Analysis of Genetic Regulatory Networks: A Model-Checking Approach

Grégory Batt[1], Hidde de Jong[1], Johannes Geiselmann[2], and Michel Page[1,3]

[1]Institut National de Recherche en Informatique et en Automatique (INRIA),
Unité de recherche Rhône-Alpes, Grenoble, France
[2] Laboratoire Adaptation et Pathogénie des Microorganismes (CNRS FRE 2620),
Université Joseph Fourier, Grenoble, France
[3]Ecole Supérieure des Affaires, Université Pierre Mendès France, Grenoble, France

Contact person: Gregory.Batt@inrialpes.fr

Abstract

Methods developed for the qualitative simulation of dynamical systems have turned out to be powerful tools for studying genetic regulatory networks. A bottleneck in the application of these methods is the analysis of the simulation results. In this paper, we propose a combination of qualitative simulation and model-checking techniques to perform this task systematically and efficiently. By means of the example of the network controlling the initiation of sporulation in *B. subtilis*, we argue that this approach is well-adapted to the kind of questions biologists habitually ask and the kind of data available to answer these questions.

Introduction

Qualitative simulation is concerned with making predictions of the behavior of dynamical systems when only qualitative information is available. In QSIM (Kuipers 1994), probably the best-known approach towards qualitative simulation, the variables of the system take qualitative values expressed in terms of a totally-ordered set of landmark values. The structure of the system is described by means of a qualitative differential equation, an abstraction of a class of ordinary differential equations. A qualitative differential equation consists of constraints on the qualitative value of the variables, corresponding to basic mathematical equations. Qualitative simulation exploits the qualitative constraints and continuity properties of the variables to predict the possible qualitative behaviors of the system. Given an initial qualitative state, consisting of a qualitative value for each of the variables, the simulation algorithm produces a branching tree of all reachable qualitative states.

Qualitative simulation provides a discrete view on the dynamics of a system. A qualitative behavior produced by QSIM consists of a sequence of qualitative states, alternating between time-points and time-intervals. The order of qualitative states in the behavior expresses a temporal order of events at which the qualitative value of some variable, and hence the qualitative state of the system, changes. The abstraction of the continuous behavior of a system into a sequence of qualitative states makes it possible to use *model-checking* techniques for the verification of properties of the system (Clarke, Grumberg, & Peled 1999). The application of these techniques has been proposed as a means to deal with one of the major problems of QSIM and other classical qualitative simulation methods: the analysis of the large number of possible sequences of qualitative states predicted (Brajnik & Clancy 1998; Shults & Kuipers 1997).

The aim of this paper is to explore the combined use of qualitative simulation and model checking techniques in the context of a biological application, the analysis of *genetic regulatory networks*. These networks of regulatory interactions between genes, proteins, metabolites, and other small molecules underlie the development and functioning of all living organisms. Mathematical methods supported by computer tools are indispensable for the analysis of genetic regulatory networks, since most networks of interest involve many genes connected through interlocking positive and negative feedback loops, thus making an intuitive understanding of their dynamics difficult to obtain (de Jong 2002). Currently, only a few networks are well-understood on the molecular level, and quantitative information on the interactions is seldom available. This has stimulated an interest in qualitative approaches towards the analysis of genetic regulatory networks.

In previous work we have developed a method for the qualitative simulation of genetic regulatory networks (de Jong *et al.* 2002a; 2002b; 2001). The method differs from traditional approaches towards qualitative simulation in that it has been tailored to a class of piecewise-linear (PL) differential equations with favorable mathematical properties (Glass & Kauffman 1973; Mestl, Plahte, & Omholt 1995; Thomas & d'Ari 1990). This allows it to deal with large and complex networks of regulatory interactions. The qualitative simulation method has been implemented in a publicly-available computer tool, called Genetic Network Analyzer (GNA) (de Jong *et al.* 2003). The program has been used to analyze several genetic regulatory networks of biological interest, including the network controlling the initiation of sporulation in *B. subtilis*.

In this paper, we will show how the graph of qualitative behaviors produced by the simulation method can be reformulated as a Kripke structure. Moreover, we will illustrate how observed properties of the behavior of the genetic regulatory network can be expressed in the temporal logic CTL

(Clarke & Emerson 1981). This allows existing, highly-efficient model-checking techniques (Clarke, Grumberg, & Peled 1999; Cimatti *et al.* 2002) to be used to validate the model of the network, that is, to check whether a statement in temporal logic representing an observed property is satisfied by the Kripke structure obtained from the model through simulation. We will argue by means of the example of the sporulation network that the chosen combination of qualitative simulation and model checking is well-adapted to the kind of questions biologists habitually ask as well as the kind of data available to answer these questions.

In the next two sections of this paper, we briefly review the qualitative modeling and simulation of genetic regulatory networks. This will set the stage for a discussion of the combined use of qualitative simulation and model-checking techniques in the third section. The applicability of this approach to the validation of actual genetic regulatory networks is the subject of the next section. We finish with a discussion of the approach in the context of related work.

Qualitative modeling of genetic regulatory networks

The dynamics of genetic regulatory networks can be modeled by a class of piecewise-linear (PL) differential equations of the following general form (Glass & Kauffman 1973; Mestl, Plahte, & Omholt 1995; Thomas & d'Ari 1990):

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{x},\ \boldsymbol{x} \geq \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)'$ is a vector of cellular protein concentrations, and $\boldsymbol{f} = (f_1, \ldots, f_n)'$, $\boldsymbol{g} = \mathrm{diag}(g_1, \ldots, g_n)$. The rate of change of each concentration x_i, $1 \leq i \leq n$, is defined as the difference of the rate of synthesis $f_i(\boldsymbol{x})$ and the rate of degradation $g_i(\boldsymbol{x})\,x_i$ of the protein.

