from 0 to 2. More generally, the number of different qualitative expression levels depends only on the targets of the considered

gene and we note $0, 1, 2 \dots b_x$ these different qualitative expression levels where b_x is called the bound associated with gene x.

Such a rough counting of expression levels for genes is in fact adequate with respect to the experimental measurement capabilities in cell biology, where a measurement of continuous expression levels is often impossible. It is not rare that the experimental data for a gene can be summarized as "expressed" or "not expressed." Such qualitatively described expression levels correspond to the qualitative intervals introduced by René Thomas.

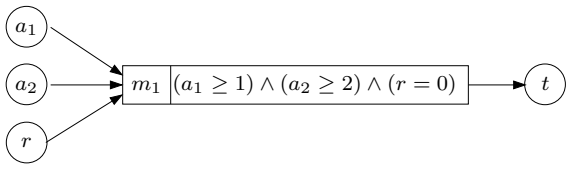
Let us also stress that the knowledge of individual interactions is not sufficient to deduce the behaviour of the entire gene network because the relative strengths of interactions can be unknown.

In addition to the Thomas' approach, when the combined effect on a common target of several actions is known, it is crucial to take into account such knowledge in order to constrain the relative strengths between interactions. For example let us consider a target t under the control of three regulator a_1, a_2 and r. Let us suppose that a_1 and a_2 are activators because they forms a complex which is itself the activator of t and that when r is present, the target t is repressed whatever the complex a_1a_2. It is clear that the interaction $a_1 \to t$ (resp. $a_2 \to t$) has not to be considered alone and that the activation of t is possible only if both activators are present and if the repressor is absent. This knowledge can be summed up by the following logical formula which specifies the conditions of the activation of t:

$$(a_1 \geq 1) \wedge (a_1 \geq 1) \wedge (r = 0)$$

where $a_1 \geq 1$ (*resp.* $a_2 \geq 1$, $r = 0$) means that the abstract expression level of a_1 (*resp.* a_2, r) is at least 1 (*resp.* at least 1, equal to 0). Such a formula represents a possible action on a particular target. This kind of information is then encoded into the interaction graph by a particular type of nodes called *multiplexes*, with which are associated the logical formulae expressing the conditions under which the corresponding regulation takes place. The targets of multiplexes are the genes, on which the actions have an effect, see Figure 5.

Figure 5: Representation of a simple multiplex



Each multiplex is labelled by a name (m_1) and a formula specifying the condition under which the regulation takes place. The target of each multiplex is pointed by an arc whose source is the multiplex. The participation of a gene to the multiplex is represented by an arc pointing on it.

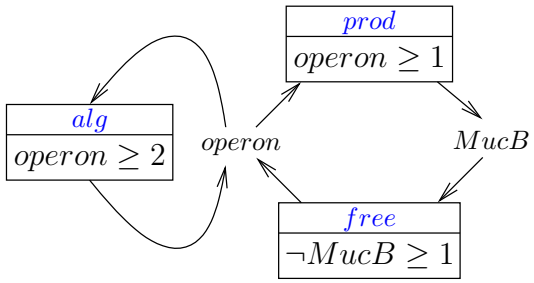
First-order logic is sufficient to formalize such knowledge within a multiplex. Even better, because of the finite number of qualitative expression levels, all the terms of the logic can be represented by a finite number of atoms, so, propositional logic is sufficient.

We can now define an *interaction graph* as a graph $\Sigma = (V \cup M, A)$ where

- the set of nodes is the union of the set V of genes to consider (also called *variables*), each variable $v \in V$ being labelled by its bound b_v, and the set M of multiplexes, each multiplex m being labelled by the formula φ_m expressing the conditions under which the regulation takes place,
- and the set of arcs A is made of arcs from multiplexes towards targets and arcs from genes towards multiplexes reflecting the participation of genes to multiplexes: $A \subset (V \times M) \cup (M \times V)$.

The interaction graph corresponds to the signature Σ of the considered general logic. Figure 6 represents the interaction graph associated with the mucus operon of *P. aeruginosa*.

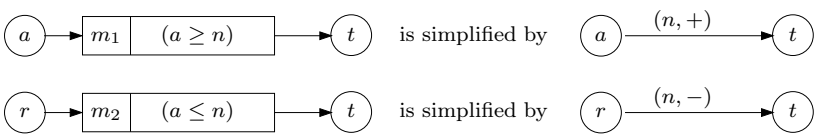
Figure 6: Mucus operon of *P. aeruginosa*



The multiplex *alg* represents the production of alginate *AlgU* by the *operon* at the threshold 2. The multiplex *prod* represents the production of $MucB$ by the *operon* at the threshold 1. The multiplex $free$ represents the absence of the inhibition induced by $MucB$. The mucus production of Figure 2 is ignored here (as *mucus* is entirely under the control of the *operon*).

The notion of multiplex is helpful for describing the cooperation or concurrency between regulators of a common target. But when a unique activator (resp. inhibitor) acts on its target, the multiplex makes heavy the representation of simple regulation. Thus we adopt in such a case a graphical notation (see Figure 7): in order to represent an activation (resp. inhibition) of a gene directly on another one, one draws an arc from the regulator towards its target (without intermediate multiplex) and one associates with this arc the couple $(s, \pm)$ where s is the discrete level from which the regulation takes place, and $+$ or $-$ is the sign of the regulation ($+$ for activation and $-$ for inhibition).
Using these graphical simplifications, Figure 6 simplifies as the black graph in Figure 2 except that the thresholds must be added.

Figure 7: Graphical convention for simple regulations



Because of the discretization described above, a discrete *state* of a gene network simply associates, to each variable $v \in V$, a qualitative expression level which is an integer belonging to $[0, b_v]$. The vector of these qualitative expression levels is called a *discrete state* of the gene network. Then, the qualitative dynamics of the gene network can be represented by transitions between these states. The possible evolutions from a current state η are controled by the set of interactions which take place at η, that is, the set of multiplexes m such that φ_m is satisfied for the state η. Thus one associates with each gene $v \in V$ a family of parameters, named $K_{v,\omega}$ where ω is a set of multiplexes whose target is v. Intuitively, the value of the parameter $K_{v,\omega}$ indicates the expression level towards which v is attracted when the set of *active* multiplexes controling v is ω (*active* means that the associated formula φ_m is satisfied in the current state).

Let us remark that experimental knowledge is helpful for the determination of these parameters. For example, let us assume that from an initial qualitative state η, one observes that the concentration of a particular gene product v increases. This qualitative observation can be transformed in terms of constraints on the parameter $K_{v,\omega}$ that applies on the considered initial state η : we have necessarily $K_{v,\omega} > \eta(v)$ (the local attracting value is greater that the current state of v).

We then call *gene regulatory network* the couple made of an interaction graph $\Sigma = (V \cup M, A)$ and the following family of parameters

$$\mathcal{K}_\Sigma = \{K_{v,\omega} \ \mid \ v \in V \text{ and } \omega \subset \Sigma^{-1}(v)\}$$

where $\Sigma^{-1}(v)$ is the set of predecessors of v in the interaction graph.

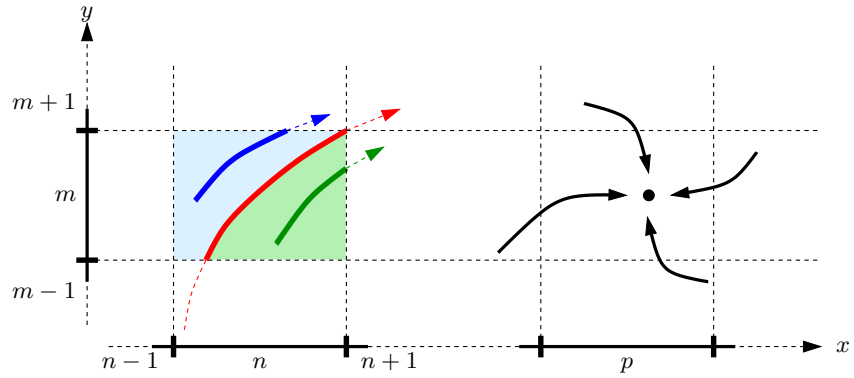
In the context of general logics, the set of all possible valuations of the parameters constitutes $Mod(\Sigma)$. For a signature (i.e. an interaction graph) Σ, the set of models is the set $Mod(\Sigma)$ of all possible assignments of the family $\mathcal{K}_\Sigma$ such that the value assigned to $K_{v,\omega}$ is an integer belonging to $[0, b_v]$. By notation abuse, a model being given, we write $K_{v,\omega} = n$ (the assignment is left implicit) and n is called the parameter value.

