Journal of Bioinformatics and Computational Biology

Symbolic modeling of genetic regulatory networks

DANIEL MATEUS and JEAN-PIERRE GALLOIS

Commissariat à l'Énergie Atomique, Saclay
91191 Gif sur Yvette Cedex, France
{*daniel.mateus, jean-pierre.gallois*}*@cea.fr*

JEAN-PAUL COMET and PASCALE LE GALL

Informatique, Biologie Intégrative et Systèmes Complexes
FRE CNRS 2873, Université d'Évry Val d'Essonne and
523 terrasses de l'Agora, 91000 Evry, France
{*jp.comet, pascale.legall*}*@ibisc.univ-evry.fr*

Received (15 October 2006)
Revised (10 January 2007)
Accepted (7 February 2007)

Understanding the functioning of genetic regulatory networks supposes a modeling of biological processes in order to simulate behaviors and to reason on the model. Unfortunately, the modeling task is confronted to incomplete knowledge about the system. To deal with this problem we propose a methodology that uses the qualitative approach developed by R. Thomas. A symbolic transition system can represent the set of all possible models in a concise and symbolic way. We introduce a new method based on model-checking techniques and symbolic execution to extract constraints on parameters leading to dynamics coherent with known behaviors. Our method allows us to efficiently respond to two kinds of questions: is there any model coherent with a certain hypothetic behavior? Are there behaviors common to all selected models? The first question is illustrated with the example of the mucus production in *Pseudomonas aeruginosa* while the second one is illustrated with the example of immunity control in bacteriophage lambda.

Keywords: Gene networks; qualitative dynamical models; symbolic execution; temporal properties; model-checking.

1. Introduction

Modeling and simulation are often necessary to understand genetic regulatory networks as the complexity of the interleaved interactions between constituents of the network (mainly genes and proteins) makes intuitive reasoning too difficult.[1] The typical difficulty in the modeling approach is the lack of precise knowledge about the system, with very few reliable quantitative data. To overpass this bottleneck, qualitative models have been developed (for example in Refs. 2, 3, 4), whose goal consists in abstracting details of the system although preserving qualitative observations. For example, in the multivalued discrete approach developed by R. Thomas and co-workers,[3] the concentrations of the constituents are represented by integer

variables which can only take a finite number of values. It has been shown that such a discrete model can be seen as a qualitative abstraction of a system of piecewise-linear differential equations.[5] This formalism has been used to model various gene networks (for example in Refs. 6, 7, 8, 9, 10).

Nevertheless, even in such a discrete and finite formalism there are usually more than one model compatible with the knowledge on the system. Knowledge generally consists, on the one hand, in inhibitions or activations between genes and other constituents of the network, and on the other hand, in behaviors, observed in experiments. Inhibitions or activations allow one to constrain the possible values of the parameters of the model, on which the evolution depends. It is more difficult to select the parameters corresponding to observed behaviors. The property relating homeostasis (stable cyclic behavior) or multi-stationarity to the steadiness of characteristic states of feedback circuits[11] can be used to decrease the number of parameter values to be considered, as in the GINsim tool.[12]

To go further, two main ideas have been proposed. The first one consists in using constraint logic programming, to manipulate partially known models.[13] As this approach does not allow one to describe all observed behaviors, the difficulty of selecting parameters according to observations remains. The other one uses a temporal logic (computational tree logic or CTL) to specify the behaviors. Verification of behavior specification is then studied for each complete model (i.e. where each parameter has a precise value) independently. Thus the tool SMBioNet[14] selects the models with respect to a given specified behavior after having exhaustively generated all possible models. In the tool GNA,[15] CTL is also used to specify behaviors but only one complete model can be simulated.

In this paper we propose a method combining the advantages of both approaches. The set of possible models can be represented by a unique formal model, a symbolic transition system (STS).[16] Symbolic execution techniques allow the simulation of the STS, generating the possible behaviors. We specify behaviors using linear temporal logic (LTL),[17] and we select parameters with respect to LTL formulas by building constraints: parameters satisfying these constraints define the set of all models verifying the specified LTL behavior. This work has been implemented in the Agatha tool, which is also used for validation purposes of industrial specifications.[18,19]

In Sec. 2, after having described the discrete modeling, we introduce constraints deduced from gene interactions and show their use in the system associated to mucus production in *Pseudomonas aeruginosa*. This system will be used as a running example to explain our method. Section 3 is divided in three parts. We firstly explain the translation of a set of models into a STS model, we then introduce symbolic execution techniques. We secondly explain how behaviors can be specified with LTL formulas, and the way we extend usual model-checking techniques to characterize parameters coherent with the LTL formulas. We thirdly show how this framework can be fruitfully applied to discover the unknowns (parameters or behaviors) of the genetic regulatory network. We address more specifically the two following questions: is there any model coherent with a certain hypothetic behavior?

