Analysis and Verification of Qualitative Models of Genetic Regulatory Networks: A Model-Checking Approach*

Grégory Batt[1], **Delphine Ropers**[1], **Hidde de Jong**[1], **Johannes Geiselmann**[2], **Radu Mateescu**[1], **Michel Page**[1,3], **Dominique Schneider**[2]
[1]INRIA Rhône-Alpes, Montbonnot, France, [2]Laboratoire Adaptation et Pathogénie des Microorganismes, Université Joseph Fourier, Grenoble, France, [3]Université Pierre Mendès France, Grenoble, France

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. We apply our approach to the analysis of the complex network controlling the nutritional stress response in the bacterium *Escherichia coli*.

1 Introduction

Qualitative simulation aims at making predictions of the behavior of dynamical systems in the absence of detailed, quantitative information on parameter values and functional relations [Kuipers, 1994]. The idea of qualitative simulation has attracted much interest in the context of a biological application, the analysis of *genetic regulatory networks*, that is, the networks of regulatory interactions between genes, proteins, and small molecules underlying the development and functioning of all living organisms. On the one hand, mathematical methods supported by computer tools are indispensable for the analysis of genetic regulatory networks: most networks of interest involve many genes connected through complex feedback loops, thus making an intuitive understanding of their dynamics difficult to obtain. On the other hand, numerical simulation methods are difficult to apply, because only a few networks are well-understood on the molecular level, and quantitative information on the interactions is seldom available.

We have previously developed a method for the qualitative simulation of genetic regulatory networks [de Jong *et al.*, 2004b]. This 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 and Kauffman, 1973]. This allows it to deal with large and complex networks of regulatory interactions. The qualitative simulation method has been implemented in the publicly-available computer tool Genetic Network Analyzer (GNA) [de Jong *et al.*, 2003]. GNA has been used to analyze several genetic regulatory networks of biological interest (*e.g.*, [de Jong *et al.*, 2004a]).

Given a model of a genetic regulatory network, the above qualitative simulation method produces a state transition graph, consisting of qualitative states and transitions between qualitative states. The graph describes the possible qualitative behaviors of the network. When simulating large and complex genetic regulatory networks, the state transition graph usually consists of hundreds or even thousands of states, as a consequence of which the analysis of the graph by visual inspection alone becomes error-prone or even practically infeasible. In order to analyze large state transition graphs, previous work in qualitative simulation has proposed the use of *model-checking* techniques [Brajnik and Clancy, 1998; Shults and Kuipers, 1997]. These techniques allow for the efficient verification of properties of discrete transition systems [Clarke *et al.*, 1999].

The aim of this paper, extending preliminary ideas in [Batt *et al.*, 2003], is to demonstrate the effectiveness and practical applicability of the combined use of qualitative simulation and model checking for the analysis of large state transition graphs produced by the qualitative simulation of genetic regulatory networks. In particular, we show how state transition graphs can be reformulated as so-called Kripke structures on which properties of the behavior of the network, expressed in temporal logic, can be verified. In addition, we present a new implementation of GNA that connects the qualitative simulator to state-of-the-art model checkers.

We have applied the above approach to the analysis of the complex regulatory network controlling the nutritional stress response in the bacterium *Escherichia coli*. Among other results, the use of model-checking tools has allowed the characterization of cycles in a large state transition graph obtained through qualitative simulation of the network. The cycles correspond to biologically-important phenomena that are currently being experimentally tested in our laboratory.

In the next section 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 application of this approach to the analysis of the nutritional stress response network is the subject of the next section. We finish with a discussion of the approach in the context of related work.

*This paper also appears in the proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI'05.