From the parameters' values, one constructs the dynamics of the system. For each possible qualitative state and for each gene, one computes the set of active multiplexes which act on it. The transition from qualitative state q_1 to q_2 exists if the three following conditions are satisfied:

1. q_2 differs from q_1 by only one component: $\exists! v \in V, q_1(v) \neq q_2(v)$
2. q_1 and q_2 are adjacent in the qualitative state space, that is $|q_1(v) - q_2(v)| = 1$
3. the direction of evolution in the direction v leads the system to come closer to $K_{v,\omega}$, that is: $sign(K_{v,\omega} - q_1(v)) = sign(q_2(v) - q_1(v))$ where ω is the set of active multiplexes acting on v for the current state q_1.

Condition 1 expresses that, in the discrete abstraction, it is not possible for two components to evolve simultaneously. In other words, the discretization consists in a desynchronization of the continuous dynamics: The directions of evolution are preserved but there exists only one product which evolves at a time. This can be interpreted by the fact that the probability that the continuous dynamics reaches simultaneously two (or more) thresholds at the same time is null. See Figure 8.
Condition 2 expresses that the discrete states constitute an abstraction of intervals of continuous concentration space: it is not possible to pass directly from abstract level n to $n + 2$ without passing through level $n + 1$.
Finally Condition 3 constrains the direction of evolution.

Figure 8: Discretization of a continuous state space



Assuming that the local dynamics make both x and y increase (qualitative state (n, m)), most of the continuous trajectories cross only one of the faces of this discrete state (green: $(n, m) \to (n + 1, m)$ and blue: $(n, m) \to (n, m + 1)$). The probability to cross both faces simultaneously is null (red curve). In a stable state (see (p, m)), the continuous derivatives do not have the same sign everywhere in the considered qualitative intervals.

Pragmatic construction of a state graph:
Let us construct the state graph from the interaction graph of Figure 6. We first have to determine the bounds of both *operon* and $MucB$. We deduce from the figure that $MucB$ has only two relevant qualitative expression levels because it has an action on a unique target (the *operon via* the multiplex $free$): the first level (0) corresponds to the situation where the $MucB$ does not inhibit *operon* whereas the second (1) correspond to the situation where it does. Thus $b_{MucB} = 1$ is a proper choice. We also deduce that *operon* has three qualitative expression levels since it has an action both on itself and on $MucB$ (*via* the multiplexes *alg* and *prod*). Thus the bound of operon is $b_{operon} = 2$. Moreover, considering that no information is available for constraining the combined actions of multiplexes *alg* and $free$ on *operon*, six parameters are needed to describe the dynamics: $K_{operon,\{\}}$, $K_{operon,\{alg\}}$, $K_{operon,\{free\}}$, $K_{operon,\{alg,free\}}$, $K_{MucB,\{\}}$ and $K_{MucB,\{prod\}}$.

The next step is to build a table which associates with each possible discrete state and with each gene, the set of regulations which are active on this gene at this state, see Figure 9:

1. Multiplex *alg* is active when *operon* is at a discrete level greater or equal to 2. Then it is an active multiplex of *operon* for each line where the discrete level of *operon* is equal to 2.
2. Multiplex *prod* is active when *operon* is at a discrete level greater or equal to 1. Then it is an active multiplex of $MucB$ for each line where the discrete level of *operon* is greater or equal to 1.
3. Multiplex $free$ is active when $MucB$ is at the discrete level 0. Then it is an active multiplex of *operon* for each line where the discrete level of $MucB$ is equal to 0.

Let us consider now particular parameters' values : $K_{operon,\{\}} = 0$, $K_{operon,\{alg\}} = 2$, $K_{operon,\{free\}} = 2$, $K_{operon,\{alg,free\}} = 2$, $K_{MucB,\{\}} = 0$ and $K_{MucB,\{prod\}} = 1$. We could have chosen another

Figure 9: A resource table for the mucus production in *P. aeruginosa*

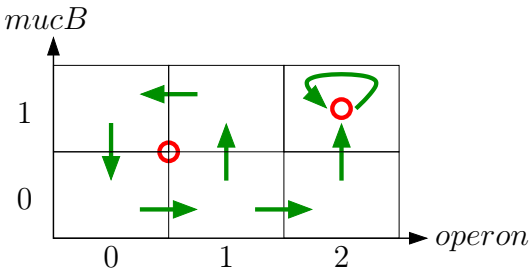
operon	$MucB$	active multiplexes acting on *operon*	active multiplexes acting on $MucB$	$K_{operon,...}$	$K_{MucB,...}$
0	0	$\{free\}$	$\{\}$	2	0
0	1	$\{\}$	$\{\}$	0	0
1	0	$\{free\}$	$\{prod\}$	2	1
1	1	$\{\}$	$\{prod\}$	0	1
2	0	$\{alg, free\}$	$\{prod\}$	2	1
2	1	$\{alg\}$	$\{prod\}$	2	1

The table is more rapidly built *by columns*.

example, here we have deliberately chosen a case where *operon or* $MucB$ are sufficient to act on the *operon*. Knowing these values, we can complete the last two columns of Figure 9. According to the desynchronization process explained before, we construct the state graph as in Figure 10.

This representation of state graphs highlights the relationships between the different qual-

Figure 10: A state graph for the mucus production in *P. aeruginosa*



The state graph associated with Figure 9.

itative states. For example, qualitative state $(1, 1)$ is a neighbour of $(2, 1)$ even if there are no transitions between them, this notion of neighbourhood comes naturally from the discretization of concentrations, and is well illustrated by a Manathan distance between qualitative states. This is the reason why the qualitative states are classically represented on the grid $\mathbb{N}^n$. Thus a state abstracts a continuous region of concentration space (Figure 8). Let us remark that when one of the parameters $K_{v,\omega}$ is not equal to the component v at the current state, the sign of the derivative of v does not change in the continuous domain corresponding to the discrete state (see state (n, m) in the figure). If the parameter $K_{v,\omega}$ is equal to the component v of the current state, then, the sign of the derivative of v does change in the domain (state (p, m) in Figure 8).

To summarize, the dynamics is thus represented by the (discrete) *state graph* whose set of nodes is the set of possible states, and the transitions are those deduced from the parameter values as shown previously. A gene regulatory network is consequently modelized by three elements:

- an interaction graph $\Sigma = (V \cup M, A \subset (V \times M) \cup (M \times V))$ where V is the set of variables (genes) to consider (labelled by their bounds), M the set of multiplexes (labelled by the formulae expressing the conditions under which the regulations take place), and the set

A is made of arcs from multiplexes to variables and of arcs from variables to multiplexes reflecting the participation of a gene to a multiplex formula.

- the family of parameters $K_{v,\omega}$ where v is a gene and ω is a subset of multiplexes acting on v,
- the state graph whose nodes are the possible discrete states and transitions are the desynchronizations shortened to the neighbourhood as explained before.

This discretization transforming continuous concentrations into qualitative expression levels leads also to a discretization of time. Indeed the discretization leads to focus on the changes of qualitative states, and leads to "measure" time only by the number of changes in the state graph. The time is taken into consideration only in a logical manner. This representation of time is said chronological and contrasts with the chronometrical approach of continuous differential systems. On the one hand it is clear that the R. Thomas' approach constitutes an approximation of the biological system in which the high complexity leads to much more richer behaviours. On the other hand the classical modelling framework of differential systems is also an approximation because (i) a cell cannot be considered as a homogeneous space in which molecules evolve continuously and (ii) in some systems the number of molecules which control a change of behaviours is very small, sometimes less than one molecule per cell in average...
Both these modelling frameworks are nevertheless compatible. It has been proved that when non-linearities are represented by sharp sigmoids, the model can be approximated by a system of piecewise linear differential equations, and that there exist some discrete models that present the same stationary states than these continuous models [Sno89].

This discrete framework exploits, as much as it can, all the non linearities of biological systems, but the discrete approach can also be used when some of regulations are linear. In the example of Figure 2, the regulation of $MucB$ on *operon* is linear since the biological phenomenon is a capture of $AlgU$ by $MucB$ which is a membrane protein. Nevertheless the discretization is valid because the composition of a linear function with a sigmoid function leads to a sigmoid function. In our example the composition of regulation of $MucB$ by *operon* and regulation of *operon* by $MucB$ preserves the existance of a sharp non linearity.

4.2 Temporal logic applied to gene networks

As in the context of differential modelling, from a practical point of view, the major problem remains the determination of the parameters' values. Nevertheless this identification problem, in our context, is much simpler than in the continuous case because whereas in the differential context, the number of parameterizations is infinite and non-countable, in our context a finite enumeration is possible. This fact opens the door to model checking algorithms, constraint programming, and so on.

In some cases the identification problem can be solved manually. Let us consider the *P. aeruginosa*'s mucus production system and the table of Figure 9. If biological knowledge is sufficient to deduce that state $(2,1)$ is stable, we immediatly deduce that $K_{operon,\{alg\}} = 2$ and $K_{MucB,\{prod\}} = 1$. If we know that from state $(0,1)$ the system goes towards $(0,0)$, and that from the state $(0,0)$ the level of *operon* increases towards 1 and can go to (2,0) from (1,0), we deduce that $K_{operon,\{\}} = 0$, $K_{MucB,\{\}} = 0$, and $K_{operon,\{free\}} = 2$. Finally the hypothesis that the steady state $(2,1)$ is reachable from $(2,0)$ leads to the value of last parameter: $K_{operon,\{alg,free\}} = 2$. These parameters' values lead to the state graph of Figure 10.