Are there behaviors common to all possible models? Section 4 demonstrates the whole methodology on the example of immunity control in bacteriophage lambda.

2. Discrete modeling of genetic regulatory networks

In this section we first present the notion of discrete descriptions, also called complete or basic models, which model the different possible behaviors. We then show how biological knowledge, in particular the well known gene interaction graph, can be used to construct a set of acceptable discrete descriptions.

2.1. *Discrete description*

After having proposed an asynchronous boolean modeling,[20] R. Thomas generalized his approach in a multivalued discrete modeling.[3,21,22]

In this approach the genetic regulatory network is described by n variables, each representing the concentration of a constituent of the actual network, mainly the proteins produced by the genes of the network. Each variable x_i can take an integer value between 0 and a maximum value b_i. A *state* $E = (E_1, \ldots, E_n)$ is a vector of values of the variables. With each state E, and each variable x_i, is associated a parameter $K(x_i, E)$, which has an integer value between 0 and b_i. This parameter is the value toward which the associated variable tends in the associated state. It means that in the state E:

- If $K(x_i, E) > E_i$, then $(E_1, \ldots, E_i + 1, \ldots, E_n)$ is a successor of E;
- If $K(x_i, E) < E_i$, then $(E_1, \ldots, E_i - 1, \ldots, E_n)$ is a successor of E;
- If $K(x_i, E) = E_i$ for all i, then E is called a steady state, and has only itself as successor.

The associated *transition graph* is constituted of the states, and the transitions between each state and its successors. This complete model, for which each parameter has been instantiated, is called in the sequel a *discrete description*.

Even if the exact values of the parameters can not be measured in vivo, equalities and inequalities between parameters can be deduced when positive or negative interactions between genes are known, as shown in Sec. 2.2.

Example 1. We consider a system with two variables x and y, corresponding to two proteins. If $b_1 = 2$ and $b_2 = 1$ then x can take values 0, 1 or 2 and y can take values 0 or 1. If $K(x, (0,0)) = 1$ and $K(y, (0,0)) = 1$ then the state $(0,0)$ has two successors, $(1,0)$ and $(0,1)$. It means that if in the system the concentrations of the two proteins are at the lowest level, the concentrations increase to reach a state corresponding to $(1,0)$ or $(0,1)$. This example will be detailed in Sec. 2.3.

2.2. *Constraints deduced from interactions*

Interactions between constituents of a network imply constraints on the parameters. Typically a product x_i can have a positive or a negative effect on a product x_j: if

the concentration of x_i is under a certain threshold, the rate of synthesis of x_j is not affected, but if the concentration of x_i is beyond the threshold, the rate of synthesis of x_j increases in the case of a positive interaction, or decreases in the case of a negative interaction. Then in the discrete description, there is an integer value θ called the *discrete threshold* of the interaction of x_i on x_j: when the integer value associated with x_i is lower than θ, there is no effect, and when the integer value is greater than or equal to θ, then the interaction is effective. It can be translated into the following constraints: let $E = (E_1, \ldots, E_n)$ be a state where $E_i < \theta$, and let $E' = (E'_1, \ldots, E'_n)$ be a state with $E'_i \geq \theta$ and for $k \neq i$, $E_k = E'_k$, that is E' differs from E only in its i^{th} coordinate, then

- $K(x_j, E) \leq K(x_j, E')$ in the case of a positive interaction;
- $K(x_j, E) \geq K(x_j, E')$ in the case of a negative interaction.

Moreover, if $E = (E_1, \ldots, E_n)$ and $E' = (E'_1, \ldots, E'_n)$ are states where $E_i < \theta$, $E'_i < \theta$ and $E_k = E'_k$ if $k \neq i$, then $K(x_j, E) = K(x_j, E')$; similarly, if $E_i \geq \theta$ and $E'_i \geq \theta$ then $K(x_j, E) = K(x_j, E')$.

These equalities allow to introduce a notation of the parameters: if X is a subset of $\{x_1, \ldots, x_n\}$ whose elements can have an interaction on x_j, then $K(x_j, X)$ is the value of all parameters $K(x_j, E)$ where E is a state such that the value of the elements of X are beyond the threshold of the interaction on x_j, and the variables not in X have a value under the threshold (or do not have an interaction on x_j).