2 Qualitative modeling and simulation 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 and Kauffman, 1973]:

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

where $\boldsymbol{x} = (x_1, \ldots, x_n)' \in \Omega$ is a vector of cellular protein concentrations, $\boldsymbol{f} = (f_1, \ldots, f_n)'$, $\boldsymbol{g} = \mathrm{diag}(g_1, \ldots, g_n)$, and $\Omega \subset \mathbb{R}^n_{\geq 0}$ is a bounded n-dimensional phase space box. The rate of change of each protein concentration x_i, $i \in \{1..n\}$, is thus 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 $f_i : \Omega \to \mathbb{R}_{\geq 0}$ expresses how the rate of synthesis of the protein encoded by gene i depends on the concentrations $\boldsymbol{x}$ of the proteins in the cell. More specifically, the function f_i is defined as a sum of terms having the general form $\kappa^l_i\, b^l_i(\boldsymbol{x})$, where $\kappa^l_i > 0$ is a rate parameter, and $b^l_i : \Omega \to \{0, 1\}$ a piecewise-continuous *regulation function* defined in terms of step functions s^+ and s^-:

$$s^+(x_i, \theta_i) = \begin{cases} 1, \text{ if } x_i > \theta_i, \\ 0, \text{ if } x_i < \theta_i, \end{cases} \text{ and } s^-(x_i, \theta_i) = 1 - s^+(x_i, \theta_i),$$

where x_i is an element of the state vector $\boldsymbol{x}$ and θ_i a constant denoting a threshold concentration for x_i.

The function g_i expressing the regulation of protein degradation is defined analogously, except that we demand that g_i is strictly positive. Notice that with the above definitions, $\boldsymbol{h}$ is a *piecewise-linear (PL)* vector-valued function. An example of a simple genetic regulatory network and its PL model is shown in Figure 1(a)-(b).

The use of step functions $s^{\pm}(x_i, \theta_i)$ in (1) gives rise to complications, because the step functions are discontinuous at $x_i = \theta_i$, and therefore $\boldsymbol{h}$ is discontinuous on $\Theta = \bigcup_{i \in [1..n], l_i \in [1..p_i]} \{\boldsymbol{x} \in \Omega \mid x_i = \theta^{l_i}_i\}$, the union of the threshold hyperplanes. In order to deal with this problem, following an approach widely used in control theory, we extend the differential *equation* (1), defined on $\Omega \setminus \Theta$, to the differential *inclusion*

$$\dot{\boldsymbol{x}} \in H(\boldsymbol{x}), \tag{2}$$

defined on Ω, where $H(\boldsymbol{x})$ is the smallest closed *hyperrectangular* set containing the set of all limit values of $\boldsymbol{h}(\boldsymbol{y})$, for $\boldsymbol{y} \notin \Theta$ and $\boldsymbol{y} \to \boldsymbol{x}$ [Gouzé and Sari, 2002; de Jong *et al.*, 2004b; Batt *et al.*, 2005]. A *solution* of the differential inclusion (2) on a time interval I is an absolutely-continuous vector-valued function $\boldsymbol{\xi}(t)$ such that $\dot{\boldsymbol{\xi}}(t) \in H(\boldsymbol{\xi}(t))$ almost everywhere on I.

Formally, we define the *PL system* Σ as the triple (Ω, Θ, H), that is, the set-valued function H given by (2), defined on the n-dimensional phase space box Ω, with Θ the union of the threshold hyperplanes. Also, we denote by Ξ_Σ the set of all solutions $\boldsymbol{\xi}(t)$, on a finite or infinite time interval, of the differential inclusion (2), that reach and leave a threshold hyperplane finitely-many times.

The mathematical analysis of PL systems has shown that mere knowledge of the relative order of threshold parameters and ratios of synthesis and degradation parameters, describing the local strength of gene expression, is sufficient for the definition of a finite partition $\mathcal{D}$ of the phase space Ω into hyperrectangular regions, called *domains*. This partition is such that the derivatives of the solutions in each domain have a unique *sign pattern* (Figure 1(b)-(d)). The definition of the sign pattern $S(\boldsymbol{x})$ for a point $\boldsymbol{x}$ in a domain D is complicated by the fact that the solution of a differential inclusion is in general not unique. In particular, if $\boldsymbol{x}$ lies on a threshold hyperplane, $H(\boldsymbol{x})$ is multi-valued, and several solutions $\boldsymbol{\xi}(t)$ may pass through $\boldsymbol{x}$ with different derivative signs. We therefore define $S(\boldsymbol{x}) = \{\boldsymbol{sign}(\dot{\boldsymbol{\xi}}(t_x)) \mid \boldsymbol{\xi} \in \Xi_\Sigma \text{ in } D, \boldsymbol{\xi}(t_x) = \boldsymbol{x}, \text{ and } \dot{\boldsymbol{\xi}}(t_x) \in H(\boldsymbol{\xi}(t_x))\}$. We have proven that $S(\boldsymbol{x})$ is the same for every $\boldsymbol{x} \in D$ [Batt *et al.*, 2005].[1]