It is then natural to apply some tools coming from temporal logics in order to question the models. This natural idea has been developped 30 years after the seminal works of R. Thomas and co-workers [BCRG04], as well as pruning the set of models by taking into account cooperation or concurrencies between regulations [KCRB09]. The first idea was to enumerate all possible valuations of parameters, to construct the state graphs and to get rid of all models which do not satisfy known temporal properties of the *in vivo* system, see Subsection 5.1. The known temporal properties have to be transcripted in a formal logic, here we have chosen the temporal logic CTL [CE81, EH82], and testing whether specifications are satisfied by a particular model is done with classical model checking algorithms [HR00, Sif82, QS83].

The Computation Tree Logic (CTL) is a branching-time logic, meaning that its model of time is a tree-like structure in which the future is not deterministic; there are different paths in the future, and any one of them might be realised. It is well suited for the formulation of properties on non deterministic state graphs, such as the ones considered here. It permits us to express, for example, that some events occur before some others, that a specific event has to take place in order to reach a given state, that it is impossible to reach a given state or that an event is always possible, *etc.*

A CTL formula on a gene regulatory network is inductively defined by:

- atomic formulae are $\top$, $\bot$ or of the form $(v \geq n)$ where $v \in V$ is a variable of the gene network and $n \in [0, b_v]$
- if ϕ and ψ are formulae, then $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$, $AX\phi$, $EX\phi$, $A[\phi U \psi]$, $E[\phi U \psi]$, $AG\phi$, $EG\phi$, $AF\phi$, $EF\phi$ are formulae.

$\top$ is the always true formula; $\bot$ is the always false formula; $(v \geq n)$ is true iff, in the current state, the concentration level of the gene product v is greater or equal to n; $\neg, \wedge, \vee, \Rightarrow$ are the usual connectives (respectively *not*, *and*, *or*, *implies*). All the temporal connectives are pairs of symbols: the first element of the pair is A or E followed by X, F, G or U whose meanings are given in the next table.

A	for **A**ll paths choices	X	ne**X**t state
E	for at least one path choice (**E**xist)	F	some **F**uture state
		G	all future states (**G**lobally)
		U	**U**ntil

Let us notice that, according to our general logic point of view, the CTL formulae constitute the set $For(\Sigma)$ which defines the syntax of the logic. The set $For(\Sigma)$ depends on the signature Σ since the atomic formulae of the form $(v \geq n)$ depend on the interaction graph (v has to be a variable in the graph) and on the bound of the considered gene.

The satisfaction relation $\models$ says whether a model M satisfies a formula ϕ ($M \models \phi$). This binary relation is simple enough to be directly decidable by a classical *Model checking* algorithm. Given a state graph (deduced from a model as explained in the previous subsection), this algorithm exhaustively and automatically checks whether this model meets a given set of CTL formulae.

Let us come back to example of mucus production system in *P. aeruginosa*. It is now established that *P. aeruginosa* exhibits two attraction basins. The first one produces mucus and the second does not. The production of mucus takes place at a level of *operon* greater than some

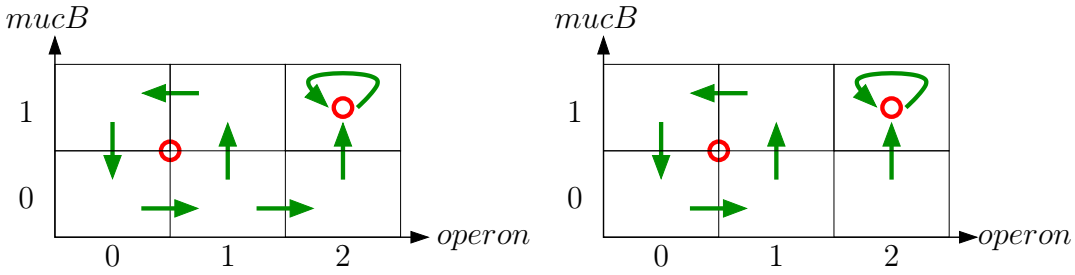
unknown threshold. So, necessarily, the system produces mucus if the expression level of *operon* is equal to its maximal value 2 and it does not if it is equal to 0. We can express the two possible behaviours by the following CTL formulae:

(1) $$x = 0 \Longrightarrow AG(\neg(x = 2))$$

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

The first formula expresses that from a very low initial level of operon, it is not possible to reach a high level where mucus is produced, and the second formula expresses that from a saturated level of operon it is not possible to reach a very low level. Figure 11 presents two possible state graphs: the left one satisfies only the second formula; the right one satisfies both formulae.

Figure 11: Two possible state graphs



(Left) with parameters of Figure 9 (Right) one parameter changes: $K_{operon,\{free\}} = 1$

Nevertheless some temporal and behavioural properties cannot be transcripted into CTL. For example, it is not possible to express in CTL, with no loss of generality, that the state graph contains a given number of attraction basins. The reason is that the CTL logic has only 2 modal connectors handling the different paths from the current state (E and A). These two connectors are not sufficient to count the number of paths satisfying a particular property. But if one adds some biological knowledge like the approximative position of the lines separating two attraction basins, it becomes possible to use CTL in order to specify this knowledge. It is exactly what we did in the previous temporal formulae 1 and 2.

From a practical point of view, one must admit a few difficulties encountered when designing CTL formulae. First of all, one needs an expertise in the biological field of interest: when a temporal property is put on the table by a biologist, it is very important to analyse the (often non-specified) context, in which this property is observed, in order to take into consideration such a context in the CTL formula. In a similar way, some trivial properties can be forgoten, simply because they seem so obvious for biologists that they do not mention them. However these trivial properties are mandatory to select the models. Also, one needs to well understand the CTL syntax and semantics. Although the syntax of CTL is simple and the intuition of the semantics is straightforward, some subtilities can occur when manipulating CTL. For example the left state graph of Figure 11 does not satisfy $x = 0 \longrightarrow AG(\neg(x = 2))$ and thus one would get rid of it. Nevertheless this model is interesting because it has two different behaviours: a loop at right and a fixed point at left. Obviously the transition from $(1, 0)$ to $(2, 0)$ allows one to go from the loop towards to steady point. The question is then to know if the loop at right corresponds to an incoming spiral or to an outcoming spiral. In the second case, the model would have to be thrown away. Further study is required to precisely answer this question, but at the level of this abstract study, one has to keep this model.

Let us remark that we did not define the binary relation *entails* ($\vdash$). In our context, the binary relation $\vdash$ is not defined by a tree whose nodes are elementary deduction rules, but is obtained by model checking of the formula on all possible models. This is the role of our software SMBioNet [KCRB09] which takes as input a formula and a interaction graph and verifies if the formula is satisfied for all possible state graphs built from the interaction graph Σ. The formula is true for the interaction graph ($\vdash_\Sigma \phi$) iff all possible state graphs built from Σ satisfy the formula ϕ. Moreover we have $\Phi \vdash_\Sigma \phi$ iff all possible state graphs satisfying Φ satisfy also ϕ.

From our experience of collaborative work with biologists during more than 15 years, CTL is sufficient most of the time, since indeterminism can be transcripted into CTL formulae, and since recurrent behaviours or stationary ones can be coded in this formal language. Nevertheless CTL is only an example of temporal logics, a similar approach with another kind of temporal logics like LTL [Pnu77] would also suit.

5 Consistency of biological hypotheses

5.1 The brute force approach

After having transcripted the biological knowledge Φ and hypotheses H into temporal formulae, the first possibility to test the consistency of biological hypotheses is to take advantage of the finite number of parameters' valuations. Indeed, as model checking is able to verify whether a formula is satisfied in a model, we can enumerate all possible models and for each of them verify whether the conjunction of all hypotheses and knowledge is satisfied. Obviously, as soon as one has found one parameters' valuation which leads to a dynamical model which satisfies the specifications, one deduces that the hypotheses are consistent with the knowledge, since $Mod(\Sigma, \Phi \cup H) \neq \emptyset$.

Let us remark that this enumeration is mainly used as a parameter identification method even if it also allows one to discard some "static" interaction graphs simply by lack of compatible parameters.