The function $f_i : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is defined as

$$f_i(\boldsymbol{x}) = \sum_{l \in L} \kappa_{il}\, b_{il}(\boldsymbol{x}), \tag{2}$$

where $\kappa_{il} > 0$ is a rate parameter, $b_{il} : \mathbb{R}^n_{\geq 0} \to \{0, 1\}$ a *regulation function*, and L a possibly empty set of indices of regulation functions. A regulation function b_{il} is the arithmetic equivalent of a Boolean function expressing the logic of gene regulation (Mestl, Plahte, & Omholt 1995; Thomas & d'Ari 1990). The function g_i expresses the regulation of protein degradation. It is defined analogously to f_i, except that we demand that $g_i(\boldsymbol{x})$ is strictly positive. In addition, in order to formally distinguish degradation rates from synthesis rates, we will denote the former by γ instead of κ.

Figure 1 gives an example of a simple genetic regulatory network. Genes *a* and *b*, transcribed from separate promoters, encode proteins A and B, each of which controls the expression of both genes. More specifically, proteins A and B repress gene *a* as well as gene *b* at different concentrations.

The network in figure 1 can be described by means of the following pair of state equations:

$$\dot{x}_a = \kappa_a\, s^-(x_a, \theta^2_a)\, s^-(x_b, \theta^1_b) - \gamma_a\, x_a \tag{3}$$

$$\dot{x}_b = \kappa_b\, s^-(x_a, \theta^1_a)\, s^-(x_b, \theta^2_b) - \gamma_b\, x_b. \tag{4}$$

Gene *a* is expressed at a rate $\kappa_a > 0$, if the concentration of protein A is below its threshold θ^2_a and the concentration of protein B below its threshold θ^1_b, that is, if $s^-(x_a, \theta^2_a)\, s^-(x_b, \theta^1_b) = 1$. Recall that $s^-(x, \theta)$ is a step function evaluating to 1, if $x < \theta$, and to 0, if $x > \theta$. Protein A is spontaneously degraded at a rate proportional to its own concentration ($\gamma_a > 0$ is a rate constant). The state equation of gene *b* is interpreted analogously.

Qualitative simulation of genetic regulatory networks

The dynamical properties of the PL models (1) can be analyzed in the n-dimensional phase space box $\Omega = \Omega_1 \times \ldots \times \Omega_n$, where every Ω_i, $1 \leq i \leq n$, is defined as $\Omega_i = \{x_i \in \mathbb{R}_{\geq 0} \mid 0 \leq x_i \leq max_i\}$. max_i is a parameter denoting a maximum concentration for the protein. Given that the protein encoded by gene i has p_i threshold concentrations, the $n-1$-dimensional threshold hyperplanes $x_i = \theta^{k_i}_i$, $1 \leq k_i \leq p_i$, partition Ω into (hyper)rectangular regions that are called *domains* (de Jong *et al.* 2002a). More precisely, a domain $D \subseteq \Omega$ is defined by $D = D_1 \times \ldots \times D_n$, where every D_i, $1 \leq i \leq n$, is defined by one of the equations below:

$$\begin{aligned}
D_i &= \{x_i \mid 0 \leq x_i < \theta^1_i\},\\
D_i &= \{x_i \mid x_i = \theta^1_i\},\\
D_i &= \{x_i \mid \theta^1_i < x_i < \theta^2_i\},\\
&\ldots\\
D_i &= \{x_i \mid \theta^{p_i}_i < x_i \leq max_i\}.
\end{aligned}$$

Figure 2(a) shows the subdivision into domains of the two-dimensional phase space box of the example network. We distinguish between domains like D^4 and D^7, which are located on (intersections of) threshold planes, and domains like D^1, which are not. The former domains are called *switching* domains and the latter *regulatory* domains.

When evaluating the step function expressions of (1) in a regulatory domain, f_i and g_i reduce to sums of rate constants. More precisely, in a regulatory domain D, f_i reduces to some μ^D_i, and g_i to some ν^D_i. It can be shown that all solution trajectories in D monotonically tend towards a stable equilibrium $\Phi(D) = \{(\mu^D_1/\nu^D_1, \ldots, \mu^D_n/\nu^D_n)\}$, the *target equilibrium* (Glass & Kauffman 1973; Mestl, Plahte, & Omholt 1995; Thomas & d'Ari 1990). The target equilibrium level μ^D_i/ν^D_i of the protein concentration x_i gives an indication of the strength of gene expression in D. If $\Phi(D) \cap D \neq \{\}$, then all trajectories will remain in D. If not, they will leave D at some point. In regulatory domain D^1 in figure 2(b), the trajectories tend towards $\Phi(D^1) = \{(\kappa_a/\gamma_a, \kappa_b/\gamma_b)\}$. Since $\Phi(D^1) \cap D^1 = \{\}$, the trajectories starting in D will leave this domain at some point. Different regulatory domains generally have different target equilibria. For instance, in regulatory domain D^3, the target equilibrium is given by $\{(0, \kappa_b/\gamma_b)\}$ (not shown).