Using the domain partition $\mathcal{D}$ of the phase space Ω, together with the above qualitative characterization of the dynamics in each of the domains, we can discretize the continuous dynamics. In the resulting abstract description, the state of the system is represented by a domain, and there exists a transition from a domain D to a domain D', if and only if there exists a solution reaching D' from D, without leaving $D \cup D'$. This leads to the introduction of the so-called *qualitative transition system*, $\Sigma\text{-QTS} = (\mathcal{D}, \to, \Pi, \models_\Pi)$, where $\mathcal{D}$ is the set of all domains, $\to \subseteq \mathcal{D} \times \mathcal{D}$ is a transition relation describing all transitions between the domains, $\Pi = \{Dsign = S \mid S \in 2^{\{-1,0,1\}^n}\}$ is a set of propositions describing the derivative sign patterns, and $\models_\Pi \subseteq \mathcal{D} \times \Pi$ is a satisfaction relation that associates to each domain a qualitative description of the dynamics of the system in the domain, defined by $D \models_\Pi Dsign = S$ iff for all $\boldsymbol{x} \in D$: $S(\boldsymbol{x}) = S$.

The graph representation of the qualitative transition system, $G = (\mathcal{D}, \to)$, is called a *state transition graph*. The domains are also called *qualitative states*. A qualitative state D is called *instantaneous*, if no solution remains in D longer than a time instant, and *persistent*, otherwise. Moreover, it is a *qualitative equilibrium state*, if $D \models_\Pi Dsign = S$, with $\mathbf{0} \in S$, since then for every $\boldsymbol{x} \in D$, $\boldsymbol{\xi}(t) = \boldsymbol{x}$ is a solution. The state transition graph for the example network is shown in Figure 1(e). A path in the graph describes a possible qualitative behavior of the system (Figure 1(f)).

In [Batt *et al.*, 2005], the qualitative transition system is defined as the *discrete quotient* of a continuous transition system having the same reachability properties as the original PL system Σ, given the equivalence relation induced by the partition $\mathcal{D}$. Using standard results from hybrid systems theory [Alur *et al.*, 2000], it directly follows that the qualitative transition system Σ-QTS is a *conservative approximation* of Σ, in the sense that if a solution $\boldsymbol{\xi}$ of the PL system Σ, defined on some time interval I, passes through the time-ordered sequence of domains $(D^0, \ldots, D^m)$, then $(D^0, \ldots, D^m)$ is a path in Σ-QTS. On the other hand, it may happen that no solution of Σ passes through a sequence of domains corresponding to a path in Σ-QTS.[2]

Simple rules exploiting the favorable mathematical properties of the class of PL models have been formulated such that the qualitative transition system can be symbolically com-

[1] In previous work [de Jong *et al.*, 2004b], a coarser-grained partition of Ω is used, for which this property does not hold.

[2] In the terminology of qualitative simulation, the method is sound but incomplete.