If a product x_i has interactions with two different products x_j and x_k, the thresholds of the two interactions are usually different. So, in the discrete description, x_i can take at least three values, as x_i can be under the two thresholds, between the two thresholds, or beyond the two thresholds.

Sometimes more precise knowledge about the interactions is available. For example the presence of two different products x *and* y can be necessary to activate a gene z, or x can activate z but the simultaneous presence of x and y produces an inhibition. These two facts are respectively translated into constraints: $K(z, \{x\}) = K(z, \{y\}) = K(z, \emptyset)$ and $K(z, \{x, y\}) \geq K(z, \emptyset)$ in the first case, or $K(z, \{x, y\}) \leq K(z, \emptyset) \leq K(z, \{x\})$ in the second case.

2.3. *Mucus production in* Pseudomonas aeruginosa

Pseudomonas aeruginosa are bacteria that secrete mucus (alginate) in lungs affected by cystic fibrosis, but not in common environment. As this mucus increases respiratory deficiency, this phenomenon is a major cause of mortality. Details of the regulatory network associated with the mucus production are described by Govan and Deretic.[23] The simplified regulatory network, as proposed by Guespin and Kaufman,[24] contains the protein AlgU (product of algU gene), and an inhibitor complex anti-AlgU (product of muc genes). AlgU has a positive effect on anti-AlgU and on itself, while anti-AlgU has a negative effect on AlgU. A sufficient concentration of AlgU leads to the production of mucus (by activating different alg genes).

The discrete description of the associated network contains two variables x and y, respectively corresponding to AlgU and anti-AlgU. x can take the values 0, 1, 2, and y can take the values 0, 1. We assume that if $x \geq 1$ then x has a positive effect on y, if $x = 2$ then x has a positive effect on itself. If $y = 1$ then y has a negative effect on x. Moreover the production of mucus is possible only if the concentration of x is greater than or equal to 2 (see Fig.1).

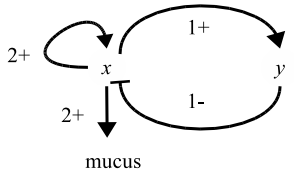

Fig. 1. Graph of interactions associated with the mucus production. Each arrow indicates an interaction from a regulator to a regulated variable; the sign indicates a positive or negative effect, and the integer is the threshold of the interaction.

Some constraints can be deduced from this graph of interactions as described in Sec. 2.2. For example $K(x, \emptyset) = K(x, (0, 0)) = K(x, (1, 0))$ (the value of x, 0 or 1, is under the threshold of the interaction on itself, which is 2, and the value of y, 0, is under the threshold of the interaction on x, which is 1). The inequalities between parameters are $K(x, \{y\}) \leq K(x, \emptyset) \leq K(x, \{x\})$ and $K(x, \{y\}) \leq K(x, \{x, y\}) \leq K(x, \{x\})$ (because y has a negative interaction on x, and x has a positive interaction on itself), and similarly $K(y, \emptyset) \leq K(y, \{x\})$ (x has a positive interaction on y). Moreover we assume that $K(x, \{y\}) = 0$ and $K(y, \emptyset) = 0$. This additional constraint means that the basal level of x and y is 0, as $K(x, \{y\})$ and $K(y, \emptyset)$ are less than or equal to the other parameters. The set of all constraints deduced from the graph of interactions will be denoted by $\mathcal{C}$ in the sequel.

It has been observed that mucoid *P. aeruginosa* can continue to produce mucus isolated from infected lungs. It is commonly thought that the mucoid state of *P. aeruginosa* is due to a mutation which cancels the inhibition of *algU* gene. An alternative hypothesis has been made: this mucoid state can occur in reason of an epigenetic modification, i.e. without mutation.[24] The models compatible with this hypothesis are constructed in Refs. 25, 14. We use the same example to explain our methodology in Sec. 3.

2.4. *Manipulating sets of discrete descriptions*

We have just shown that the only knowledge of the graph of interactions is not sufficient to precisely determine which is the true behavior of the biological system: numerous discrete descriptions can fit the constraints deduced from the interaction

graph. In the example of Fig. 1, there $3^4 \times 2^2 = 324$ discrete descriptions since[a] parameters $K(x, \emptyset)$, $K(x, \{x\})$, $K(x, \{y\})$ and $K(x, \{x, y\})$ can take three different values (0, 1 or 2), and parameters $K(y, \emptyset)$ and $K(y, \{x\})$ can take two different values (0 or 1).