In switching domains, f_i and g_i may not be defined, because some concentrations assume their threshold value. Moreover, f_i and g_i may be discontinuous in switching domains. In order to cope with this problem, the system of

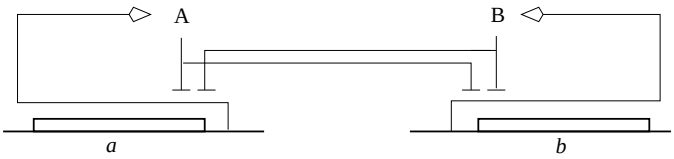


Figure 1: Example of a genetic regulatory network of two genes (*a* and *b*), each coding for a regulatory protein (A and B) (see figure 4 for the legend).

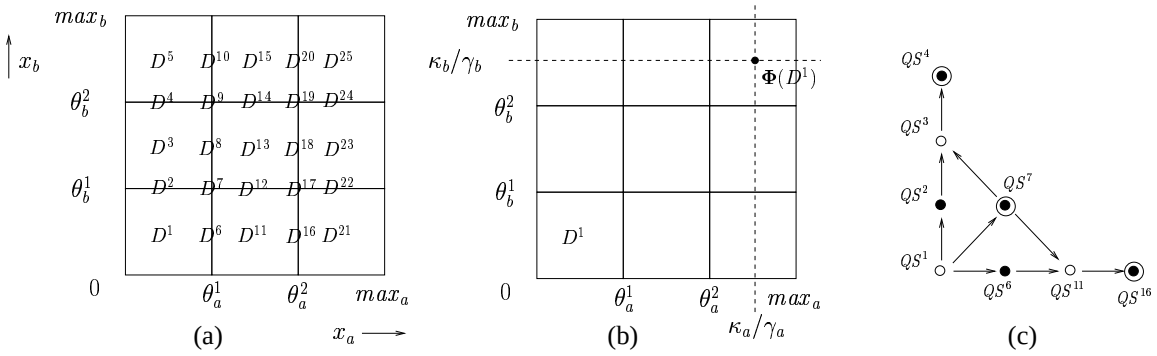


Figure 2: Qualitative simulation of the regulatory network in figure 1. (a) Subdivision of the phase space into regulatory and switching domains. (b) Analysis of the model in regulatory domain D^1, using the parameter inequalities (5)-(6). (c) Transition graph resulting from a simulation of the example system starting in the domain D^1. Qualitative states associated with regulatory domains and switching domains are indicated by unfilled and filled dots, respectively. Qualitative states associated with domains containing an equilibrium point are circled (de Jong *et al.* 2002a).

differential equations (1) is extended into a system of differential inclusions, following an approach widely used in control theory (Gouzé & Sari 2003). Using this generalization, it can be shown that, in the case of a switching domain D, the trajectories either traverse D instantaneously or remain in D for some time, tending towards a target equilibrium set $\Phi(D)$. Here, $\Phi(D)$ is the smallest closed convex set including the target equilibria of regulatory domains having D in their boundary, intersected with the hyperplane containing D (see (de Jong *et al.* 2002a) for technical details). If $\Phi(D) \cap D \neq \{\}$, then the trajectories may remain in D. If not, they will leave D at some point.

Most of the time, precise numerical values for the threshold and rate parameters in (1) are not available. However, the above summary of the properties of PL models reveals that a qualitative understanding of the dynamics of a regulatory system can be obtained by knowing the relative position of D and $\Phi(D)$. This relative position can be determined from a set of qualitative constraints that are called *parameter inequalities* (de Jong *et al.* 2002a). More precisely, the parameter inequalities specify a total ordering of the p_i threshold concentrations of gene i, as well as the possible target equilibrium levels μ_i^D/ν_i^D of x_i in all regulatory domains $D \subseteq \Omega$. The parameter inequalities for the example network described by (3)-(4) are given by

$$0 < \theta_a^1 < \theta_a^2 < \kappa_a/\gamma_a < max_a, \quad (5)$$

$$0 < \theta_b^1 < \theta_b^2 < \kappa_b/\gamma_b < max_b. \quad (6)$$

They constrain $\Phi(D^1)$ to lie somewhere in D^{25}, so that trajectories starting in D^1 reach one of the domains D^2, D^6, and D^7 at some point. The information necessary to specify the parameter inequalities can usually be inferred from the biological data.

A domain D supplemented by the relative position of D and $\Phi(D)$ will be called a *qualitative state* of the system. Given the qualitative state associated with D, it can be inferred which domains can be reached, in finite time, by trajectories starting in D. Since a qualitative state can be associated to each of these domains in turn, this amounts to the computation of *transitions* between qualitative states. In (de Jong *et al.* 2002a), a simulation algorithm is described that recursively generates qualitative states and transitions from qualitative states, starting at the qualitative state associated with an initial domain D^0. This results in a *transition graph*, a directed graph of qualitative states and transitions between qualitative states. The transition graph may contain *qualitative equilibrium states* or *qualitative cycles*. These may correspond to equilibrium points or limit cycles reached by solutions, and hence indicate functional modes of the regulatory system. Moreover, it has been shown that the transition graph produced by the qualitative simulation algorithm is guaranteed to cover all possible solutions of the PL model of the genetic regulatory network. The qualitative simulation algorithm is sound (de Jong *et al.* 2002a).

Figure 2(c) shows the transition graph generated for the example network, when starting in the regulatory domain

D^1. It shows that the system has a choice between three qualitative equilibrium states, two of which are stable (QS^4 and QS^{16}) and one of which is unstable (QS^7). This conforms to the expected behavior of the system, which is a simplified version of a well known molecular switch determining the response of *E. coli* to phage λ infection (Ptashne 1992).