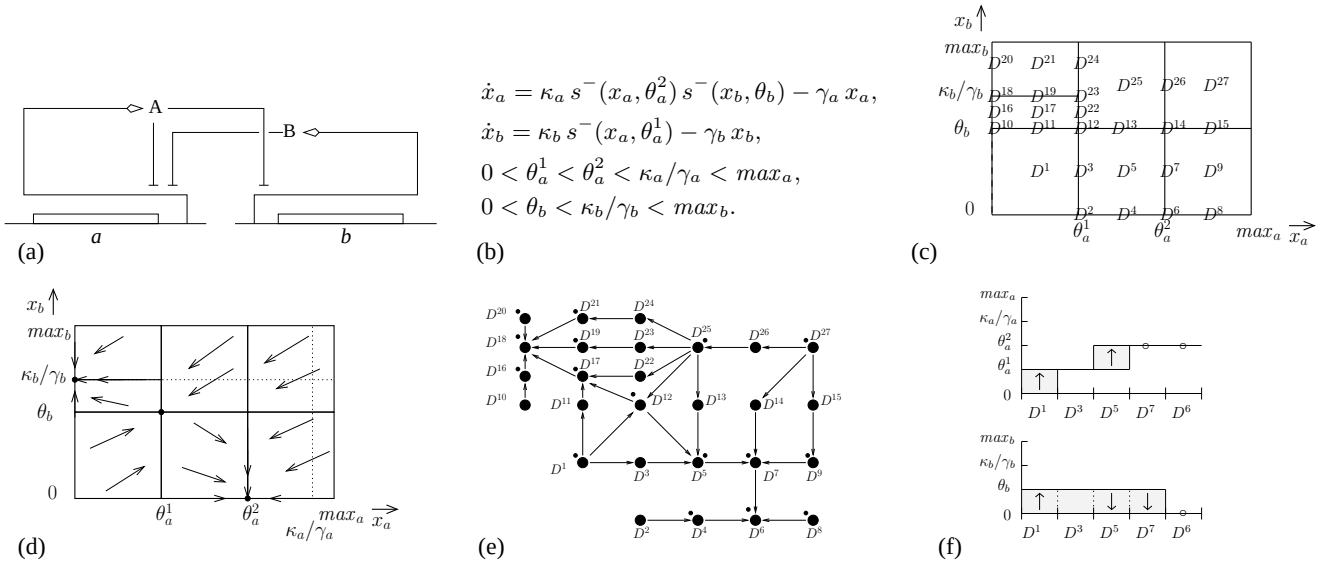


Figure 1: Qualitative simulation of a two-gene network. (a) Graphical representation (legend in Figure 3(a)). Gene *b* encodes protein B, which inhibits gene *a* expression, that is, protein A synthesis. (b) Model consisting of PL differential equations and parameter inequalities. x_a and x_b are the protein concentrations, κ_a and κ_b (γ_a and γ_b) the synthesis (degradation) rates of the proteins, and θ_a^1, θ_a^2, and θ_b threshold concentrations. The second differential equation states that the synthesis rate of protein B is κ_b if x_a is below θ_a^1, and 0 if x_a is above θ_a^1, and that its degradation rate is $\gamma_b x_b$. The parameter inequalities for protein B state that gene *b* may inhibit gene *a* expression, because when synthesized, protein B concentration tends towards the level κ_b/γ_b, for which $s^-(x_b, \theta_b) = 0$. (c) Domain partition of the phase space Ω: $\mathcal{D} = \{D^1, \ldots D^{27}\}$. (d) Sketch of the flow in Ω showing the unicity of the derivative sign pattern in each domain. Dots denote equilibrium points. (e) Graph representation of the qualitative transition system, Σ-QTS. Dots self-transitions. (f) Temporal evolution of the protein concentrations in the path $(D^1, \ldots, D^6)$. Arrows indicate the derivative signs in persistent states ($\uparrow$ for $S = \{1\}$; $\downarrow$ for $S = \{-1\}$; $\circ$ for $S = \{0\}$).

puted from a PL model of the network and a set of parameter inequalities defining the relative order of the parameters [Batt *et al.*, 2005]. The implementation of these rules has resulted in a new version of the computer tool *Genetic Network Analyzer (GNA)*, available at http://www-helix.inrialpes.fr/gna.

3 Analysis of genetic regulatory networks by model checking

Models of genetic regulatory networks of biological interest may produce large state transition graphs, thus making manual analysis of dynamical properties error-prone or even practically infeasible. To address this problem, we propose to combine the qualitative modeling and simulation method outlined above with techniques for *model checking* [Clarke *et al.*, 1999]. These techniques allow for the efficient verification of properties of the behavior of discrete transition systems, expressed as formulae in some temporal logic.