In order to precise the behavior of the biological system, complementary biological knowledge, different from previously used interaction graphs, have to be taken into consideration. Each new biological information allows us to reduce the set of acceptable discrete descriptions. In the example of Fig. 1 if we take for granted that y tends towards its basal level (i.e. 0) when x does not activate it, then we can deduce that $K(y, \emptyset) = 0$; then there remain 162 acceptable discrete descriptions.

The aim of this paper is to propose a method for manipulating such sets of discrete descriptions. The two main technical contributions are:

- expressing biological knowledge by temporal logic formulas involving equalities and inequalities on gene expression levels;
- denoting sets of acceptable discrete descriptions by mean of a symbolic model provided with constraints on parameters.

Next section focuses on these both aspects.

3. Symbolic modeling and analysis

3.1. *Symbolic transition systems and symbolic execution*

A symbolic transition system (STS)[16] is a transition system whose transitions are labeled by conditions on STS variables and assignments of STS variables. Each initialization of STS variables yields a basic model where each variable has a precise initial value, and all transitions are defined according to the STS transitions. Thus, a STS, denoted by M, is parameterized by an initialization function. We can associate to M the set of all basic models obtained by applying an initialization function: $\{\sigma(M) \mid \sigma \text{ initialization function}\}$ denotes this set.

For specifying genetic regulatory networks, the STS variables correspond to the set of parameters $K(x_i, E)$ of the associated discrete descriptions and the set of STS states is exactly the set of states of the associated discrete descriptions. The transitions are labeled according to the rules defined in Sec. 2.1. Nevertheless we need to take into account additional knowledge corresponding to constraints deduced from interactions. These constraints can naturally be expressed as first order formulas over the set of parameters. In this article we call symbolic model any couple $(M, \mathcal{C})$, where M is the STS with parameters $K(x_i, E)$ as STS variables and $\mathcal{C}$ a set of constraints over parameters $K(x_i, E)$. It defines a set of basic models $\{\sigma(M) \mid \sigma \text{ initialization function} \wedge \forall C \in \mathcal{C}, \sigma \models C\}$. Each basic model $\sigma(M)$ is precisely a discrete description associated to the values of parameters defined by σ and $\sigma \models C$ means that the parameters instantiated by σ satisfy the constraint C.

[a]Let us recall that a discrete description is completely defined by the values of parameters.

Symbolic execution has been introduced for analysis purposes of computer programs.[26] The method has been extended to STSs, and is used in the Agatha tool for behavioral analysis[27] and conformance testing.[28] As the known constraints and rules of evolution of a discrete description can easily been specified in a STS, we have adapted symbolic execution techniques to generate all behaviors compatible with the constraints on the parameters. The method constructs a tree whose vertices are states labeled by constraints, with the following rules:

- The root of the tree is a state, associated with the initial constraints $\mathcal{C}$.
- Let us suppose that E is an already constructed state of the tree, labeled by the constraints $\mathcal{C}_E$, and that there is a STS transition from E to E' labeled by the condition D. The state E' provided with the constraint $\mathcal{C}_E \cup \{D\}$ is built iff the conjunction of the constraints of $\mathcal{C}_E \cup \{D\}$ is satisfiable. A new transition is built from $(E, \mathcal{C}_E)$ to $(E', \mathcal{C}_{E'})$, where $\mathcal{C}_{E'} = \mathcal{C}_E \cup \{D\}$.
- The process is repeated until the new state has already been encountered in the tree path from the root to the new state.

Let us point out that every state in the tree is associated with constraints whose conjunction is called *path condition*; this path condition is the condition on parameters under which the path exists.

Figure 2 shows the possible paths in the case of the mucus production system in *P. aeruginosa*, with $(x, y) = (0, 0)$ as initial state, and $\mathcal{C}$ as initial constraints as described in Sec. 2.3. For simplicity reason, the constraints labeling vertices are not represented in the figure.

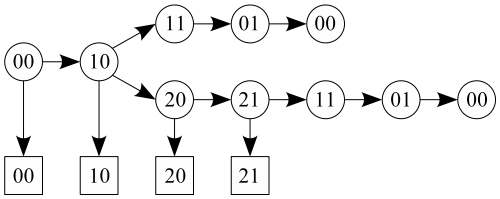

Fig. 2. Symbolic execution from the initial state $(0, 0)$. Squares indicate steady states. Each state of the figure is associated with constraints: for example $(1, 0)$ is a successor of $(0, 0)$ if $K(x, \emptyset) > 0$. So the set of constraints associated with $(1, 0)$ is $\mathcal{C} \cup \{K(x, \emptyset) > 0\}$; $(0, 0)$ is a steady state if $(K(x, \emptyset) = 0 \wedge K(y, \emptyset) = 0)$. But $(0, 1)$ is not a successor of $(0, 0)$ because in this case $K(y, \emptyset) > 0$, which is not compatible with the initial constraint $K(y, \emptyset) = 0$.

