Analysing Gene Regulatory Networks by both Constraint Programming and Model-Checking

Jonathan Fromentin[1], Jean-Paul Comet[2], Pascale Le Gall[2] & Olivier Roux[1]

[1] IRCCyN UMR 6597, CNRS & École Centrale de Nantes
1, rue de la Noë - BP 92 101 - 44321 Nantes CEDEX 03
{jonathan.fromentin,olivier.roux}@irccyn.ec-nantes.fr
[2] IBISC UMR 8042, CNRS & Université d'Évry
Boulevard François Mitterrand - 91025 Évry CEDEX
{legall,jp.comet}@ibisc.univ-evry.fr

Abstract— **In this article, we propose a formal method to analyse gene regulatory networks (GRN). The dynamics of such systems is often described by an ordinary differential equation system, but has also been abstracted into a discrete transition system. This modeling depends on parameters for which different values are possible. Each instantiation of these parameters defines a possible dynamics and verification tools can be used to select the tuples of values which lead to dynamics consistent with known behaviours. GRN are so complex that their discrete modeling gives a number of possible dynamics exponential in function of the GRN's size (number of genes and interactions). In this paper, we propose to use constraint programming and CTL formal language to determine the set of all dynamics consistent with the known behavioral properties without enumerating all of them. This approach allows us therefore to minimize the computation time necessary for the research of these parameters.**

I. Introduction

The Gene Regulatory Networks (GRN) and their formal models [1] were designed to describe the interactions between genes inside the cell. They attempt to foresee the complex dynamics of relative concentrations of several interacting proteins. Numerous examples of GRN [2], [3] have been modeled by using a discrete modeling approach. It has been shown that computer tools can be used to analyse their structure and functional properties.

In [3], authors show that the modeling of R. Thomas [4] may be completed with a model-checking method which allows one to verify if a model satisfies a temporal property expressing particular biological knowledge. In the modeling of R. Thomas, the dynamics of the system is built from the parameters of the model. To construct the set of all dynamics consistent with the known biological properties, a first approach [3] consists in enumerating all possible valuations of the parameters of the model and in selecting those that satisfy biological properties transcripted into formal logical formulae.

Using constraint programming [5], [6], [7] allows one to bypass the previous enumeration step, by generating constraints on the parameters that oblige the dynamics of the model to respect the specified properties. Therefore, constraint programming seems to be a promising approach to minimize the computation time necessary to the research of the parameters of the model that lead to a dynamics consistent with the biological observations. The goal of this work is to design and implement a combined approach (model-checking and constraint programming) to select the valuations of the parameters of a GRN that lead to a dynamics consistent with the biological specification.

Other works are close to our approach. The work of F. Corblin *et al.* [8], which is based on the piece-wise linear differential equation [9], also uses constraint logic programming for analysis of biological networks but without using temporal logic. The authors conclude on the difficulty to put constraints on any length paths and, more generally, constraints corresponding to property expressed in a particular temporal formal language: CTL. Another approach consists in using symbolic execution [10] and LTL[1] model-checking technics [11] in order to build consistent model parameterizations from a LTL formula and a symbolic model.

The paper is organized as follows. We recall in section II the principles of discrete modeling for genetic regulatory networks. Our specific approach combining model-checking and constraint programming is explained in Section III and its implementation is presented in section IV. We introduce in section V examples on GRN of the mucus production system in bacteria *Pseudomonas aeruginosa* [12], [13]. We finally conclude in section VI.

II. Gene Regulatory Networks

A GRN is a labelled directed graph $G = (V_G, E_G)$ where each vertex $i \in V_G$ is provided with a boundary $\beta_i = \max(1, d_i)$ where d_i is the number of out-going edges of i. Each edge $(i \to j) \in E_G$ is labelled with a couple $(\varepsilon_{ij}, \tau_{ij})$ where $\varepsilon_{ij} \in \{+,-\}$ is the sign of the interaction and τ_{ij}, called threshold, is an integer between 1 and β_i.