For the purpose of validating models of genetic regulatory networks, it is usually more convenient to consider a refined version of the transition graph. Here the qualitative states are associated with (hyper)rectangular regions in the phase space where the derivatives of the concentration variables have a determinate sign. Often these hyperregions coincide with the domains defined above, for instance in the case of regulatory domain D^1, where $\dot{x}_a > 0$ and $\dot{x}_b > 0$, for all $\boldsymbol{x} \in D^1$. However, sometimes a domain may need to be divided into subdomains, to each of which a separate qualitative state is associated. In that case, transitions may need to be added between the refined qualitative states, in order to keep the soundness property. The refined transition graph can be deduced from the transition graph described in the previous paragraph. In our simple example, the refinement is straightforward. However, this may not be true in general (the automatization of this step is currently under way). In what follows, we assume that the transition graph generated by the qualitative simulator is the refined transition graph.

The qualitative simulation method described in this section has been implemented in Java 1.3 in the program *Genetic Network Analyzer (GNA)* (de Jong *et al.* 2003). GNA is available for non-profit academic research purposes at (GNA 2003). The core of the system is formed by the simulator, which generates a transition graph from a qualitative PL model and initial conditions. The input of the simulator is obtained by reading and parsing text files specified by the user. A graphical user interface (GUI), named *VisualGNA*, assists the user in specifying the model of a genetic regulatory network as well as in interpreting the simulation results.

Analysis of genetic regulatory networks by model checking

We have presented above how predictions of the behavior of a genetic regulatory network can be obtained by qualitative simulation. The model of the network, expressing hypotheses on the genes and proteins involved and their mutual interactions, can be validated by means of experimental data. The validation of a model is complicated by the size of the transition graphs obtained through simulation, which for networks with more than a dozen genes become too big to analyze by hand.

Our aim is to develop a method that can be used to test automatically if a transition graph satisfies an observed property. In this section, we propose an approach based on model checking. Model-checking techniques are widely used for the formal analysis of discrete state systems. Computer tools exists that can test automatically if a given property, expressed as a temporal logic statement, is satisfied by a discrete state system, represented by a Kripke structure. They combine formal precision and computational efficiency.

Expressing observed properties in temporal logic

As a first step in the validation of a model, we must express properties of the observed behavior of a genetic regulatory network in a formal language, here a *temporal logic* (Clarke, Grumberg, & Peled 1999). That is, we have to define the set of atomic propositions that will be used to describe the states of the system and choose an appropriate temporal logic.

The atomic propositions we will consider describe qualitative properties of the value of protein concentrations, since the qualitative simulation method yields predictions of this kind. More particularly, the atomic propositions concern the range in which a protein concentration falls and the sign of the derivative of the protein concentration. Let Λ_i be the set of concentration landmarks for gene i, defined as

$$\Lambda_i = \{0, \theta_i^1, \ldots, \theta_i^{p_i}, max_i\} \cup \{\mu_i^D/\nu_i^D \mid D \text{ regulatory domain}\}.$$

We now introduce the variables $range(x_i)$ and $sign(\dot{x}_i)$.

Definition 1 A state of a regulatory system is described using the variables $range(x_i)$ and $sign(\dot{x}_i)$, $1 \leq i \leq n$. The domains of these variables are $\mathcal{D}_{range(x_i)}$ and $\mathcal{D}_{sign(\dot{x}_i)}$, respectively, where $\mathcal{D}_{range(x_i)}$ is the set of (semi-)open or closed intervals $R_i \subseteq \Omega_i$, such that $inf(R_i), sup(R_i) \in \Lambda_i$, and $\mathcal{D}_{sign(\dot{x}_i)}$ is the set $\{-1, 0, 1, \top\}$.

$range(x_i) = R_i$ is interpreted as meaning that the concentration x_i lies between the two landmark concentrations $inf(R_i)$ and $sup(R_i)$. $sign(\dot{x}_i) = s_i$ is interpreted as meaning that the sign of the derivative of x_i is positive, negative, or zero, if s_i equals 1, -1, or 0, respectively. The special value $\top$ is used to express that $\dot{x}_i$ does not have a unique sign. This may occur in certain switching domains, as a consequence of the extension of the differential equations (1) to differential inclusions (see previous section).

We can define the set of atomic propositions in terms of $range(x_i)$ and $sign(\dot{x}_i)$.

Definition 2 The set of atomic propositions $\mathcal{AP}$ is given by:

$$\mathcal{AP} = \{range(x_i) = r_i,\ sign(\dot{x}_i) = s_i \mid r_i \in \mathcal{D}_{range(x_i)},\ s_i \in \mathcal{D}_{sign(\dot{x}_i)},\ 1 \leq i \leq n\}.$$

For example, $range(x_a) =]0, \theta_a^1]$, $range(x_b) =]\kappa_b/\gamma_b, max_b]$, $sign(\dot{x}_a) = -1$, and $sign(\dot{x}_b) = \top$ are valid atomic propositions.