3.2. *Specification of paths and synthesis of constraints*

3.2.1. *Linear temporal logic*

To search a specific path in the symbolic execution tree we adapt model-checking techniques for linear temporal logic (LTL).[17] Intuitively model-checking techniques

consist in exploring all states of a basic model to state whether this model satisfies or not a given temporal logic formula.[29]

A LTL formula expresses properties of a path. This logic adds to the classical operators of propositional logic[b] mainly two temporal operators, called Next (N), and Until (U). If f and g are formulas, Nf means that f is true in the following state of the path, and fUg means that f is true in each state of the path, until g becomes true (and g eventually happens). We can then define the operators Finally (F) and Globally (G); Ff means that f eventually happens (and can be written $\top Uf$); Gf means that f is always true (and can be written $\neg F(\neg f)$).

We extend classical LTL model-checking techniques designed for basic models to STSs. Just as classical LTL model-checking only considers pertinent paths according to the formula, our method also considers pertinent paths according to the formula, but in our case each state of a path is provided with constraints on parameters. The key point is that a path is eliminated as soon as the conjunction of constraints is no more satisfiable. This leads to a minimal tree construction and gives us the solutions in term of constraints: the disjunction of the path conditions associated to all remaining paths. The resulting constraint represents all parameter valuations compatible with the behavior specified by the formula. To summarize, given a symbolic model $(M, \mathcal{C})$, extended LTL model-checking allows us to compute all initialization functions (i.e. parameter valuations) leading to basic models satisfying a LTL formula. In other words, the extended LTL model-checking associates to any LTL formula a characteristic constraint defining the discrete descriptions satisfying it.

Let us remark that the developed technique constructs the disjunction of constraints on possible paths. Then satisfying a LTL formula for a model means that there exists at least a path satisfying the LTL formula. Such a property is qualified as existential. On the contrary we may want to select models whose all paths satisfy the formula (universal property). In such a case the negation of the universal property is unsatisfiable. We have then to specify this impossible behavior as a LTL formula. It suffices to take the negation of the associated constraint to find all models compatible with the universal property.

3.2.2. *Adding experimental knowledge to the symbolic model*

When considering behaviors, expressed as LTL formulas, supposed to be known to occur in the actual system, we can add the corresponding characteristic constraints $\mathcal{D}$ to the symbolic model $(M, \mathcal{C})$. We get the symbolic model $(M, \mathcal{C} \cup \mathcal{D})$ restricting the set of discrete descriptions.

For example, *P. aeruginosa* do not produce mucus in a common environment, so there is no path from a state where $x = 0$ to a state where $x = 2$. That is clearly an universal property. In order to show that it is not possible to reach $x = 2$ from

[b]As $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\top$ (true), $\bot$ (false).

$x = 0$, we consider the formula $(x = 0) \wedge F(x = 2)$. The associated constraint, generated by our method, is $K(x, \emptyset) > 1$. The negation is simply $K(x, \emptyset) \leq 1$. All discrete descriptions verifying the latter constraint satisfy the universal property.

3.3. *Extracting knowledge from the symbolic model*

Let us come back to the two central questions asked in the Introduction: is there any model coherent with a certain hypothetic behavior? Are there behaviors common to all possible models?

The first question consists in specifying the hypothesis with LTL formulas, and finding the associated constraints. When the constraints are not satisfiable, there is no model compatible with the LTL formulas. When they are satisfiable, the solutions of the constraints give all parameter valuations, each one corresponding to a discrete description satisfying the LTL formulas (see example 2).

The second question consists in finding properties common to all discrete descriptions associated to a symbolic model $(M, \mathcal{C})$. The set of constraints $\mathcal{C}$ precisely represents such common properties (see Sec. 4 for an illustration).

Example 2. If the hypothesis of an epigenetic change in mucoid *P. aeruginosa* is verified, bacteria which produce mucus can continue to produce mucus in a common environment. A path beginning with $x = 2$ which turns back forever to $x = 2$ is described by $((x = 2) \wedge G(F(x = 2)))$. The resulting constraint can be written $((K(x, \{x, y\}) = 2 \wedge K(y, \{x\}) = 1) \vee (K(x, \{x\}) = 2 \wedge K(y, \{x\}) = 0))$. This constraint implies that the (mucoid) state $(2, 1)$ is a steady state, or that $(2, 0)$ is a steady state. There are 8 discrete descriptions verifying the constraints; in these models the mucoid state can be related to an epigenetic modification. These constraints imply the existence of a stable mucoid state, but not that all paths from a mucoid state come back to a mucoid state. This more restrictive behavior, is achieved if $K(x, \{x, y\}) > 1$, i.e. for 4 models from the 8.