A qualitative state of the system is then defined as a vector constituted by qualitative concentration levels.

Definition 1 (Qualitative state): A qualitative state of a GRN is a tuple $n = (n_1, n_2, \ldots, n_p)$, where p is the number of genes and n_i, the qualitative concentration level of gene i, is an integer value between 0 and β_i.

[1]Another temporal formal language.

The resources of a gene g at a given state are defined as the set of genes which have a "positive" influence on g at a particular state.

Definition 2 (Resources): Given a GRN $G=(V_G,E_G)$ and a state $n=(n_1,n_2,\dots,n_p)$, the set of resources of i is $\omega_i(n)=$

$$\left\{ j \in V_G \;\middle|\; \begin{array}{ll} \text{either} & (j \xrightarrow{(+,\tau_{ji})} i) \in E_G \text{ and } (n_j \geq \tau_{ji}) \\ \text{or} & (j \xrightarrow{(-,\tau_{ji})} i) \in E_G \text{ and } (n_j < \tau_{ji}) \end{array} \right\}$$

A model of GRN is a couple of sets: a set of parameters which gives the dynamics of the system and a set of constraints on these parameters which enables one to construct the specification of the system. Intuitively, parameters abstract the direction for the future concentration of each i.

Definition 3 (Model of a GRN): Let $G=(V_G,E_G)$ be a GRN. The model of G is $M(G)=(Var,Const)$ with:

- $K \subseteq Var$ is a set of parameters such that $K_{i,\omega_i(n)} \in K$ denotes the level toward which the abstract concentration of $i \in V_G$ evolves when the resources are $\omega_i(n)$.
- *Const* is a set of constraints on the parameters of *Var*.

The system goes from a state to another one provided that they are linked together by the neighbourhood relation.

Definition 4 (Neighbourhood): Given two states, $n=(n_1,n_2,\dots,n_p)$ and $n'=(n'_1,n'_2,\dots,n'_p)$, n and n' are neighbours (denoted by $n \leftrightarrow n'$) iff $|n-n'|=\sum_{j\in[1,p]}|n_j-n'_j|=1$.

Notation 1 (Transition and path): Let n and n' be two states. If n and n' are neighbours then the passage of the system from the state n to the state n' is denoted $n \to n'$ and we say that there is a transition from n to n'. Therefore, n is called a predecessor of n' and n' a successor of n. The existence of a sequence of transitions going from n to n' is called a path from n to n'.

III. Approach of Constraint Programming

Constraint programming is a programming paradigm where relations among variables are constraints. The constraint programming approach consists in searching the values for the variables in building a search tree and using a backtracking method.

A. Domains and Basic Constraints

The parameters of the model are variables of the problem which are defined in a domain.

Definition 5 (Domain): The domain of a parameter $K_{i,\omega_i(n)}$ is $[0,\beta_i]$ for all states n.

Definition 6 (Transition constraint): Let n and n' be two neighbour states. We introduce $C_{nn'} \in Const$ the transition constraint for the transition $n \to n'$. The rank, constraint $C_{nn'}$ and sign of the transition $n \to n'$ are defined by:

$$\begin{aligned} rank(n \to n') &= i \text{ such that } |n_i - n'_i| = 1; \\ sign(n \to n') &= \begin{cases} + & \text{if } n'_i = n_i + 1; \\ - & \text{otherwise.} \end{cases} \\ C_{nn'} &= \begin{cases} K_{i,\omega_i(n)} > n_i & \text{if } sign(n \to n') = +; \\ K_{i,\omega_i(n)} < n_i & \text{otherwise.} \end{cases} \end{aligned}$$

This definition for $C_{nn'}$ is the translation of the modeling of R. Thomas in terms of constraints on parameters $K_{i,\omega_i(n)}$. The constraints $C_{nn'}$ are built on the neighbourhood relation, but by abuse of notation we say that C_{nn} are the constraints guaranteeing the stability of the state n. The stability constraints C_{nn} are defined by: $\bigwedge_i \left(K_{i,\omega_i(n)} = n_i\right)$.