Of the several temporal logics that exist (Emerson 1998), we have chosen to use *Computation Tree Logic (CTL)*. Four our purposes, a CTL formula is verified by a qualitative state of the system if the possible qualitative behaviors starting from that state satisfy the formula. A CTL formula consists in atomic propositions connected by operators. The operators are either the usual logical operators ($\neg$, $\vee$, $\wedge$, $\Rightarrow$, ...) or a restricted combination of path quantifiers and temporal operators. The path quantifiers A or E are used, respectively, to specify that all or some of the behaviors starting at a state have some property. The temporal operators describe properties that hold during a behavior. X, F, or G are temporal operators used to specify that the neXt state, some Future state, or (Globally) all future states in a behavior satisfy

some property. In CTL a path quantifier is necessarily paired with a temporal operator (see Clarke and Emerson (1981) for the formal syntax and semantics of CTL).

CTL, unlike some other temporal logics, allows us to quantify over the behaviors of the system. This is necessary for our application, since an observation provides information on one particular behavior, but not on all possible behaviors. Efficient algorithms for performing CTL model-checking exist (Clarke, Grumberg, & Peled 1999), which is a key issue for the practical use of the method.

As an example of the use of CTL, consider the observation that, in the system of figure 1, the concentrations x_a and x_b increase at first, while x_a is steady and x_b decreases afterwards. This can be expressed by means of the following CTL statement:

$$EF(sign(\dot{x}_a) = 1 \wedge sign(\dot{x}_b) = 1 \wedge EF(sign(\dot{x}_a) = 0 \wedge sign(\dot{x}_b) = -1)). \quad (7)$$

The CTL statement says that, from the initial state onwards, there Exists at least one behavior of the system leading to some Future state in which (1) the concentrations x_a and x_b increase, and (2) from that state onwards, there Exists at least one behavior leading to some Future state in which x_a is steady and x_b decreases.

Translating transition graph into Kripke structure

In the framework of CTL model checking, the discrete state system is described by means of a Kripke structure. A *Kripke structure* $\mathcal{M}$ over the set of atomic propositions $\mathcal{AP}$ is a four-tuple $\mathcal{M} = \langle S, S_0, R, L\rangle$, where S is a finite set of states, $S_0 \subseteq S$ the set of initial states, $R \subseteq S \times S$ a total transition relation and $L : S \to 2^{\mathcal{AP}}$ a function that labels each state with the atomic propositions true in that state (Clarke, Grumberg, & Peled 1999).

We have to define how to generate a Kripke structure from the transition graph produced by the qualitative simulator. Recall that a transition graph consists of qualitative states and transitions between qualitative states. Every qualitative state in the transition graph is defined as $QS = \langle SD, \boldsymbol{s}\rangle$, where $SD = SD_1 \times \ldots \times SD_n$ is a hyperrectangular region included in a domain D and $\boldsymbol{s} = (s_1, \ldots, s_n)'$ the sign vector of the derivatives $\dot{\boldsymbol{x}}$. The information contained in a qualitative state can be straightforwardly expressed in terms of the atomic predicates $\mathcal{AP}$ of definition 2. This gives the following Kripke structure corresponding to a transition graph.

Definition 3 A Kripke structure $\mathcal{M} = (S, S_0, R, L)$ over $\mathcal{AP}$ corresponds to a transition graph produced by the qualitative simulator, if

1. S is the set of qualitative states in the transition graph;
2. S_0 is the set of initial qualitative states;
3. $R \subseteq S \times S$ the transition relation, such that $R(QS, QS')$ holds, iff there is a transition from QS to QS' in the transition graph, or $QS = QS' = \langle SD, \boldsymbol{s}\rangle$ and $SD \subseteq D$, such that $\Phi(D) \cap D \neq \{\}$;
4. $L : S \to 2^{\mathcal{AP}}$ such that for all $QS = \langle SD, \boldsymbol{s}\rangle$,

$$L(QS) = \{range(x_i) = SD_i, sign(\dot{x}_i) = s_i \mid 1 \leq i \leq n\}.$$

It can be shown that the transition relation in the definition is total. The Kripke structure corresponding to the transition graph obtained from qualitative simulation of the example network in figure 1 is shown figure 3.

Checking if model is validated by observations

When properties of the observed behavior of the system have been expressed in CTL, and the transition graph obtained through qualitative simulation translated into a Kripke structure, the validation of the model is straightforward to achieve. Highly-efficient algorithms for CTL model checking have been developed and implemented in publicly-available computer tools. We will use NuSMV2, a symbolic model checker that combines BDD-based and SAT-based model-checking components (Cimatti *et al.* 2002).

The key steps of the approach advocated in this paper can be summarized as follows:

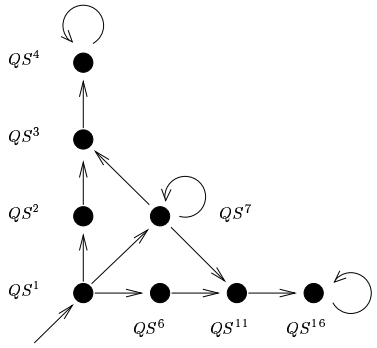
1. Perform a qualitative simulation of the genetic regulatory network;
2. Translate the resulting transition graph into a Kripke structure;
3. Formulate properties of the observed behavior of the system as a CTL statement;
4. Use NuSMV2 to test the validity of the model of the network.

The validation of the model gives rise to one of two results. First, there may be a qualitative behavior predicted from the model satisfying the observed properties of the system. In this case, we say that the model is corroborated by the observations. Second, if there is no qualitative behavior predicted from the model satisfying the observed properties of the system, then the model is invalidated by the observations. Recall that the transition graph produced by the qualitative simulation algorithm is guaranteed to cover all possible solutions of the PL model of the genetic regulatory network. This is critical for the decision to reject or revise a model when it is invalidated by the observations.