Various model-checking frameworks exist, differing by their expressiveness, user-friendliness, and computational efficiency. For the sake of clarity of exposition, we focus on one particular framework in this paper, in which the discrete transition system takes the form of a *Kripke structure*, and the behavioral properties are expressed in *Computation Tree Logic (CTL)* [Clarke *et al.*, 1999]. However, we emphasize that our approach is not restricted to CTL model-checking, and allows other, more expressive temporal logics to be used as well (Section 3.3).

3.1 Translate qualitative transition system into Kripke structure

As a preliminary step, we introduce a set of *atomic propositions*, AP, to describe the state of the system. More precisely, the set of atomic propositions we use consists of simple expressions describing a protein concentration (*e.g.*, $value\_x_i = \theta_i$, $value\_x_i < \kappa_i/\gamma_i$), the sign of its derivative (*e.g.*, $dsign\_x_i = \{1\}$), or the type of a state ($type = pers$, $type = inst$, $type = eq$, for persistent, instantaneous, and equilibrium state, respectively). The set of atomic propositions for the two-gene example is shown in Figure 2(a).

We define the *Kripke structure* over AP corresponding to Σ-QTS as Σ-KS $= (\mathcal{D}, \rightarrow, \models_{AP})$, where $\models_{AP} \subseteq \mathcal{D} \times 2^{AP}$ is the satisfaction relation associating to every domain D the atomic propositions that hold in D. The existence of at least one solution $\boldsymbol{\xi}$ on some time interval $[0, \tau]$, $\tau > 0$, with initial condition $\boldsymbol{\xi}(0) = \boldsymbol{x}_0$, for every $\boldsymbol{x}_0 \in \Omega$, guarantees that $\rightarrow$ is total (*i.e.*, every state has a successor), which is required for Kripke structures. The Kripke structure for the two-gene example is shown in Figure 2(b).

3.2 Express dynamical properties in temporal logic

CTL formulae are interpreted with respect to a Kripke structure. A CTL formula is built upon atomic propositions. The usual operators from propositional logic, such as negation ($\neg$), disjunction ($\vee$), conjunction ($\wedge$), and implication ($\rightarrow$), can also be used. In addition, CTL provides two types of op-

(a) $AP = \{\ value\_x_a = 0, value\_x_a > 0, \ldots, dsign\_x_a = \{-1\}, dsign\_x_a = \{0\}, \ldots, type = inst, type = eq\ \}$.

(b) $\Sigma\text{-KS} = (\mathcal{D}, \rightarrow, \models_{AP})$, with

- $\mathcal{D} = \{D^1, D^2, D^3, \ldots, D^{27}\}$,
- $\rightarrow = \{\ (D^1, D^1), (D^1, D^2), (D^1, D^{11}), (D^1, D^{12}), \ldots, (D^{27}, D^{26})\ \}$,
- $\models_{AP} = \{\ (D^1, value\_x_a \geq 0), (D^1, value\_x_a < \theta_a^1), \ldots, (D^1, dsign\_x_a = \{1\}), (D^1, dsign\_x_b = \{1\}), (D^1, type = pers),$
  $(D^2, value\_x_a = \theta_a^1), \ldots, (D^2, type = inst), \ldots, (D^{27}, value\_x_a > \theta_a^2), \ldots, (D^{27}, type = pers)\ \}$.