Although we gave up simulations and their associated "brute force" computational approach, we clearly implement here another intensive computation approach. In the context of gene regulatory networks, the number of parameters' valuations is finite but it can be huge because the number of parameters associated with each gene depends exponentially on the number of its predecessors: if a variable v has n multiplexes acting on it, there are 2^n parameters because one has to consider all subsets of predecessors. Moreover, if we do not have information about the cooperation/concurrency of predecessors, these parameters are independent, and the number of possible values for each of them is b_v+1 where b_v is the bound associated with v. So, the number of parameters and the number of possible parameters' valuations are respectively

$$\sum_{v \in V} 2^{|\Sigma^{-1}(v)|} \qquad \text{and} \qquad \prod_{v \in V} (b_v + 1)^{2^{|\Sigma^{-1}(v)|}}$$

where $\Sigma^{-1}(v)$ is the set of predecessors of v in the interaction graph Σ. For a gene regulatory network with 10 genes in which the bounds associated with genes are 2 and the in-degree of each gene is also 2, the previous formula gives $(3^4)^{10} \approx 1.2 \times 10^{20}$ possible parameters' valuations... Thus, performing model checking exhaustively on all models would be actually computationally expensive.

In practice the number of models is not necessarily the main limiting factor. Often, the

enumeration can be considerably reduced at the price of hand-made work which can prune large parts of the set of models. For example, when the first filter selects too many models, it is very instructive to choose at random some dynamics or some traces on these dynamics, and to analyse with a biologist the reasons why they have to be accepted or rejected. The formalization of these reasons allows one to enrich the knowledge and therefore reduces the search space. In addition, a lot of theorems are known, which establish strong constraints on parameters from a few biological knowledge.

For example, the Snoussi's hypothesis states that when the set of multiplexes ω_1 is included in the set of multiplexes ω_2 then $K_{v,\omega_1} \leq K_{v,\omega_2}$. It means that when one adds some regulations that help a target to be expressed, the associated parameter cannot be decreased. This hypothesis comes from the comparison of discrete and continuous frameworks [Sno89]: without this condition on parameters, the discretization does not always preserve the stationnary states. Note that, sometimes, the Snoussi's hypothesis is not applicable. It is the case for example when two different activators act together as an inhibitor if they simultaneously present.

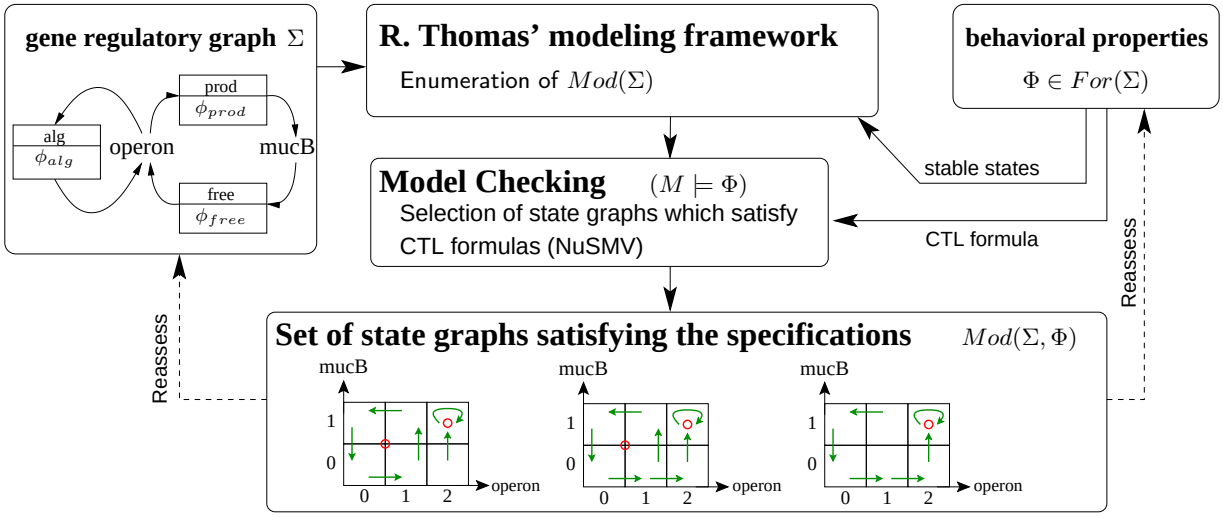
We can also take into account the functionality of feedback loops. A positive (resp. negative) feedback loop[6] is said "functional" when it gives rise to a multistationarity (resp. a homeostasis) of the system. René Thomas [Tho81] stated two general rules on dynamical systems. Informally, the first (resp. second) rule asserts that the presence of a positive (resp. negative) circuit in the interaction graph of a dynamical system is a necessary condition for the presence of several stable states (resp. sustained oscillations). These conjectures have been proved later on, in different modelling frameworks, see [RC07, Ric10] for the proofs in the multi-valuated discrete case. Thus if we know that a homeostasis is present in the system, and that this homeostasis implies a particular gene, we deduce that there exists a negative functional feedback loop which implies this gene. And what reduces again the search space is that the functionality of a feedback loop leads to strong constraints on parameters, see the notion of characteristics states in [TTK95].

We developped a software platform that selects, in the context of Thomas' modelling, the models that satisfy a conjunction of properties expressed in CTL [KCRB09], see Figure 12. For each parameterization enumerated by `SMBioNet`, it constructs the corresponding state graph, and checks whether the CTL temporal formula is satisfied by this state graph. This verification step is performed by the model checker NuSMV [CCG+02]. Only parameterizations leading to state graphs satisfying the given behavioural properties are retained. If none of them are retained it is necessary to reassess either the regulatory graph (it can be too simple to be able to lead to a state graph that expresses the specified properties) or the temporal properties.

It is in fact possible to enumerate all models which satisfy the specification $Mod(\Sigma, \Phi)$ without enumerating $Mod(\Sigma)$: In [FCT+04] L. Trilling proposed to use logic programming with constraints in order to identify the parameter values. The method extracts all the parameter values that make possible a given set of observed paths in the state graph. The method has also been extended and implemented by F. Corblin [CTF+09], and the results are impressive. Provided that the temporal properties under consideration can be expressed *via* a finite number of paths of fixed length, a few seconds of computing time are needed for problems where `SMBioNet` alone would need several hours. The idea is to specify in PROLOG the Thomas construction of the state graph, according to symbolic representations of the $K_{...}$ parameters. Then, by specifying that a given path exists in the state graph, PROLOG generates the constraints on the parameters that permit each transition of the path. Lastly, constraint solving algorithms

[6] A feedback loop is said positive (resp. negative) when the direct or indirect action of a gene on itself is positive resp. negative). It is positive if the number of negative interactions along the cycle is even, and negative when this number is odd.

Figure 12: `SMBioNet`**'s fundamentals**



From a gene regulatory graph and some behavioural properties, the tool SMBioNet is able to enumerate each possible parameterization and to select only those that lead to a state graph which is coherent with the temporal properties.

try to exhibit parameter values or to prove inconsistencies. Thus temporal specifications can be viewed as constraints on parameters $K_{...}$ and the constraint programming paradigm allows one to overcome the limitations due to enumeration. All together make it possible to automate the parameters' identification.

Another alternative to enumeration is based on Hoare Logic. Wet experiments are viewed as particular executions of a program leading to particular *post-conditions* at the end of the experiments, and Hoare logic and weakest preconditions [Dij75] give a simple algorithm to construct the minimal conditions that are needed to imply these conditions, see 6.

5.2 Model simplifications

We have shown in the previous subsection that the number of possible parameterizations is growing exponentially with the size of the models and with their connectivity. So, a powerful way to reduce the computing time needed to establish consistency is obviously to manage smaller models. Such an idea often appears suspicious to biologists: they have the well established empirical knowledge that lot of "small details" cannot be simplified because they drastically influence the behaviour of the studied biological object. Consequently, for biologists, the more detailed the description, the better the model, and this has a great impact when establishing together the first set of possible models. These models contain many elements that seem redundant to mathematicians and computer scientists. However, from the methodological point of view, it is necessary to start from sufficiently detailed models in order to master the credibility gap between the biological question and the considered mathematical models. Of course the modelling activity is inherently a simplification and an abstraction process, but the first description must carefully transcribe the basic elements that biologists consider involved in the studied behaviour.

Once the first set of possible models has been established, as well as the formalized knowledge Φ and the hypothesis under consideration H, one can however perform lot of simplification steps before establishing consistency. For instance:

- if there is a long chain of variables and multiplexes in the interaction graph that have only one predecessor and only one successor,
- if none of the variables in the chain appear in the formulae of $\Phi \cup H$,
- and if the neXt modality X does not occur in the formulae of $\Phi \cup H$,

then it may be a wise decision to reduce the length of this chain by removing some intermediate variables and making the appropriate substitutions in the remaining multiplexes... a simple situation that is surprisingly rather frequent in practice.

In fact, there is a considerably high degree of freedom for model simplifications if we properly consider that our goal is simply to establish consistency. The answer we are looking for is indeed reduced to a boolean decision: *is* $Mod(\Sigma, \Phi \cup H)$ *empty?*. The previous "chain example" was an example where we simplified Σ (the interaction graph) into a simpler Σ' in such a way that:

- $(\Phi \cup H) \subset (For(\Sigma) \cap For(\Sigma'))$
- there is an obvious "forgetful functor" f from $Mod(\Sigma)$ to $Mod(\Sigma')$ such that for all formulae in $\Phi \cup H$, $M \models \varphi$ if and only if $f(M) \models \varphi$.