The approach sketched above can be illustrated by means of the simple network of two genes and their mutual interactions. Using the Kripke structure derived from the transition graph (figure 3), we can check whether the observation formulated as the CTL statement (7) is consistent with the model. The test of this property by means of NuSMV2 gives a positive answer. The reader can verify that this answer is correct by looking at the path $(QS^1, QS^6, QS^{11}, QS^{16})$ in the Kripke structure in figure 3.

Applicability of the approach

The previous section has given an outline of the use of model checking techniques in the analysis of genetic regulatory networks. Although we have given a proof of principle by applying the approach to an example of a small network, one can legitimately ask whether it is applicable to the genetic regulatory networks actually studied by biologists in their laboratory. Below we will argue that this is indeed the case,



L	$range(x_a)$	$range(x_b)$	$sign(\dot{x}_a)$	$sign(\dot{x}_b)$
QS^1	$[0, \theta_a^1[$	$[0, \theta_b^1[$	1	1
QS^2	$[0, \theta_a^1[$	$[\theta_b^1, \theta_b^1]$	$\top$	1
QS^3	$]0, \theta_a^1[$	$]\theta_b^1, \theta_b^2[$	-1	1
QS^4	$]0, \theta_a^1[$	$[\theta_b^2, \theta_b^2]$	-1	0
QS^6	$[\theta_a^1, \theta_a^1]$	$[0, \theta_b^1[$	1	$\top$
QS^7	$[\theta_a^1, \theta_a^1]$	$[\theta_b^1, \theta_b^1]$	0	0
QS^{11}	$]\theta_a^1, \theta_a^2[$	$]0, \theta_b^1[$	1	-1
QS^{16}	$[\theta_a^2, \theta_a^2]$	$]0, \theta_b^1[$	0	-1

Figure 3: Kripke structure corresponding to the transition graph obtained from the qualitative simulation of the example network in figure 1. The labeling function is shown separately in the adjacent table.

illustrating our arguments by means of the network controlling the initiation of sporulation in the bacterium *Bacillus subtilis*.

Qualitative modeling and simulation of sporulation network

Under conditions of nutrient deprivation, *B. subtilis* cells may cease to divide and form a dormant, environmentally-resistant spore instead (Burkholder & Grossman 2000). The decision to either divide or sporulate is controlled by a regulatory network integrating various environmental, cell-cycle, and metabolic signals. A graphical representation of the network is shown in figure 4, displaying key genes and their promoters, proteins encoded by the genes, and the regulatory action of the proteins.

Sporulation in *B. subtilis* is one of the best-understood model systems for prokaryotic development. However, notwithstanding the enormous amount of work devoted to the elucidation of the network of interactions underlying the sporulation process, very little quantitative data on kinetic parameters and molecular concentrations are available. This has motivated the use of the qualitative formalism described at the beginning of this paper to model the sporulation network and to simulate the response of the cell to nutrient deprivation.

The graphical representation of the network has been translated into a PL model supplemented by qualitative constraints on the parameters (de Jong *et al.* 2003). The resulting model consists of nine state variables and two input variables. The 49 parameters are constrained by 58 parameter inequalities, the choice of which is largely determined by biological data. Simulation of the sporulation network by means of GNA reveals that essential features of the initiation of sporulation in wild-type and mutant strains of *B. subtilis* can be reproduced by means of the model (de Jong *et al.* 2003). In particular, the choice between vegetative growth and sporulation is seen to be determined by competing positive and negative feedback loops influencing the accumulation of the phosphorylated transcription factor Spo0A. Above a certain threshold, Spo0A∼P activates various genes whose expression commits the bacterium to sporulation, such as genes coding for sigma factors that control the alternative developmental fates of the mother cell and the spore.

Towards the analysis of sporulation network by means of model checking

Although the predictions obtained by qualitative simulation lack numerical precision, the sporulation example illustrates that they do nevertheless capture essential features of the dynamics of the regulatory system and provide interesting insights into the underlying regulatory logic. However, the conclusions summarized above were arrived at through painstaking manual analyses of the transition graphs produced by the simulator, usually consisting of several hundreds of states. The proposed model-checking approach can be used to speed up the analysis and reduce interpretation errors of the modeler, induced by the failure to extract crucial information from the transition graph. We will give two examples to illustrate that experimental data used to validate a model can be expressed in terms of temporal logic.

Figure 5 represents the expression of two genes in the course of the sporulation process in a *B. subtilis* strain (Perego & Hoch 1988). The authors have used an experimental technique in which the specific activity of an enzyme (here β-galactosidase) reflects the expression of the gene. The lowest curve represents the expression of the gene *hpr*, which "increased in proportion of the growth curve, reached a maximum level at the early stationary phase [($T1$)], and remained at the same level during the stationary phase" ((Perego & Hoch 1988), p. 2564). This interpretation can be expressed by means of the CTL statement $EF(sign(\dot{x}_{hpr}) = 1 \wedge EFEG(sign(\dot{x}_{hpr}) = 0))$, where x_{hpr} denotes the concentration of Hpr. This formula can be paraphrased as "starting from the initial state, there exists at least one behavior of the system leading to a future state in which the concentration of Hpr is increasing, and continuing from which there exists at least one behavior leading to a future state continuing from which there exists a behavior in which the concentration of Hpr is constant.