B. CTL Formal Language

Dynamical properties of a system can often be translated into computational tree logic (CTL) formulae [14], [15]. The set of atomic propositions which depends on the GRN $G=(V_G,E_G)$ is denoted AP. The subset of AP of formulae which are true in a state n, is given by the labeling function $\mathscr{L}(n)=\{v_i=n_i \mid i \in V_G,\ n_i \in [1,\beta_i]\}$ where $v_i=n_i$ means that the gene i has the concentration level n_i. We consider in the sequel the formulae containing only the CTL operators: $\neg$, $\wedge$, EX, EU and AU. It is well known that any CTL formula can be transformed into a semantically equivalent CTL formula containing only these operators.

Definition 7: (semantics of CTL) Let $\{s_i \mid i \in \mathbb{N}\}$ be states. The semantics of CTL is defined inductively by:

- $s_0 \models \top$,
- $\forall p \in AP$, $s_0 \models p$ iff $p \in \mathscr{L}(s_0)$,
- $s_0 \models \neg\phi$ iff $s_0 \not\models \phi$,
- $s_0 \models \phi_1 \wedge \phi_2$ iff $s_0 \models \phi_1$ and $s_0 \models \phi_2$,
- $s_0 \models EX(\phi)$ iff for a particular s_1 such that $s_0 \to s_1$, we have $s_1 \models \phi$,
- $s_0 \models E[\phi_1 U \phi_2]$ iff there exists a particular path $s_0 \to s_1 \to \cdots \to s_j \to \cdots$ such that $s_j \models \phi_2$ and $\forall i<j\ s_i \models \phi_1$.
- $s_0 \models A[\phi_1 U \phi_2]$ iff for all paths $s_0 \to s_1 \to \cdots \to s_j \to \cdots$ there exists a s_j such that $s_j \models \phi_2$ and $\forall i<j$ we have $s_i \models \phi_1$.

C. Intuitive Semantics of Constraints for CTL formula

Translating CTL formulae into constraints is difficult because the operators EU and AU can represent a large number of possible paths. Before giving the semantic of constraints for CTL formula in subsection III-D, we give the intuitive interpretation of constraints for the operators. We introduce the notation C^n_ϕ: we say that the constraint C^n_ϕ is satisfied iff the state n validates the formula ϕ. For the operators EU and AU, we introduce the notion of ϕ-path: we say that a path is a ϕ-path if each state of the path validates ϕ.

- If ϕ is an atomic proposition, then C^n_ϕ is satisfied iff $\phi \in \mathscr{L}(n)$.
- If $\phi=\phi_1 \wedge \phi_2$, C^n_ϕ is equivalent to the conjunction of constraints $C^n_{\phi_1}$ and $C^n_{\phi_2}$.
- If $\phi=\neg\phi_1$ then $C^n_\phi=\neg C^n_{\phi_1}$.
- If ϕ is $EX(\phi_1)$ then a state n validates the formula ϕ iff there exists a neighbour state n' such that $C_{nn'} \wedge C^{n'}_{\phi_1}$ is satisfied.
- If ϕ is $E[\phi_1 U \phi_2]$, for treating the formula at the state n we use the additional boolean variables $B^{n'}_\phi$ for defining the $E[\phi_1 U \phi_2]$-paths. The semantics of variables $B^{n'}_\phi$ is: $B^{n'}_\phi$ is true iff the state n' belongs to a $E[\phi_1 U \phi_2]$-path. A $E[\phi_1 U \phi_2]$-path is a non-cyclic $(\phi_1 \vee \phi_2)$-path such that at least an of its states satisfy ϕ_2. The states of a $E[\phi_1 U \phi_2]$-path can be defined inductively: $B^{n'}_\phi$ is true iff