4. Application to immunity control in bacteriophage lambda

4.1. *Immunity control in bacteriophage lambda and associated STS*

Bacteriophage lambda is a virus whose DNA can integrate into bacterial chromosome and be faithfully transmitted to the bacterial progeny. After infection, most of the bacteria display a lytic response and liberate new phages, but some display a lysogenic response, i.e. survive and carry lambda genome, becoming immune to infection. Figure 3 is the graph of interactions described by Thieffry and Thomas[6] which has also been studied in Ref. 30. Four genes are involved, called cI, cro, cII and N. The states, represented by a vector (cI, cro, cII, N), are in $\{0, 1, 2\} \times \{0, 1, 2, 3\} \times \{0, 1\} \times \{0, 1\}$. Even with the constraints deduced following Sec. 2.2, the associated symbolic model represents 1 052 000 different discrete descriptions.

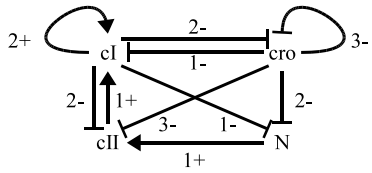

Fig. 3. Graph of interactions associated with immunity control in bacteriophage lambda. Arrows are labeled by the threshold and sign of the corresponding interaction.

4.2. *Lytic and lysogenic pathways of lambda-phage*

4.2.1. *Specification of known behaviors*

The lytic response leads to the states (0,2,0,0) or (0,3,0,0) where cro is fully expressed. The lysogenic response leads to the state (2,0,0,0), where cI is fully expressed, and the repressor produced by cI blocks the expression of the other viral genes, leading to immunity. We define the LTL formulas: $lytic = (\mathrm{cI} = 0 \wedge \mathrm{cro} \geq 2 \wedge \mathrm{cII} = 0 \wedge \mathrm{N} = 0)$ and $lysogenic = (\mathrm{cI} = 2 \wedge \mathrm{cro} = 0 \wedge \mathrm{cII} = 0 \wedge \mathrm{N} = 0)$.

When the system reaches a lytic or lysogenic state it does not leave it, so $lytic \wedge F(\neg lytic)$, and $lysogenic \wedge F(\neg lysogenic)$ specify impossible paths. The viral proteins are initially absent when the viral genome integrates a cell, and lytic and lysogenic responses are possible: this is translated into the two formulas $init \wedge F(lytic)$ and $init \wedge F(lysogenic)$ where $init = (\mathrm{cI} = 0 \wedge \mathrm{cro} = 0 \wedge \mathrm{cII} = 0 \wedge \mathrm{N} = 0)$.

4.2.2. *Deduced constraints*

As $lytic \wedge F(\neg lytic)$ and $lysogenic \wedge F(\neg lysogenic)$ specify impossible paths, the constraints added to the symbolic model are $(K(\mathrm{cI}, \{\mathrm{cro}\}) = 0 \wedge K(\mathrm{cro}, \emptyset) > 1 \wedge K(\mathrm{cII}, \emptyset) = 0 \wedge K(\mathrm{N}, \{\mathrm{cro}\}) = 0)$ and $(K(\mathrm{cI}, \{\mathrm{cI}\}) = 2 \wedge K(\mathrm{cro}, \{\mathrm{cI}\}) = 0 \wedge K(\mathrm{cII}, \{\mathrm{cI}\}) = 0 \wedge K(\mathrm{N}, \{\mathrm{cI}\}) = 0)$. All models verifying the previous constraints verify $init \wedge F(lytic)$. The constraint associated with $init \wedge F(lysogenic)$ is $(K(\mathrm{cI}, \emptyset) = 2) \vee (K(\mathrm{cI}, \{\mathrm{cII}\}) = 2 \wedge K(\mathrm{cII}, \{\mathrm{N}\}) = 1 \wedge K(\mathrm{N}, \emptyset) = 1)$.

4.3. *Questioning the symbolic model*

In all discrete descriptions represented by the symbolic model, $K(\mathrm{cro}, \emptyset) > 1$: in this case there is always the path to lysis $(0000) \rightarrow (0100) \rightarrow (0200)$.