These two properties are very restrictive when compared to the simplicity of the boolean question. There is formally no limitation in modifying Σ, Φ and H. One can perform an elaborated sequence of simplifications (Σ_i, Φ_i, H_i) in such a way that $(\Sigma_0, \Phi_0, H_0) = (\Sigma, \Phi, H)$ and:

$$\forall i = 1..n, \quad (Mod(\Sigma_{i-1}, \Phi_{i-1} \cup H_{i-1}) = \emptyset) \Longleftrightarrow (Mod(\Sigma_i, \Phi_i \cup H_i) = \emptyset)$$

In other words, one can manipulate the models, the biological knowledge and the hypothesis under study *ad libitum*, provided that one does not change the answer to the consistency question. As a consequence, the final models that we formally study are "*Kleenex models*" because they have been simplified with respect to the studied biological question H. So, they do not necessarily reflect a biological "reality"; they solely abstract a behaviour with respect to the studied biological question. They cannot be reused for other biological questions. Nonetheless, the acquired expertise with respect to this biological object remains for the next modelling process.

In the current practice, modellers rarely take benefit of this large freedom. There are however a few research works in this direction for discrete models of gene networks:

- In [BT09], the case where the hypotheses only concern a subnetwork of the considered gene network is studied and a necessary and sufficient condition to preserve the behaviour of the subnetwork is established. Additional works have been done in [MAC+11], that weaken the condition by preserving only some formulae instead of the full behaviour of the subnetwork.
- In [NRT+09], a useful sufficient condition is given, that permits to remove some genes from a gene network without modifying the stable states of the system or the number of attraction basins.

- [BT09] and [NRT+09] have introduced an intermediate technical notion of "level folding," which can also easily be applied to reduce the number of thresholds within the possible levels of expression of a gene. Some thresholds can be ignored if they do not appear in the considered formulae, thus reducing the state space and facilitating the consistency proofs.
- Level folding reduces the number of states into the state graphs. Another efficient way to reduce the state graphs would be to reduce the number of transitions and/or rearrange them in such a way that model checking (among other algorithms) is facilitated. Although very difficult, this kind of subject has got recently a step forward *via* systematic studies about the degree of synchronism within automata networks and its impact on the global behaviour [Nou12].

We probably miss several other approaches here and we believe that the reduction of models is going to be a major research subject for dynamic systems in biology, because the size of the biological models grows much faster than the computer speed. One of the true difficulties with respect to this subject is to keep a trace of the successive simplifications in such a way that any sensible observation about the simplified models can be translated back into a sensible remark for biology.

6 Validation of biological hypotheses

The consistency of biological hypotheses being established, it proves that there are models that validate both the biological knowledge Φ and the hypotheses H. Of course, it does not prove that the biological object under study satisfies the hypotheses H, i.e., that it belongs to this non-empty set of consistent models.

In this section, we will assume however that the biological object satisfies the biological knowledge Φ: a reasonable assumption. There is in fact another hidden assumption: we assume that there is a way to abstract the behaviour of the true biological object into a mathematical model M_{bio} which belongs to $Mod(\Sigma)$ as defined in Section 3.1. This means that we assume that all biological experiments will exhibit a behaviour which is observationally consistent with the signature (i.e. with the interaction graph for gene networks). This assumption allows us to consider that, when a wet experiment establishes without ambiguity that the biological object satisfies a given property φ, one can definitely consider that $M_{bio} \models \varphi$. Of course, not all properties can be established by an experiment: φ is usually a simple formula whose semantics matches experimental capabilities (e.g., a closed formula). The next subsections give more insight about such "observable" formulae and their role in order to validate or refute H.

Of course, after an experiment has been performed, the corresponding observed formula φ can be added to the knowledge Φ, thus reducing the set of models to consider. That is exactly the reason why, when working with *researchers* in biology, we cannot use a modelling approach based on a single model that mimics the known biological properties: any new experiment could refute such a single model and, in such a case, what to do next?

6.1 "Wet" experiments and logic formulae

At first glance, extracting a logic formula that exactly transcribes the results of a given wet experiment may seem a difficult task. What a biological experiment shows is indeed a temporal trace of the system under the environment imposed by the experimental conditions. Moreover,

the experimental trace can be partial: often, only some of the involved genes can be simultaneously observed (using fluorescence or whatever) or experimental constraints do not allow a sufficiently dense timing of successive measurements, and so on.

In fact, if the environment of the biological object does not change during the experiment, formalizing an observed experimental trace is not so difficult. Using CTL for example, it mostly involves a large amount of nested EX statements or nested EF statements depending of the density of the timing of successive measurements. What is more difficult is to reflect experiments where the environment of the biological object is modified during the experiment. For example, if biologists perform a Knock Out of a gene in the middle of the experiment (e.g. using siRNA) then the biological model is not the same at the beginning of the experiment than at the end of the experiment: the gene network suddenly looses a gene. Other common examples can use modulations of the temperature for instance: one has to enrich the model accordingly by including the temperature as a new element of the model and carefully formalizing its action on each other element of the model.

When it comes to gene networks, we have made the choice to use a different formal method in order to extract formulae from experimental traces. Once again, formal methods gave us a significant contribution but we have not used temporal logic directly. We have rather "exhumed" an old algorithm that was decisive in the proofs of correctness for imperative programs, namely the weakest preconditions of *Hoare logic* [Hoa69, Dij75], and we have modified its inference rules in order to formalize entirely the Thomas approach described in Section 4. We have initiated the method in [Kha10] and we are currently enriching it with the help of Olivier Roux.

A simplified view of Hoare logic and weakest preconditions [Dij75] is to see them as an algorithm that starts from the property Q that we want to establish at the end of the program and that crosses backward each instruction of the program while maintaining the minimal property that is needed to imply Q at the end. So, we obtain successive properties: Q_1 just before entering the last instruction of the program, Q_2 just before entering the before last instruction, and so on. At last, the property Q_n obtained at the beginning of the program (before the first instruction) is the weakest precondition that is needed to ensure the postcondition Q. If we know a precondition P about the input variables of the program, the correctness of the program[7] is equivalent to $(P \Longrightarrow Q_n)$.

Here, we see an experimental trace as a "program." The elementary instructions of this "program" are the observed transitions in the state graph of the gene network. It means that for each gene there are thresholds (e.g. thresholds on observed fluorescence intensity) that separate the different discrete expression levels as described in Section 4. Consequently an elementary instruction is an assignment of the form "$x := x + 1$" or "$x := x - 1$" and it correspond to an actual observation at this timepoint of the experiment, where the gene x has increased (which we note $x+$ for simplicity), or respectively decreased (which we note $x-$ for simplicity). Such an observation tells us that there is a transition somewhere in the model M_{bio} where gene x has changed its expression level.

For example, let us consider the interaction graph of Figure 6 (Section 4), which formalizes with multiplexes the functionning of the operon of *P. aeruginosa* shown in Figure 2. Let us consider a wet experiment that ends with a state where no expression of the operon is observed and on the contrary the membrane protein $MucB$ is present. Then the postcondition is the conjunction

[7] Assuming that it terminates

$$Q \equiv \left\{ \begin{array}{l} operon = 0 \\ MucB = 1 \end{array} \right. .$$

Let us assume moreover that during the experiment, we have successively observed that $MucB$ switches from absent to present and that the operon has lowered its expression level (crossing a threshold), then the program that formalizes these observations is

$$p \equiv (MucB+; operon-)$$

(where the ";" stands for the sequential composition as usual).

According to Hoare logic, an assignment of the form "$x := expression$" is treated as follows. If Q_i has to be true after this instruction, then Q_{i+1} has to be true before this instruction, where Q_{i+1} is obtained from Q_i by substituting each occurrence of the variable x in Q_i by the *expression*. This is, indeed, a fully obvious transformation of the formula.

Here, the framework of René Thomas does not only tell us that Q_{i+1} is obtained from Q_i by substituting each occurrence of the variable x in Q_i by $x+1$ (resp. $x-1$). It also tells us that the parameter $K_{x,\omega}$, where ω is the set of resources of x at this timepoint of the experiment, is greater (resp. lower) than the current value of x. So, Q_{i+1} additionally contains the atom $K_{x,\omega} > x$ (resp. $K_{x,\omega} < x$). Moreover, it also contains the formula

$$\bigwedge_{m \in G^{-1}(x)} (m \in \omega \Longleftrightarrow \varphi_m)$$

that formalizes the fact that the set of resources of x is the set of multiplexes m that are the predecessors of x in the interaction graph G whose formula φ_m is satisfied[8].