  - either $C^{n'}_{\phi_2}$ is satisfied,
  - or $C^{n'}_{\phi_1}$ is satisfied and that there exists a neighbour n'' of n' such that $B^{n''}_{\phi}$ is true and $C_{n'n''}$ is satisfied.
- If ϕ is $A[\phi_1 U \phi_2]$, for treating the formula at the state n we use the additional boolean variables $D^{n'}_{\phi}$ for defining the $A[\phi_1 U \phi_2]$-paths. A $A[\phi_1 U \phi_2]$-path is a path where at least one of its states satisfy ϕ_2, and, where all the successors of its states preceding the first state satisfying ϕ_2, belong to a $A[\phi_1 U \phi_2]$-path. The semantics of variables $D^{n'}_{\phi}$ is: $D^{n'}_{\phi}$ is true iff the state n' belongs to a $A[\phi_1 U \phi_2]$-path. The states of a $A[\phi_1 U \phi_2]$-path can be defined inductively: $D^{n'}_{\phi}$ is true iff
  - either $C^{n'}_{\phi_2}$ is satisfied,
  - or $C^{n'}_{\phi_1}$ is satisfied, and there exists at least one neighbour n'' of n' such that $C_{n'n''}$ is satisfied and, for all n'' neighbour of n' such that $C_{n'n''}$ is satisfied we have $D^{n''}_{\phi}$ true.

However, with the inductive definition of $E[\phi_1 U \phi_2]$-path, it is possible to have only the cyclic (ϕ_1)-paths where no state validates ϕ_2 (also called cyclic $(\phi_1 \wedge \neg\phi_2)$-paths). In fact, if the semantics of $E[\phi_1 U \phi_2]$ is respected then there exists at least one path such that at least one state of this path validates ϕ_2. The topological sort of the graph containing only the $(\phi_1 \vee \phi_2)$-path is introduced to translate the satisfiability of $E[\phi_1 U \phi_2]$ into constraints. For this purpose, we introduce the variable $V^{n'}_{\phi}$ for each state n' which has a value between 1 and the number of states. These variables rank the states belonging to $(\phi_1 \vee \phi_2)$-paths. We manage variables together with an order on these variables such that $V^{n'}_{\phi} < V^{n''}_{\phi}$ iff

- the state n' belongs to the $(\phi_1 \vee \phi_2)$-path,
- the state n'' belongs to the $(\phi_1 \vee \phi_2)$-path and
- $C_{n'n''}$ is satisfied.

The topological sort is possible iff the graph containing only the $(\phi_1 \vee \phi_2)$-paths is non-cyclic. Therefore, with this new constraint, it is possible to get rid of the $(\phi_1 \wedge \neg\phi_2)$-paths. This topological sort must also be done for $A[\phi_1 U \phi_2]$ and we use for that the variables $W^{n'}_{\phi}$.

D. Constraints Obliging a State to Satisfy a CTL Formula

Let n be a state. Let $C^n_{\phi} \in Const$ be a constraint associated with ϕ and state n. Let N_n be the set of neighbours: $N_n = \{n' \mid n \leftrightarrow n'\}$. Let q be the number of states of GRN. The constraints $C^n_{\phi} \in Const$ are defined inductively as follows:

- if $\phi = \top$ then $C^n_{\phi} = true$.
- if $\phi \in AP$ then $C^n_{\phi} = true$ iff $\phi \in \mathscr{L}(n)$.
- if $\phi = \neg\phi_1$ then $C^n_{\phi} = \neg C^n_{\phi_1}$.
- if $\phi = \phi_1 \wedge \phi_2$ then $C^n_{\phi} = C^n_{\phi_1} \wedge C^n_{\phi_2}$.
- if $\phi = EX(\phi_1)$ then $C^n_{\phi} = \bigvee_{n' \in N_n \cup \{n\}} \left(C_{nn'} \wedge C^{n'}_{\phi_1}\right)$.
- if $\phi = E[\phi_1 U \phi_2]$, then there exists boolean variables $B^{n'}_{\phi} \in Var$ and variables $V^{n'}_{\phi} \in Var$ defined in the domain $[1, q]$ such that $C^n_{\phi} = B^n_{\phi}$ and the relation between variables is given by the constraints of *Const* defined below:

$$\bigwedge_{n'} \left(B^{n'}_{\phi} \Leftrightarrow C^{n'}_{\phi_2} \vee C^{n'}_{\phi_1} \wedge \bigvee_{n'' \in N_{n'}} \left(B^{n''}_{\phi} \wedge C_{n'n''} \right) \right) \quad (1)$$

$$\bigwedge_{n'} \left(B^{n'}_{\phi} \Rightarrow C^{n'}_{\phi_2} \vee C^{n'}_{\phi_1} \wedge \bigvee_{n'' \in N_{n'}} \left(B^{n''}_{\phi} \wedge C_{n'n''} \wedge V^{n'}_{\phi} < V^{n''}_{\phi} \right) \right) \quad (2)$$

- if $\phi = A[\phi_1 U \phi_2]$, then there exists boolean variables $D^{n'}_{\phi} \in Var$ and variables $W^{n'}_{\phi} \in Var$ defined in the domain $[1, q]$ such that $C^n_{\phi} = D^n_{\phi}$ and the relation between variables is given by the constraints of *Const* defined below:

$$\bigwedge_{n'} \left(D^{n'}_{\phi} \Leftrightarrow C^{n'}_{\phi_2} \vee C^{n'}_{\phi_1} \wedge \bigvee_{n'' \in N_{n'}} \left(C_{n'n''} \right) \wedge \bigwedge_{n'' \in N_{n'}} \left(C_{n'n''} \Rightarrow D^{n''}_{\phi} \right) \right) \quad (3)$$

$$\bigwedge_{n'} \left(D^{n'}_{\phi} \Rightarrow C^{n'}_{\phi_2} \vee C^{n'}_{\phi_1} \wedge \bigvee_{n'' \in N_{n'}} \left(D^{n''}_{\phi} \wedge C_{n'n''} \wedge W^{n'}_{\phi} < W^{n''}_{\phi} \right) \right) \quad (4)$$

IV. Implementation

A first implementation of this approach is made in the language *Java* with the library *JaCoP* (http://jacop.cs.lth.se/) for constraint programming. This software is "SeMoCo-GRN" (Selecting Models by Constraints for GRN). For more efficiency in the resolving of the constraint (3), we use a property linked to the modeling of R. Thomas: there exists a transition $n \rightarrow n$ iff there does not exist any transition $n \rightarrow n'$ such that $n' \neq n$. This property allows simplifying the constraint $\bigvee_{n'' \in N_{n'}} \left(C_{n'n''} \right)$ into $\neg C_{nn}$. It is also possible to simplify the constraints (2) by removing the constraint $C^{n'}_{\phi_1}$ because the constraints (1) oblige the state n' to validate ϕ_1 or ϕ_2 if $B^{n'}_{\phi}$ is true. Thus according to (1), if $B^{n'}_{\phi}$ is true and if n' does not validate ϕ_2, then n' validates ϕ_1. This simplification is also true for the constraint (4).

V. Examples

We present in this section some examples based on two different dynamics (fig. 2 and 3) of the mucus production system in bacteria *Pseudomonas aeruginosa* (fig. 1). *Pseudomonas aeruginosa* are bacteria that secrete mucus (alginate) in lungs of patients affected by cystic fibrosis, but not in common environment. As this mucus increases respiratory deficiency, this phenomenon is a major cause of mortality.

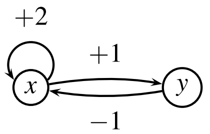

Fig. 1. GRN of mucus production in *Pseudomonas aeruginosa*. The gene x synthesizes a protein which activates the expression of gene y (when its concentration level exceeds the abstract threshold 1) and itself (when it exceeds 2) by binding the promoters. In turn, the protein of y inhibits the expression of x when its concentration level exceeds 1.