The additional constraint associated with $init \wedge F(lysogenic)$ is $(K(\mathrm{cI}, \emptyset) = 2) \vee (K(\mathrm{cI}, \{\mathrm{cII}\}) = 2 \wedge K(\mathrm{cII}, \{\mathrm{N}\}) = 1 \wedge K(\mathrm{N}, \emptyset) = 1)$. In the models where $K(\mathrm{cI}, \emptyset) = 2$, there is a direct pathway to immunity: $(0000) \rightarrow (1000) \rightarrow (2000)$; in all other models, $(K(\mathrm{cI}, \{\mathrm{cII}\}) = 2 \wedge K(\mathrm{cII}, \{\mathrm{N}\}) = 1 \wedge K(\mathrm{N}, \emptyset) = 1)$ implies the existence of the path: $(0000) \rightarrow (0001) \rightarrow (0011) \rightarrow (1011) \rightarrow (2011) \rightarrow (2010) \rightarrow (2000)$, which is the most likely pathway according to Thieffry and Thomas.[6]

There are 2156 models with the specified behaviors: lytic and lysogenic states are stable, and there is a pathway from initial state to lysis and to lysogeny. In all these models, there is a common path to lysis, and one path from the two described paths to lysogeny.

5. Conclusion

We have shown how a symbolic model representing a set of possible discrete descriptions of a genetic regulatory network permits to deal with incomplete knowledge. Known interactions can be translated in constraints on the parameters, which can be specified in a symbolic transition system. This STS can be simulated with symbolic execution techniques. The known behaviors can be specified with LTL formulas, and then, model-checking techniques have been extended to select the constraints on parameters associated with these behaviors. Adding these contraints to the STS, a symbolic model representing the discrete descriptions coherent with the known behaviors is obtained.

Then we have explained how the symbolic model can be used to reveal new results: the possibility of hypothetic behaviors can be tested (as the epigenetic change in *P. aeruginosa*) or common behaviors between all possible descriptions can be found (as possible pathways to lysis or lysogeny in bacteriophage lambda).

A promising follow-up to the work presented here would be to develop a method to find not only the common behaviors, but also discriminating properties, in order to refine the symbolic model. Such a property would permit one to propose biological experiments, in order to restrict the possible parameter valuations.

References

1. de Jong H. Modeling and simulation of genetic regulatory systems: A literature review. *J Comput Biol*, 9(1):67–103, 2002.
2. Glass L and Kauffman SA. The logical analysis of continuous non-linear biochemical control networks. *J Theor Biol*, 39:103–129, 1973.
3. Thomas R and D'Ari R. *Biological Feedback*. CRC Press, Boca Raton, Florida, 1990.
4. de Jong H, Gouzé JL, Hernandez C, Page M, Sari T, and Geiselmann J. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull Math Biol*, 66(2):301–340, 2004.
5. Snoussi EH. Qualitative dynamics of piecewise-linear differential equations : A discrete mapping approach. *Dyn Stability Syst*, 4:189–207, 1989.
6. Thieffry D and Thomas R. Dynamical behaviour of biological regulatory networks –ii. immunity control in bacteriophage lambda. *Bull Math Biol*, 57(2):277–97, 1995.
7. Mendoza L, Thieffry D, and Alvarez-Buylla ER. Genetic control of flower morphogenesis in arabidopsis thaliana: a logical analysis. *Bioinformatics*, 15(7-8):593–606, 1999.
8. Sanchez L and Thieffry D. Segmenting the fly embryo: a logical analysis of the pair-rule cross-regulatory module. *J Theor Biol*, 224(4):517–37, 2003.
9. Mendoza L. A network model for the control of the differentiation process in th cells. *Biosystems*, 84(2):104–14, 2006.