For example, from the postcondition Q and the last assignment "$operon-$" of the program p it comes: $Q_1 \equiv \left\{ \begin{array}{l} operon - 1 = 0 \\ MucB = 1 \\ K_{operon,\omega_1} < operon \\ alg \in \omega_1 \Longleftrightarrow operon \geq 2 \\ free \in \omega_1 \Longleftrightarrow \neg MucB \geq 1 \end{array} \right.$ which is equivalent to $\left\{ \begin{array}{l} operon = 1 \\ MucB = 1 \\ \omega_1 = \emptyset \\ K_{operon} = 0 \end{array} \right.$.

Similarly, crossing upside the instruction $MucB+$, it comes $Q_2 \equiv \left\{ \begin{array}{l} operon = 1 \\ MucB + 1 = 1 \\ K_{operon} = 0 \\ K_{MucB,\omega_2} > MucB \\ prod \in \omega_2 \Longleftrightarrow operon \geq 1 \end{array} \right.$

which simplifies as $\left\{ \begin{array}{l} operon = 1 \\ MucB = 0 \\ K_{operon} = 0 \\ \omega_2 = \{prod\} \\ K_{MucB,prod} = 1 \end{array} \right.$.

Let us notice that $\left\{ \begin{array}{l} operon = 1 \\ MucB = 0 \end{array} \right.$ are only the initial conditions of the experiment and that the ω_i are only intermediate variables. So, *in fine*, the information from the wet experiment is entirely contained in the formula $(K_{operon} = 0 \wedge K_{MucB,prod} = 1)$.

[8]For sake of simplicity we assume here that, implicitely, all properties on finite sets are known. A fully formal version of our "genetically modified Hoare logic" without such simplifying hypotheses will be published soon.

Simple assignments are also allowed for gene networks and they carry a great advantage because they reflect an external action on a gene. For example the KO of a gene x in the middle of an experiment is simply expressed by the assignment $x = 0$. As we see, unlike the CTL approach, the Hoare approach can jump easily within the state graph of M_{bio} and it can overstep the transitions if required by the experimental protocol.

Hoare logic and Dijkstra weakest precondition algorithm also treat conditional statements and loop statements, with the following rules.

- Let us assume that Q_1 is the weakest precondition to ensure Q after the program p_1, and that Q_2 is the weakest precondition to ensure Q after the program p_2. The weakest precondition to ensure Q after "`if` b `then` p_1 `else` p_2" is $Q_3 \equiv (b \wedge Q_1) \vee (\neg b \wedge Q_2)$.
- Let b be the condition of a `while` statement and let us assume that Q_I is a formula (usually called a *loop invariant*) such that:
  - under the precondition $(Q_I \wedge b)$, the property Q_I is ensured after the program p (in other words, $(Q_I \wedge b \Longrightarrow Q_p)$ where Q_p is the weakest precondition obtained from the postcondition Q_I *via* the program p)
  - and $Q_I \wedge \neg b \Longrightarrow Q$.

  Then Q_I is a sufficient condition to ensured Q after the program "`while` b `do` p".

There would be a lot to say about the fact that Q_I is not necessarily the *weakest* precondition and that finding the good formula Q_I is undecidable, but it is out of the scope of this chapter.

For gene networks, conditional and loop statements are also useful when one wants to factorize several experiments into a single program. The corresponding Hoare/Dijkstra rules stay unchanged.

We also add *quantifiers*. The program "$\exists(p_1, \cdots, p_n)$" means that there exists at least one of the programs p_i for which we observe Q at the end. The program "$\forall(p_1, \cdots, p_n)$" means that whatever the programs p_i, we observe Q at the end. Not surprisingly, if the formulae Q_i are respectively the weakest preconditions for Q *via* the programs p_i, then $(\bigvee_{i=1..n} Q_i)$ [resp. $(\bigwedge_{i=1..n} Q_i)$] is the weakest precondition for Q *via* $\exists(p_1, \cdots, p_n)$ [resp. $\forall(p_1, \cdots, p_n)$].

The quantifiers are useful, for instance, if only a subset of the genes can be observed during an experiment. Assume for example that $MucB$ cannot be observed, only *operon* can be observed. If during an experiment we successively observe an increment of the *operon* followed by a decrement and finally $operon = 0$. Considering that the positive regulation of *operon* on itself cannot be responsible of the oscillation of *operon*, it implies that some non-observable genes have changed in between. Here, it must be $MucB$, so we write the program $(operon+; \exists(MucB+, MucB-); operon-)$ with the postcondition $Q \equiv (operon = 0)$. Similarly to what we did for the previous program example, the weakest precondition procedure, on this example, ends with the formula $\left(\begin{array}{l} (K_{MucB,prod} = 1 \wedge K_{operon} = 0 \wedge K_{operon,free} > 0) \\ \vee \\ (K_{MucB,prod} = 0 \wedge K_{operon,free} = 0 \wedge K_{operon} > 0) \end{array} \right)$. The Snoussi condition contradicts the second alternative (within which $K_{operon,free} < K_{operon}$) and, consequently, the information of the wet experiment with partial observation capabilities is entirely contained in the formula $(K_{MucB,prod} = 1 \wedge K_{operon} = 0 \wedge K_{operon,free} > 0)$. Many other examples could be given, in particular some of them mix together loops and existential quantifiers when several genes are non-observable.

To conclude this rather technical subsection:

- Let us first underline that formal logic can bridge the gap between the results of wet experiments and the mathematical models. Each wet experiment ultimately produces a formula that characterizes the information extracted from it.
- The accumulation of such formulae can be seen as an identification procedure (our examples mostly result in constraints on the parameters). In this regard, let us remind that the only goal is to validate the hypotheses H. Consequently, it is often the case that the identification stay incomplete, provided that all remaining possible parameter values satisfy the hypotheses. It is also worth mentioning that several different models can have identical observational behaviours, taking into account the partial observation capabilities in experimental biology.

6.2 Experimental strategy

The previous subsection establishes that it is a credible idea to consider that a wet experiment is characterised, at the level of mathematical models, by a formula. Consequently, establishing an experimental strategy is mostly equivalent to define a strategy of choices for successive formulae, each formula being choosen among "observable" ones. This remark allows us to *define a framework* for the feedback from modelling to experiments that is independent of the underlying general logic.

Let us assume that the signature of the models is known, as in Section 3. Therefore, the set of all possible formulae about the biological question is $For(\Sigma)$. Remind that $For(\Sigma)$ is much larger than the set of biologically sensible formulae with respect to the biological object (for instance if the underlying logic admits the negation then both φ and $\neg\varphi$ belong to $For(\Sigma)$ for any property φ) but $For(\Sigma)$ somehow delimits the reasonning space (see Figure 13).

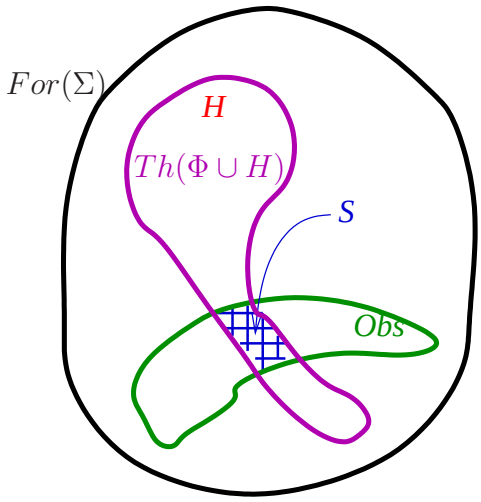
Let us remind that, at this step of the multidisciplinary research process, we know that the biological hypotheses H that motivate the biological research is consistent (Section 5). So, $Mod(\Sigma, \Phi \cup H)$ is non-empty and our goal is to validate/refute that $M_{bio} \in Mod(\Sigma, \Phi \cup H)$. The hypotheses H constitute a formula (possibly a conjunction of formulae) of $For(\Sigma)$ and we have printed H in red in Figure 13, as an element of the set $For(\Sigma)$ in black.

Next, a formula φ is said *observable* if the experimental capabilities permit to perform a wet experiment that directly decides whether $M_{bio} \models \varphi$ without any ambiguity. When discussing with biologists, characterising exactly the set of observable formulae is not obvious, nevertheless, pragmatically, it appears easy to define an approximation of this set, more precisely it appears easy to properly define a credible under-approximation that does not miss the essential of the possible wet experiments. So, one can define a set *Obs* of *observable* formulae, represented in green in Figure 13. The following property is required: for each formula φ in *Obs* there exists a wet experiment which decides without any ambiguity, and without using the deduction rules of the underlying logic, if the biological object satisfies φ (i.e., if $M_{bio} \models \varphi$).