It is known that the mucus production occurs when abstract concentration level of x (alginate) is greater or equal to 2. If we add the following properties:

- the state $(2,1)$ ($x=2$ and $y=1$) is a stable state (translated by $(x=2 \wedge y=1) \Rightarrow AG(x=2 \wedge y=1)$).
- a bacteria producing mucus will always produce mucus (translated by $(x=2) \Rightarrow AG(x=2)$).
- a bacteria being in the state $(0,0)$ can produce mucus in the future (translated by $(x=0 \wedge y=0) \Rightarrow E[\top U x=2]$).

then we have a set of six possible dynamics, among which the one in fig 3. On the other hand, the dynamics of the fig. 2 presents two different behaviours (production of mucus and non-production of mucus) since there are two non-connected subgraphs. Thus, fig. 2 presents a dynamics where it is not possible to reach level 2 of x from level 0 without external signal (epigenitic switch) although fig. 3 presents a dynamics where level 2 is reached from all initial states.

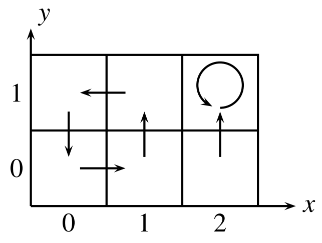

Fig. 2.

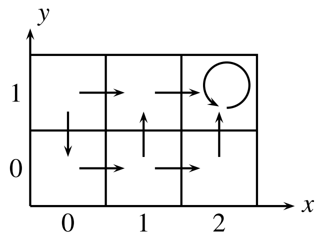

Fig. 3.

From fig. 2 and 3, we show two examples with the property $(x=0 \wedge y=0) \Rightarrow E[\top U x=2]$ allowing better understanding of the use of additional variables. In the examples, ϕ denotes $E[\top U x=2]$.

For the dynamics of fig. 2, we begin with $C_\phi^{(0,0)} = (B_\phi^{(0,0)} = 1)$. As the states $(2,0)$ and $(2,1)$ satisfy the atomic proposition $x=2$, we have by equations (1) and (2): $B_\phi^{(2,0)} = 1$, $B_\phi^{(2,1)} = 1$ and later by these equations and these two results,

- either $B_\phi^{(1,0)} = B_\phi^{(1,1)} = B_\phi^{(0,0)} = B_\phi^{(0,1)} = 0$ leads us to the contradiction $B_\phi^{(0,0)} = 1 \wedge B_\phi^{(0,0)} = 0$.
- or $B_\phi^{(1,0)} = B_\phi^{(1,1)} = B_\phi^{(0,0)} = B_\phi^{(0,1)} = 1$ leads us to the contradictions $V_\phi^{(1,0)} \neq V_\phi^{(1,0)}$, $V_\phi^{(1,1)} \neq V_\phi^{(1,1)}$, $V_\phi^{(0,0)} \neq V_\phi^{(0,0)}$ and $V_\phi^{(0,1)} \neq V_\phi^{(0,1)}$.

This example shows that the constraints corresponding to the formula CTL are not consistent. It is therefore normal that this solution does not appear in the six previously obtained dynamics. We notice that without equation (2) this example would have been consistent with equation (1).

For the dynamics of the fig. 3, we begin also with $C_\phi^{(0,0)} = (B_\phi^{(0,0)} = 1)$ and for the same reasons as previously, we have:

- $B_\phi^{(2,0)} = 1$ and $B_\phi^{(2,1)} = 1$ and arbitrary we put also
- $V_\phi^{(2,0)} = 6$ and $V_\phi^{(2,1)} = 6$.
- Later, by these equations and these results, we have successively:
  1) $B_\phi^{(1,0)} = 1$, $V_\phi^{(1,0)} = 5$, $B_\phi^{(1,1)} = 1$ and $V_\phi^{(1,1)} = 5$.
  2) $B_\phi^{(0,0)} = 1$, $V_\phi^{(0,0)} = 4$, $B_\phi^{(0,1)} = 1$ and $V_\phi^{(0,1)} = 4$.