Under conditions of nutrient deprivation, a fraction of the cells in a *B. subtilis* culture enters sporulation, whereas the other cells continue to divide. In Chung *et al.* (1994) this phenomenon is related to the observation that "within

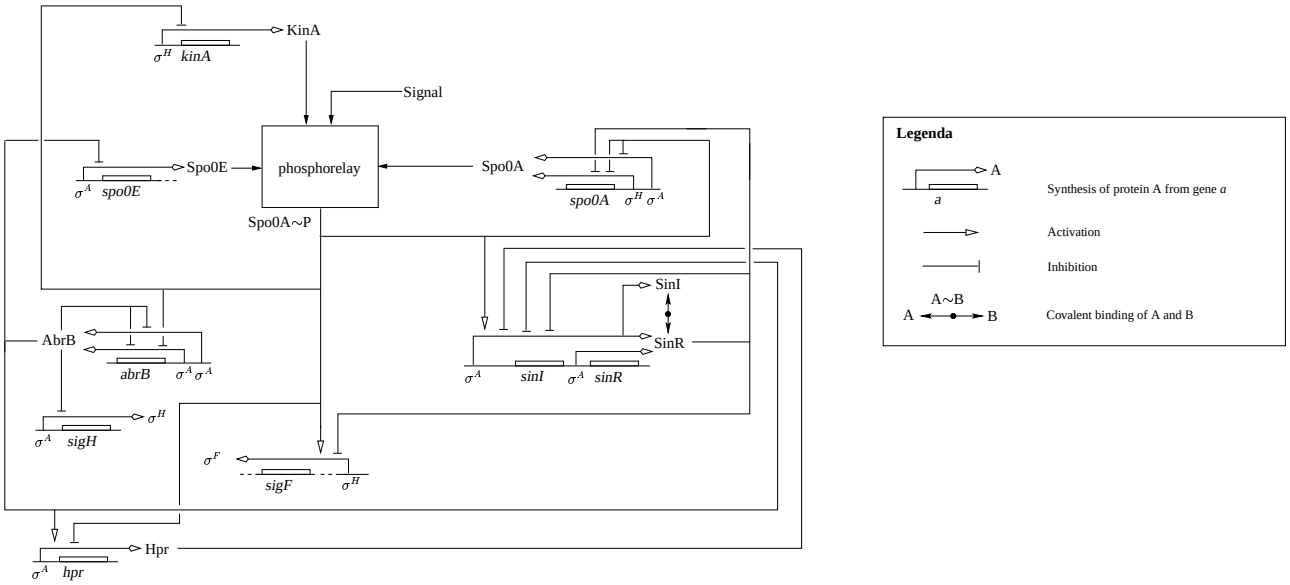


Figure 4: Key genes, proteins, and regulatory interactions making up the network involved in *B. subtilis* sporulation. In order to improve the legibility of the figure, the control of transcription by the sigma factors σ^A and σ^H has been represented implicitly, by annotating the promoter with the sigma factor in question.

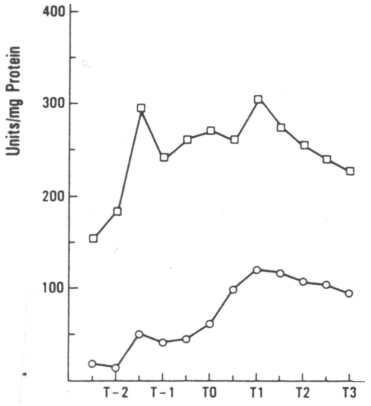


Figure 5: Time-series data showing the expression of two genes during sporulation in a wild-type *B. subtilis* strain (Perego & Hoch 1988).

a culture of sporulating cells of *B. subtilis*, there are two distinct subpopulations, one that has initiated the developmental program [leading to sporulation] ... and one in which early developmental gene expression remains uninduced" (p. 1977). The gene *sigF*, shown in figure 4, is an example of such a developmental gene. Representing the concentration of the protein σ^F encoded by *sigF* by the variable x_{sigF}, the above expression can be translated into the following CTL statement: $EF(range(x_{sigF}) = [0, \theta_{sigF}[) \wedge EF(range(x_{sigF}) =]\theta_{sigF}, max_{sigF}])$. Here, θ_{sigF} and max_{sigF} denote a threshold and the maximum concentration of the protein. This simply states that, starting from the initial state, there exist two behaviors of the system, one leading to a future state characterized by a low concentration of σ^F (below the threshold), the other leading to a state characterized by a high concentration of σ^F (above the threshold).

These two examples illustrate that temporal logic formulas can be used for expressing biological observation in a formal manner. They illustrate also that the formalization of the observation is not an easy task, as a sentence given in natural language may correspond to several CTL formulas, having a slightly different meaning, and thus possibly yielding different results.

Discussion

We have presented an approach towards the analysis of genetic regulatory networks based on the combination of qualitative simulation and model-checking techniques. The approach consists of the translation of the transition graph produced through qualitative simulation into a Kripke structure and the expression of observed properties of the behavior of a system in temporal logic. Using an existing efficient model-checking tool, the validity of the model of a genetic regulatory network can be tested. We have shown the in-principle feasibility of the approach on a simple network of two genes and argued for its applicability to networks actually studied by biologists.