(c)

Biological property	CTL formula	$\mathcal{D}_0$	NuSMV result and diagnostic
• The network reaches a state in which both protein concentrations increase, and from that state onwards, a second state in which protein A concentration increases and that of protein B decreases	$\mathbf{EF}(dsign\_x_a = \{1\} \wedge dsign\_x_b = \{1\} \wedge \mathbf{EF}(dsign\_x_a = \{1\} \wedge dsign\_x_b = \{-1\}))$	$\{D^1\}$	True: $(D^1, D^3, D^5, D^7, D^6)$ (witness)
• The network will inevitably reach an equilibrium state where protein A has vanished	$\mathbf{AF}(type = eq \wedge value\_x_a = 0)$	$\{D^1\}$	False: $(D^1, D^3, D^5, D^7, D^6)$ (counterexample)
• A high level of protein B guarantees the eventual disappearance of protein A	$\mathbf{AG}(value\_x_b > \theta_b \rightarrow \mathbf{AF}\, value\_x_a = 0)$	$\mathcal{D}$	False: $(D^{25}, D^{13}, D^5, D^7, D^6)$ (counterexample)

Figure 2: Verification of properties of the two-gene network in Figure 1. (a) Set of atomic propositions AP. (b) Kripke structure Σ-KS over AP corresponding to the qualitative transition system Σ-QTS shown in Figure 1(e). (c) Biological properties, with their CTL translation, their truth value in Σ-KS for the set of initial states $\mathcal{D}_0$, and a diagnostic (witness or counterexample), as computed by NuSMV.

erators: path quantifiers, **E** and **A**, and temporal operators, such as **F** and **G**. Path quantifiers are used to specify that a property p holds for some ($\mathbf{E}\, p$) or every ($\mathbf{A}\, p$) path starting from a given state in the Kripke structure. Temporal operators are used to specify that, given a state and a path starting from that state, a property p holds for some ($\mathbf{F}\, p$) or for every ($\mathbf{G}\, p$) state of the path. Each path quantifier must be paired with a temporal operator [Clarke *et al.*, 1999].

A variety of biologically-interesting properties of the dynamics of a genetic regulatory network can be expressed in CTL [Chabrier-Rivier *et al.*, 2004]. Typical questions concern the *possibility* or the *inevitability* for the network to reach some state(s), the *absence* or the *universality* of some property, and *response* or *precedence* properties (*i.e.*, a property is always followed or preceded by some other property). Some examples are given for the two-gene network in Figure 2(c). More generally, these properties can be used for testing the validity of a model given experimental data, or understanding how the dynamics of a system emerges from the interactions between its components.

3.3 Check if model satisfies dynamical properties

The model-checking problem consists in verifying whether a temporal-logic formula holds in the discrete transition system at the initial state(s). Highly-efficient algorithms have been developed and implemented in model checkers for automatically solving this problem. Two widely-used model checkers are NuSMV [Cimatti *et al.*, 2002] and Evaluator, a component of the CADP toolbox [Mateescu and Sighireanu, 2003]. NuSMV is an efficient, state-of-the-art model checker for CTL, whereas Evaluator is an on-the-fly model checker for the regular alternation-free μ-calculus, a temporal logic based on regular expressions. The algorithmic complexity of these tools is linear in the size of the transition system and the formula. In addition to a yes/no answer, the tools return a diagnostic, either a witness or a counterexample, depending on whether the property holds or not.

In order to combine our qualitative simulator with model-checking tools, we have integrated export functionalities in GNA. The text files generated by GNA can be imported in the model checkers, after which the verification of the properties of interest continues in the environment of the latter tools. Figure 2(c) illustrates the verification of some properties of the two-gene network. Fairness conditions [Clarke *et al.*, 1999] have been used to ensure that attractors (equilibrium states, strongly connected components) are eventually reached (not shown).