This example shows that the constraints corresponding to the formula CTL are consistent.

VI. Conclusion

We have designed and implemented a combined approach using CTL and constraint programming, in order to perform model-checking of gene regulatory networks. This approach translates temporal properties of the biological system into constraints on the parameters of the discrete model. Among other things, this allows us to have all consistent dynamics with the temporal properties of the biological system without having to construct the inconsistent dynamics. The advantage of this approach meets again in the practice. For example, for the GRN of immunity control in bacteriophage lambda [16], which is bigger than the one of the mucus production in *Pseudomonas aeruginosa*, there are 32 interesting dynamics among the possible 3 millions . Our prototype SeMoCo-GRN finds these interesting dynamics in 800 milliseconds whereas the enumerating approach [3] finds them in 8 minutes and 30 seconds. Our future work will be to optimize the relations between the variables and to reduce the number of additional variables, in order to be able to treat bigger GRN.

References

[1] H. de Jong, "Modeling and simulation of genetic regulatory systems: a literature review." *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103., 2002.

[2] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page, "Qualitative simulation of the initiation of sporulation in *Bacillus subtilis*," *Bull. of Math. Bio.*, vol. 66, no. 2, pp. 261–299, 2004.

[3] A. Richard, J.-P. Comet, and G. Bernot, *Modern Formal Methods and Applications*. Springer, ISBN: 1-4020-4222-1, 2006, ch. Formal Methods for Modeling Biological Regulatory Networks.

[4] R. Thomas, "Regulatory networks seen as asynchronous automata : A logical description," *J. theor. Biol.*, vol. 153, pp. 1–23, 1991.

[5] F. Fages, *Programmation Logique Par Contraintes*. Ellipses (ISBN 2729846131), 1996.

[6] T. Frühwirth and S. Abdennadher, *Essentials of Constraint Programming*. Springer (ISBN: 3540676236), 2003.

[7] K. R. Apt, *Principles of Constraint Programming*. Cambridge University Press (ISBN 0521825830), 2003.

[8] F. Corblin, E. Fanchon, and L. Trilling, "Inférer et simuler un modèle biologique décrivant l'adhérence entre cellules," in *Premières Journées Francophones de Programmation par Contraintes*, 2005.

[9] H. de Jong, M. Page, C. Hernandez, and J. Geiselmann, "Qualitative simulation of genetic regulatory networks: Method and application," in *IJCAI*, 2001, pp. 67–73.

[10] M. Aiguier, C. Gaston, P. Le Gall, D. Longuet, and A. Touil, "A temporal logic for input output symbolic transition systems," *In Proc. of the 12th Asia-Pacific Software Eng. Conf.*, pp. 43–50, 2005.

[11] D. Mateus, J.-P. Gallois, J.-P. Comet, and P. Le Gall, "Symbolic modeling of genetic regulatory networks," *Journal of Bioinformatics and Computational Biology*, 2007.

[12] J. Guespin-Michel and M. Kaufman, "Positive feedback circuits and adaptive regulations in bacteria." *Acta Biotheor.*, vol. 49, no. 4, pp. 207–18, 2001.

[13] M. McCaw, G. Lykken, P. Singh, and T. Yahr, "Exsd is a negative regulator of the pseudomonas aeruginosa type iii secretion regulon," *Mol. Microbiol.*, vol. 46, no. 4, pp. 1123–33, 2002.

[14] E. Emerson, *Handbook of theoretical computer science, Volume B : formal models and semantics*. MIT Press, 1990, ch. Temporal and modal logic, pp. 995–1072.

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

[16] D. Thieffry and R. Thomas, "Dynamical behaviour of biological regulatory networks - II. immunity control in bacteriophage lambda." *Bull. Math. Biol.*, vol. 57, no. 2, pp. 277–297, 1995.