The integration of qualitative simulation and model checking has been proposed before as a remedy for the analysis of the large number of qualitative behaviors produced by qualitative simulators. Shults and Kuipers (1997) have combined QSIM and CTL*, whereas Brajnik and Clancy (1998) have focused on QSIM and a variant of PLTL. Our work differs from these approaches in that, apart from a different temporal logic, we employ a qualitative simulation method tailored to a class of PL models. This allows us to

deal with large and complex genetic regulatory networks. Several groups are currently working on the application of model-checking techniques to the analysis of biochemical networks. As in this paper, Antoniotti *et al.* (2003) and Chabrier and Fages (2003) have chosen CTL, but they work with either completely numerical models or rather simple rule-based models. The advantage of the qualitative models used in our approach is that they are at the same time biologically valid and actually applicable.

Further work will focus on the implementation of the approach sketched in this paper and its application to the analysis of the initiation of sporulation in *B. subtilis* and other regulatory processes in prokaryotes.

References

Antoniotti, M.; Park, F.; Policriti, A.; Ugel, N.; and Mishra, B. 2003. Foundations of a query and simulation system for the modeling of biochemical and biological processes. In Altman, R.; Dunker, A.; Hunter, L.; and Klein, T., eds., *Proceedings of the Pacific Symposium on Biocomputing, PSB 2003*, 116–127.

Brajnik, G., and Clancy, D. 1998. Focusing qualitative simulation using temporal logic: theoretical foundations. *Annals of Mathematics and Artificial Intelligence* 22(1-2):59–86.

Burkholder, W., and Grossman, A. 2000. Regulation of the initiation of endospore formation in *Bacillus subtilis*. In Brun, Y., and Shimkets, L., eds., *Prokaryotic Development*. Washington, DC: American Society for Microbiology. chapter 7, 151–166.

Chabrier, N., and Fages, F. 2003. Symbolic model checking of biochemical networks. In C.Priami., ed., *Computational Methods in Systems Biology(CMSB-03)*, volume 2602 of *Lecture Notes in Computer Science*. Berlin: Springer-Verlag. 149–162.

Chung, J.; Stephanopoulos, G.; Ireton, K.; and Grossman, A. 1994. Gene expression in single cells of *Bacillus subtilis*: Evidence that a threshold mechanism controls the initiation of sporulation. *Journal of Bacteriology* 176(7):1977–1984.

Cimatti, A.; Clarke, E.; Giunchiglia, E.; Giunchiglia, F.; Pistore, M.; Roveri, M.; Sebastiani, R.; and Tacchella, A. 2002. NuSMV2: An OpenSource tool for symbolic model checking. In Brinksma, E., and Larsen, K. G., eds., *Proceedings of the 14th International Conference on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, 359–364. Berlin: Springer-Verlag.

Clarke, E., and Emerson, E. 1981. Design and synthesis of synchronisation skeletons using branching-time temporal logic. In Kozen, D., ed., *Logic of Programs*, number 131 in Lecture Notes in Computer Science, 52–71. Berlin: Springer-Verlag.

Clarke, E.; Grumberg, O.; and Peled, D. 1999. *Model Checking*. Boston, MA: MIT Press.

de Jong, H.; Page, M.; Hernandez, C.; and Geiselmann, J. 2001. Qualitative simulation of genetic regulatory networks: Method and application. In Nebel, B., ed., *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-01*, 67–73. San Mateo, CA: Morgan Kaufmann.

de Jong, H.; Gouzé, J.-L.; Hernandez, C.; Page, M.; Sari, T.; and Geiselmann, H. 2002a. Qualitative simulation of genetic regulatory networks using piecewise-linear models. Technical Report RR-4407, INRIA. Submitted for publication.

de Jong, H.; Gouzé, J.-L.; Hernandez, C.; Page, M.; Sari, T.; and Geiselmann, J. 2002b. Dealing with discontinuities in the qualitative simulation of genetic regulatory networks. In van Harmelen, F., ed., *Proceedings of Fifteenth European Conference on Artifical Intelligence, ECAI-02*, 412–416. Amsterdam: IOS Press.

de Jong, H.; Geiselmann, J.; Hernandez, C.; and Page, M. 2003. Genetic Network Analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics* 19(3):336–344.

de Jong, H. 2002. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology* 9(1):69–105.

Emerson, E. 1998. Temporal and modal logic. In van Leeuwen, J., ed., *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics. Cambridge, MA: MIT Press. 995–1072.

Glass, L., and Kauffman, S. 1973. The logical analysis of continuous non-linear biochemical control networks. *Journal of Theoretical Biology* 39:103–129.

GNA. 2003. `http://www-helix.inrialpes.fr/gna`.

Gouzé, J.-L., and Sari, T. 2003. A class of piecewise linear differential equations arising in biological models. *Dynamical Systems* 17(4):299–316.

Kuipers, B. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge, MA: MIT Press.

Mestl, T.; Plahte, E.; and Omholt, S. 1995. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology* 176:291–300.

Perego, M., and Hoch, J. 1988. Sequence analysis of the *hpr* locus, a regulatory gene for protease production and sporulation in *Bacillus subtilis*. *Journal of Bacteriology* 170(6):2560–2567.

Ptashne, M. 1992. *A Genetic Switch: Phage λ and Higher Organisms*. Cambridge, MA: Cell Press & Blackwell Science, 2nd edition.

Shults, B., and Kuipers, B. 1997. Proving properties of continuous systems: Qualitative simulation and temporal logic. *Artificial Intelligence* 92(1-2):91–130.

Thomas, R., and d'Ari, R. 1990. *Biological Feedback*. Boca Raton, FL: CRC Press.