In practice, *Obs* is a compromise between a careful description of the experimental capabilities and the simplicity to formally describe *Obs*. For example, if the general logic is limited to CTL formulae, then a reasonnable description could be:

> Let V_{op} be the subset of V containing the genes whose expression level can be tuned at the beginning of the experiments (V_{op} is classically called the set of *operable* vari-

Figure 13: Selection of wet experiments



Within the set $For(\Sigma)$ of all considered formulae, $Th(\Phi \cup H)$ is the set of all the consequences of the biological hypothesis H, assuming the biological knowledge Φ; and Obs is the set of *observable* formulae i.e. formulae that can be decided without any ambiguity by a single wet experiment. The intersection $S = Th(\Phi \cup H) \cap Obs$ represents the set of experiments that must be considered in order to validate the hypothesis H.

ables). Let V_{obs} be the subset of V containing the variables whose expression level can be measured at the end of the experiments (V_{obs} is classically called the set of observable variables). Then, Obs is the set of formulae of the form "$PRE \Longrightarrow EF(POST)$" where PRE is a conjunction of atoms using variables of V_{op} and $POST$ is a conjunction of atoms using variables of V_{obs}.

In this example, the simplicity to define Obs has been largely favoured, so that Obs is a strong under-approximation of experimental capabilities. For instance, one may want to consider that observable variables can be observed at any time during the experiments. Then, one can use our genetically modified Hoare logic in order to describe the observations by "programs" as in the previous subsection. The set Obs would become the set of formulae of the form "$PRE \Longrightarrow POST$" where $POST$ can contain conjunctions, disjunctions and atoms which involve some parameters $K_{v,\dots}$, so that it would contain the possible weakest preconditions extracted from the algorithm described in the previous subsection. Many other *ad hoc* formal definitions of Obs can be imagined according to the needs of the biological question.

Sometimes, the multiplexes themeselves can play a role to define Obs. For example, back to Figures 2 and 6, this model and the epigenetic switch hypothesis have been experimentally validated in [BCRG04] with limited experimental means: the multiplex *alg* was the only operable part of the model (a saturation of the algU protein in the cell) and the mucus production ("mucus" in Figure 2) was the only observable variable of the system. However, for biological reasons it was also possible to "observe" the modalities AF and AG (due to a huge number of bacteria and the ability to observe the behaviour along several generations of bacteria).

Back to Figure 13, H does not belong to Obs, otherwise it would be trivial to validate or refute H and the modelling activity would be useless. In order to link the hypotheses H and wet experiments, one needs to consider the consequences of H that can be checked experimentally. If φ is such a consequence, then there is a wet experiment that decides if $M_{bio} \models \varphi$. If $M_{bio} \not\models \varphi$ then the hypothesis H is refuted. Obviously, the biological knowledge being supposedly validated, it can be used in order to select φ. So, φ must belong both to Obs and to the set $Th(\Phi \cup H)$ defined as follows (magenta in Figure 13):

$$Th(\Phi \cup H) = \{\ \varphi \in For(\Sigma) \mid \forall M \in Mod(\Sigma)\ \ (M \models \Phi \cup H) \Longrightarrow (M \models \varphi)\ \}$$

If the underlying logic is complete, then $Th(\Phi \cup H) = \{\varphi \in For(\Sigma) \mid \Phi \cup H \vdash \varphi\}$ and theorem proving algorithms can be used.

Let $S = Th(\Phi \cup H) \cap Obs$ be the blue subset of Figure 13. S represents the set of possible experiments able to refute H. Depending to the biological question, some heuristics can be used in order to drive the theorem proving algorithms. Following Popper [Pop63], one should try to select first the properties $\varphi \in S$ that maximize the chance to refute H and there is a major condition: the refutability. *Refutability* means that, if H is not satisfied by the biological object, then there must exist at least one experiment $\varphi \in S$ such that $M_{bio} \not\models \varphi$. By contraposition, and because we do not know which model is M_{bio} in $Mod(\Sigma, \Phi)$, refutability is equivalent to:

$$\forall M \in Mod(\Sigma, \Phi),\ (M \models S) \Longrightarrow (M \models H)$$

If the underlying logic is complete and if it admits the classical deduction theorem, the refutability is equivalent to $(\Phi, S \vdash H)$. If Obs can be described under a suitable form, so that S can be expressed under a finite description, then this alternative form may be easier to manipulate.

The refutability is in fact the first property to study and, pragmatically, if we encounter major difficulties to establish this property:

- either it means that the biological hypothesis under consideration is likely to be too ambitious with respect to the experimental capabilities,
- or it means that the level of description of the mathematical model is too detailed, so that the observable experiments are unable to sufficiently identify the parameters.

In both cases, serious further discussions between modellers and biologists are required.

Back to the example of mucus production of *Pseudomonas aeruginosa* (Figures 2 and 6), we have been somehow lucky in spite of the limited observability: we have proved in [BCRG04] that a unique experiment in S was sufficient to imply the epigenetic switch hypothesis H, namely "$alg \Longrightarrow AGAF(mucus = 1)$". So, refutability was ensured.

Once refutability has been established, there are rare cases where a finite number of experiments (i.e., a finite subset of S) is sufficient to imply H. In general, the definition of efficient heuristics in order to enumerate a (possibly infinite) suite of successive experiments $\varphi_i \in S$ is still an active research subject. Notice that the biological knowledge Φ evolves after each experiment: if the experiment φ_i is in success, i.e. $M_{bio} \models \varphi_i$, then one has to consider $\Phi_{i+1} = \Phi_i \cup \{\varphi_i\}$ for the next selection of experiment. This implies another notion of refutability, which we can call the completeness of the heuristics: the limit knowledge ($\bigcup_{i=1\cdots\infty} \Phi_i$) must imply H.

Things are obviously easier when $Mod(\Sigma, \Phi)$ is finite, as it is the case for the discrete models of gene networks described in Section 4. Moreover, model checking being a very efficient way to check properties at the semantic level ($\models$ instead of $\vdash$), the refutability and the completeness of heuristics are less difficult to study (but that's another story in the general case).

7 Conclusion

This chapter illustrates the usefulness of formal methods and formal logic in the process of establishing a mathematical model for a dynamic biological system, and using it to suggest new experiments. More precisely, we have firstly shown that there is a convenient match between the general logics [Mes89] and the main concepts that research in biology must manipulate in order to understand dynamical systems:

- The notion of signature conveniently carries the *static* knowledge about the system (e.g. an inventory of the possible interactions between constitutive elements).
- The notion of formulae can conveniently formalize the *dynamic* properties of the system under study (the known properties as well as the hypothetical properties).
- The notion of logical models and the satisfaction relation conveniently manipulate *sets* of possible models. Manipulating sets of models instead of a single model is crucial when accompanying a research process in biology where one manages incomplete and evolutive knowledge.

Secondly, according to this matching, two obviously necessary phases clarify the research process, in the context of the Popper definition of experimental sciences [Pop63], namely:

- the *consistency* of the biological hypotheses that motivates the research in biology
- and the *validity* of these hypotheses, which is checked by biological experiments chosen according to their ability to refute the hypotheses.

The entailment relation of general logics allows one to partly mechanize these two steps, by proving that the set of models satisfying both knowledge and hypotheses is not empty, and by offering a way to produce *observable* consequences of the hypotheses (provided that a formal description of possible experiments is made).

In this chapter, we illustrated the approach through the example of gene networks, using the discrete modelling approach proposed by René Thomas in the 70's [Tho78]. We have shown that temporal logics are particularly suited in order to formalize the Thomas' framework [BCRG04]. Moreover, because the models are transition systems, a properly modified Hoare logic [Hoa69, Dij75] makes the bridge between actual biological experiments and these models.

Numerous success stories on several biological questions have in fact allowed one to offer the global view of this chapter. The global approach is very general and it can be applied on many other formal frameworks than the discrete gene network models. We have also used hybrid approaches in order to capture chronological time and delays [ARB+08, CFBR10], extended Petri Nets [TCB09] (using HFPN [MTA+03]), and many others, depending on the biological question under study.

Acknowledgements: the general formal approach outlined in this chapter is the result of a long maturation of ideas. Many colleagues have worked with us and they allowed us to move towards a truly useful methodology of modelling for biology. *René Thomas* is of course the first colleague we would like to thank. His ideas have been at the heart of our discipline. *Janine Guespin-Michel* was the first biologist who took a large amount of her time to explain the main lines of biological reasoning. At the end of the 90's we were pure computer scientists with little idea of how one should properly treat biological problems, and most of our current knowledge is due to the patience of Janine. Similarly, *François Képès* had a key contribution. He has made us understand that what is important in our research field is not necessarily to learn biology in details, but rather to properly understand what researchers in biology consider as an important result. *Adrien Richard* has been the first PhD (in the early 2000's) who inherited of the beginning of this adventure. Since that time, his exceptional ability both to obtain difficult mathematical results and to have a precise understanding of the biological case studies is a major help for us. More recently (almost ten years nevertheless) *Olivier Roux* and *Morgan Magnin* are the ones who have opened our mind to hybrid approaches, to which most of the facts established in this chapter can be extended in order to consider chronometric time instead of the only (chrono)logical succession of events.

References

[ARB+08] J. Ahmad, O. Roux, G. Bernot, J.-P. Comet, A. Richard: *Analysing formal models of genetic regulatory networks with delays: Applications to Lambda phage and T-cell Activation Systems,* Intl J. of Bioinformatics Research and Applications (IJBRA). Inderscience pub., ISSN (Paper) 1744-5485, Vol.4, Num.3, p.240-262, 2008.