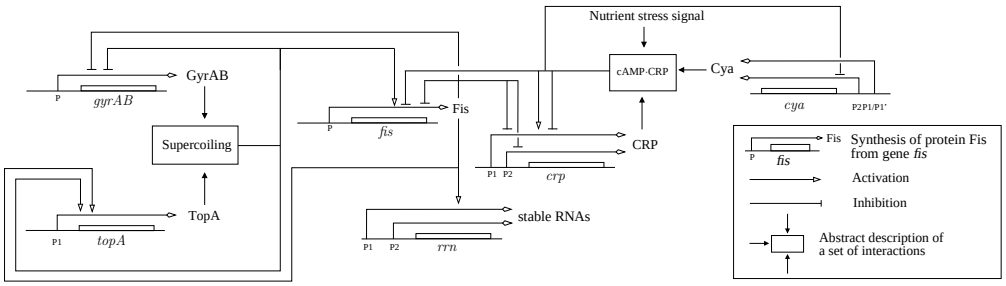
Recall that the qualitative abstraction used to obtain Σ-QTS from Σ is a conservative approximation. Can the conclusions drawn from the verification of a property on Σ-QTS (or more precisely on Σ-KS) be transposed to Σ? If *every* path in Σ-QTS satisfies a property, then from the conservativeness, a corresponding property holds for every solution of Σ. Conversely, if a property holds for *some* path in Σ-QTS, then, since this path may be spurious, the corresponding property may hold for no solution of Σ. One can define the set of temporal-logic formulae such that, if a property holds in Σ-QTS, it does so in Σ. In the CTL model-checking framework, this set is the set of formulae that can be expressed in the temporal logic called $\forall$CTL [Grumberg and Long, 1994].

4 Application: Analysis of the nutritional stress response in *E. coli*

In case of nutritional stress, an *E. coli* population abandons exponential growth and enters a non-growth state called *stationary phase*. This growth-phase transition is accompanied by numerous physiological changes in the bacteria [Huisman *et al.*, 1996]. On the molecular level, this transition is controlled by a complex genetic regulatory network integrating various environmental signals. A network including six genes believed to play a key role in the nutritional stress response is shown in Figure 3(a) [Ropers *et al.*, 2004].

A PL model of seven variables has been constructed, one protein concentration variable for each gene and one input variable representing the presence or absence of a nutritional stress signal. Seven differential equations and forty parameter inequalities describe the dynamics of the system (Figure 3(b)). Using the new version of GNA, we have simulated two phenomena, namely nutrient starvation, causing the

(a)



(b)

$$\dot{x}_{fis} = \kappa^1_{fis}\,(1 - s^+(x_{crp},\theta^1_{crp})\; s^+(x_{cya},\theta^1_{cya})\; s^+(x_s,\theta_s)\,)\; s^-(x_{fis},\theta^5_{fis}) + \kappa^2_{fis}\; s^-(x_{topA},\theta^2_{topA})\; s^+(x_{gyrAB},\theta^2_{gyrAB})\; s^-(x_{fis},\theta^5_{fis})(1 - s^+(x_{crp},\theta^1_{crp})\; s^+(x_{cya},\theta^1_{cya})\; s^+(x_s,\theta_s)\,) - \gamma_{fis}\; x_{fis}$$

$$0 < \theta^1_{fis} < \kappa^1_{fis}/\gamma_{fis} < \theta^2_{fis} < \theta^3_{fis} < \theta^4_{fis} < \theta^5_{fis} < (\kappa^1_{fis} + \kappa^2_{fis})/\gamma_{fis} < max_{fis}$$

(c)

Biological property	CTL formula	$\mathcal{D}_0$	NuSMV result	Time
• "The transcription of cya is negatively regulated by cAMP and CRP" [Kawamukai *et al.*, 1985]	$\mathbf{AG}(value\_x_{crp} > \theta^3_{crp} \wedge value\_x_{cya} > \theta^3_{cya} \wedge value\_u_s > \theta_s \rightarrow \mathbf{EF}\, dsign\_x_{cya} = \{-1\})$	$\{D^s\}$	True	$< 0.1s.$
• "rrn[*P1*] transcription increases independently of Fis [...] following [nutrient] upshift" [Appleman *et al.*, 1998]	$\mathbf{E}(value\_x_{fis} > \theta^3_{fis}\ \mathbf{U}(value\_x_{fis} > \theta^3_{fis} \wedge dsign\_x_{rrn} = \{1\}))$	$\{D^u\}$	False	$< 0.1s.$

Figure 3: Network of key genes, proteins, and regulatory interactions involved in the nutritional stress response in *E. coli* [Ropers *et al.*, 2004]. (b) PL differential equation and parameter inequalities for the protein Fis. (c) Some expected properties of the network. Their translation into CTL and interpretation are not detailed here. **U** is the standard CTL operator Until [Clarke *et al.*, 1999]. D^s and D^u correspond to the initial states for nutrient starvation and nutrient upshift, respectively.

transition from exponential to stationary phase, and nutrient upshift, leading to the reentry into exponential phase. The simulation results have been analyzed by means of the model-checker NuSMV, using the export functionality of GNA. Below, we give some examples.