10. Chaouiya C, Thieffry D, and Sanchez L. From gradients to stripes: a logical analysis of drosophila segmentation genetic network. In N Kolchanov, R Hofestaedt, and L Milanesi, editors, *Bioinformatics of Genome Regulation and Structure II*, pages 379–90. Springer, 2006.
11. Thomas R, Thieffry D, and Kaufman M. Dynamical behaviour of biological regulatory networks –i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull Math Biol*, 57(2):247–76, 1995.
12. Gonzalez AG, Naldi A, Sanchez L, Thieffry D, and Chaouiya C. Ginsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, 84(2):91–100, 2006.
13. Fanchon E, Corblin F, Trilling L, Hermant B, and Gulino D. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In *Proceedings of Computational Methods in Systems Biology (CMSB 2004)*, volume 3082 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2005.
14. Bernot G, Comet JP, Richard A, and Guespin J. Application of formal methods to biological regulatory networks: extending thomas' asynchronous logical approach with temporal logic. *J Theor Biol*, 229(3):339–347, 2004.
15. Batt G, Ropers D, de Jong H, Geiselmann J, Mateescu R, Page M, and Schneider D. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in escherichia coli. *Bioinformatics*, 21(suppl. 1):i19–i28, 2005.
16. Aiguier M, Gaston C, Le Gall P, Longuet D, and Touil A. A temporal logic for input output symbolic transition systems. In *Proceedings of the 12th Asia-Pacific Software Engineering Conference*, pages 43–50. IEEE Computer Society Press, 2005.
17. Vardi MY. An automata-theoretic approach to linear temporal logic. In *Proceedings of the Banff Higher order workshop conference on Logics for concurrency : structure versus automata*, pages 238–266, Secaucus, NJ, USA, 1996. Springer-Verlag New York.
18. Bigot C, Faivre A, Gallois JP, Lapitre A, Lugato D, Pierron JY, and Rapin N. Automatic test generation with agatha. In *Proceedings of Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2003)*, volume 2619 of *Lecture Notes in Computer Science*, pages 591–596. Springer, 2003.
19. Lugato D, Bigot C, Valot Y, Gallois JP, Gérard S, and Terrier F. Validation and automatic test generation on uml models: the agatha approach. *International Journal on Software Tools for Technology Transfer (STTT)*, 5(2-3):124–139, 2004.
20. Thomas R. Boolean formalization of genetic control circuits. *J Theor Biol*, 42(3):563–585, 1973.
21. Thomas R. Regulatory networks seen as asynchronous automata : a logical description. *J Theor Biol*, 153(1):1–23, 1991.
22. Thomas R and Kaufman M. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180–95, 2001.
23. Govan JR and Deretic V. Microbial pathogenesis in cystic fibrosis: mucoid pseudomonas aeruginosa and burkholderia cepacia. *Microbiol rev*, 60(3):539–74, 1996.
24. Guespin-Michel J and Kaufman M. Positive feedback circuits and adaptive regulations in bacteria. *Acta Biotheor*, 49(4):207–218, 2001.
25. Guespin-Michel J, Bernot G, Comet JP, Mérieau A, Richard A, Hulen C, and Polack B. Epigenesis and dynamic similarity in two regulatory networks in pseudomonas aeruginosa. *Acta Biotheor*, 52(4):379–390, 2004.
26. King JC. Symbolic execution and program testing. *Commun ACM*, 19(7):385–394,

1976.
27. Rapin N, Gaston C, Lapitre A, and Gallois JP. Behavioral unfolding of formal specifications based on communicating extended automata. In *Proceedings of the first international workshop on Automated Technology for Verification and Analysis (ATVA'03)*, National Taipei University, Taiwan, 2003.
28. Gaston C, Le Gall P, Rapin N, and Touil A. Symbolic execution techniques for test purpose definition. In *Proceedings of Testing of Communicating Systems (TestCom 2006)*, volume 3964 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2006.
29. Clarke EM, Grumberg O, and Peled D. *Model Checking*. MIT Press, Cambridge, Mass., 2000.
30. Richard A, Comet JP, and Bernot G. Formal methods for modeling biological regulatory networks. In AH Gabbar, editor, *Modern Formal Methods and Applications*, pages 83–122. Springer, 2006.

Daniel Mateus received his master degree in mathematical logic from University Paris 7 Denis Diderot, France, in 2003. He is currently a graduate student at the CEA (the French Atomic Energy Agency) in Saclay, France, on secondment from the French ministry of education. His research interests include modeling biological processes by using formal methods of computer science.

Jean-Paul Comet is Assistant Professor in Computer Sciences at the University of Evry (France). He is member of the FRE 2873 IBISC laboratory and the Epigenomics Project of Genopole Evry (France). His research interests are bioinformatics including sequence analysis, expression data analysis and modeling of complex biological systems.

Jean-Pierre Gallois is an engineer graduated from SUPELEC (the French Superior School of Electricity). He has been working at the CEA for 13 years. He leads the AGATHA research project that focuses on methods and tools for the validation and the testing of concurrent and real time systems. He has started an activity in bioinformatics which topic is to adapt formal methods for biology in collaboration with the IBISC laboratory.

Pascale Le Gall is Professor in Computer sciences at the University of Evry (France). She is member of the FRE 2873 IBISC laboratory and of the Epigenomics Project of Genopole Evry (France). Her research interests concern formal methods for software engineering and their applications. She is interested in geometric modeling, telecommunication services, conformance testing, systems biology.