[Bar97] J. Barrow-Green: *Poincaré and the Three Body Problem,* American Mathematical Society pub., History of mathematics, Vol.2, ISBN 9780821803677, 1997.

[BCRG04] G. Bernot, J-P. Comet, A. Richard, J. Guespin: *Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic,* J. of Theoretical Biology (JTB), Vol.229, Issue 3, p.339-347, 2004.

[BGM91] G. Bernot, M.C. Gaudel, B. Marre: *Software testing based on formal specifications: A theory and a tool,* Software Engineering Journal (SEJ), Vol.6, num.6, p.387-405, 1991. 1990.

[Bou06] N. Bourbaki: *Théorie des ensembles,* Éléments de mathématique, Vol.1, Springer, 2006.

[BT09] G. Bernot, F. Tahi: *Behaviour Preservation of a Biological Regulatory Network when Embedded into a Larger Network,* Fundamenta Informaticae, IOS Press Amsterdam, Vol.91, Issue.3-4, p.463-485, 2009.

[CCC+04] K.C. Chen, L. Calzone, A. Csikasz-Nagy, F.R. Cross, B. Novak, J.J. Tyson: *Integrative analysis of cell cycle control in budding yeast,* Molecular Biology of the Cell, Vol.15 No.8, p.3841-3862 , 2004.

[CCG+02] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, A. Tacchella: *Nusmv 2: An opensource tool for symbolic model checking,* Proc of Computer Aided Verification (CAV), p.359-364, 2002.

[CE81] E.M. Clarke, E.A. Emerson: *Design and syntheses of synchronization skeletons using branching time temporal logic,* Proc. of Logics of Programs Workshop, Yorktown Heights, Springer LNCS, New York, No.131, p.52-71, 1981.

[CFBR10] J.-P. Comet, J. Fromentin, G. Bernot, O. Roux: *A formal model for gene regulatory networks with time delays,* Proc of Intl Conf on Computational Systems-Biology and Bioinformatics (CSBio'2010), Springer CCIS, Vol.115, p.1-13, 2010.

[CTF+09] F. Corblin, S. Tripodi, E. Fanchon, D. Ropers, L. Trilling: *A declarative constraint-based method for analysing discrete genetic regulatory networks,* Biosystems, Vol.98, p.91-104, 2009.

[Dij75] E.W. Dijkstra: *Guarded commands, nondeterminacy and formal derivation of programs,* Communications of ACM, New York USA, Vol.18 Issue.8, p.453-457, 1975.

[EH82] E.A. Emerson, J.Y. Halpern: *Decision procedures and expressiveness in the temporal logic of branching time,* Proc. of Fourteenth Annual ACM Symposium on Theory of Computing, San Francisco, California, p.169-180, 1982.

[EIP01] J.S. Edwards, R.U. Ibarra, B.O. Palsson: *In silico predictions of Escherichia coli metabolic capabilities are consistent with experimental data,* Nature Biotechnology, Nature Publishing Group, Vol.19 No.2, p.125-130, 2001.

[FCT+04] E. Fanchon, F. Corblin, L. Trilling, B. Hermant, D. Gulino: *Modeling the Molecular Network Controlling Adhesion Between Human Endothelial Cells: Inference and Simulation Using Constraint Logic Programming,* Proc of CMSB 2004, Springer, p.104-118, 2005.

[FH07] J. Fisher, T.A. Henzinger: *Executable cell biology,* Nature Biotechnology, Nature Publishing Group, Vol.25 No.11, p.1239-1249, 2007.

[FMB+06] D. Filopon, A. Merieau, G. Bernot, J.-P. Comet, R. Leberre, B. Guery, B. Polack, J. Guespin: *Epigenetic acquisition of inducibility of type III cytotoxicity in P. aeruginosa,* BMC Bioinformatics, Vol.7, p.272-282, 2006.

[God31] K. Gödel: *On formally undecidable propositions of Principia Mathematica and related systems I,* Oxford Univ. Press, Solomon Feferman ed., 1986. Kurt Gödel Collected works, Vol.I, p.144-195, 1931.

[Hoa69] C.A.R. Hoare: *An axiomatic basis for computer programming,* Communications of the ACM, Vol.12 No.10, p.576-585, 1969.

[HR00] M. Huth, M. Ryan: *Logic in Computer Science: Modelling and reasoning about systems,* Cambridge University Press, 2000.

[KCRB09] Z. Khalis, J.-P. Comet, A. Richard, G. Bernot: *The SMBioNet Method for Discovering Models of Gene Regulatory Networks,* Genes Genomes and Genomics, A. Mansour Ed., Global Science Books, Vol.3, Special Issue 1, p.15-22, 2009.

[Kep13] F. Kepes: *Scientific and technological conditions of the emergence of synthetic biology,* Medecine sciences: M/S, Vol.29, P.13-15, 2013.

[Kha10] Z. Khalis: *Logique de Hoare et identification formelle des paramètres d'un réseau génétique,* PhD thesis, University of Evry-Val d'Essonne, 2010.

[LBK+08] H. Lodish, A. Berk, C.A. Kaiser, M. Krieger, M.P. Scott, A. Bretscher, H. Ploegh, P. Matsudaira: *Molecular Cell Biology,* Freeman, New York, 2008.

[MAC+11] M. Mabrouki, M. Aiguier, J.-P. Comet, P. Le Gall, A. Richard: *Embedding of biological regulatory networks and properties preservation,* Mathematics in Computer Science, Vol.5, No.3, p.263-288, 2011.

[Mes89] J. Meseguer: *General logics,* SRI Intl Menlo Park CA, 1989.

[MTA+03] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, S. Miyano: *Biopathways representation and simulation on hybrid functional Petri net,* In silico biology, IOS Press, Vol.3, No.3, p.389-404, 2003.

[Nou12] M. Noual: *Updating Automata Networks (Mises à jour de réseaux d'automates),* PhD thesis, Ecole Normale Supérieure de Lyon-ENS LYON, 2012.

[NRT+09] A. Naldi, E. Remy, D. Thieffry, C. Chaouiya: *A reduction of logical regulatory graphs preserving essential dynamical properties,* Proc of Computational Methods in Systems Biology (CMSB), Springer, p.266-280, 2009.

[Pnu77] A. Pnueli: *The Temporal Logic of Programs,* Proc. of 18th IEEE Symposium Foundations of Computer Science (FOCS), p.46-57, 1977.

[Pop63] K.R. Popper: *Conjectures and Refutations: The Growth of Scientific Knowledge,* Classics Series 2002, Routledge pub., ISBN 9780415285940, 1963.

[PW09] P.EM Purnick, R. Weiss: *The second wave of synthetic biology: from modules to systems,* Nature Reviews Molecular Cell Biology, Nature Publishing Group, Vol.10 No.6, p.410-422, 2009.

[QS83] Queille J.-P., J. Sifakis: *Fairness and Related Properties in Transition Systems - A Temporal Logic to Deal with Fairness,* Acta Inf., Vol.19, p.195-220, 1983.

[RC07] A. Richard, J.-P. Comet: *Necessary conditions for multistationarity in discrete dynamical systems,* Discrete Applied Mathematics, Vol.155, No.18, p.2403-2413, 2007.

[RH09] C. Rovida, T. Hartung: *Re-evaluation of animal numbers and costs for in vivo tests to accomplish REACH legislation requirements for chemicals-a report by the transatlantic think tank for toxicology (t(4)),* ALTEX J., Vol.26 No.3, p.187-208, 2009.

[Ric10] A. Richard: *Negative circuits and sustained oscillations in asynchronous automata networks,* Advances in Applied Mathematics, Vol.44, No.4, p.378-392, 2010.

[Sif82] J. Sifakis: *A Unified Approach for Studying the Properties of Transition Systems,* Theor. Comput. Sci. (TCS), Vol.18, p.227-258, 1982.

[Sno89] E.H. Snoussi: *Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach,* Dynamics and Stability of Systems, Vol.4, No.3-4, p.565-583, 1989.

[TCB09] S. Troncale, J.-P. Comet, G. Bernot: *Enzymatic Competition: Modeling and Verification with Timed Hybrid Petri Nets,* Pattern Recognition, Elsevier, Vol.42, Num.4, p.562-566, 2009.

[TdA90] R. Thomas, R. d'Ari: *Biological Feedback,* CRC Press book, 1990.

[Tho78] R. Thomas: *Logical analysis of systems comprising feedback loops,* J. of Theoretical Biology (JTB), Vol.73, Issue 4, p.631–656, 1978.

[Tho81] R. Thomas: *On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations,* Springer Series in Synergies, Vol.9, p.180-193, 1981.

[TTK95] R. Thomas, D. Thieffry, M. Kaufman: *Dynamical behaviour of biological regulatory networks - I,* Bull. Math. Biol., Vol.57, No.2, p.247-276, 1995.