The simulation of a nutrient starvation has given rise to a state transition graph of 66 states (27 of which are persistent), computed in less than 1 s. on a PC (800 MHz, 256 Mb). The graph contains a single equilibrium state corresponding to stationary-phase conditions. Are the predictions obtained from the model consistent with experimental data? In [Azam *et al.*, 1999], the measured concentration of the global regulator Fis is shown to decrease and become steady in stationary phase, which is characterized by a low concentration x_{rrn} of stable RNAs:

$$\mathbf{EF}(dsign\_x_{fis} = \{-1\} \wedge \mathbf{EF}(dsign\_x_{fis} = \{0\} \wedge value\_x_{rrn} < \theta_{rrn})) \quad (3)$$

The verification of (3) takes a fraction of a second to complete and shows that the observed temporal evolution of the Fis concentration is reproduced by the model.

We have also studied the reentry into exponential phase after a nutrient upshift. Using the same model as above, but starting the simulation from the qualitative state characterizing stationary-phase conditions and with the nutritional stress signal switched off, qualitative simulation results in a state transition graph of 1143 states (202 of which are persistent), generated in 1.7 s. The graph contains several strongly connected components (SCCs). One of these SCC, containing 222 states, can be reached from all states in the graph. The qualitative transition system satisfies the property

$$\mathbf{AG}(statesInSCC \rightarrow \mathbf{AG}\, statesInSCC), \quad (4)$$

where the predicate $statesInSCC$ is satisfied by all and only states in the SCC. That is, if the system has reached this SCC, it always remains in it. Checking this property by visual inspection is tedious and error-prone, while it takes NuSMV only 3.3 s. Further mathematical analysis has revealed that the cyclic paths correspond to solutions spiraling inwards to an equilibrium point [Ropers *et al.*, 2004]. In other words, during the reentry into stationary phase, the concentrations of some of the proteins oscillate towards a new equilibrium level. This is a surprising result, which is currently subject to experimental verification in our laboratory. We also checked a dozen of other properties, two of which are shown in Figure 3(c).

5 Discussion

We propose a combination of qualitative simulation and model-checking techniques for the analysis and verification of qualitative models of genetic regulatory networks. Model checkers help in dealing with the problem that the state transition graphs generated by qualitative simulation may become prohibitively large. They permit complex dynamical properties, whose verification defeats manual analysis capabilities, to be efficiently and reliably checked. The analysis of the genetic regulatory network composed of key genes involved in the nutrient stress response in *E. coli* illustrates the applicability of the approach.

Model-checking techniques have been used before for analyzing biological networks. Most approaches start from discrete models, such as concurrent transition systems [Chabrier-Rivier *et al.*, 2004] and Boolean networks [Bernot *et al.*, 2004]. In this paper we show that model-checking techniques can also be used for more conventional continuous models, in particular differential equation models, when using qualitative abstractions to discretize the dynamics of the system. In comparison with ideas along the same line

[Brajnik and Clancy, 1998; Shults and Kuipers, 1997], our approach is adapted to a particular class of PL differential equations with favorable mathematical properties, allowing the development of tailored algorithms that scale up well to models of large and complex genetic regulatory networks.

The model-validation approach of this paper has been illustrated in the context of CTL model checking. While CTL allows a variety of biologically-meaningful properties to be expressed, some properties fall outside its scope. For instance, in Section 4 we would have liked to be able to express the occurence of oscillations in some of the protein concentrations after a nutrient upshift. The formula $\mathbf{EG}(p \rightarrow \mathbf{F}\neg p \ \wedge \ \neg p \rightarrow \mathbf{F}p)$ expresses this property, where p means that the concentration of some protein is above a certain threshold. Unfortunately it is not a CTL formula and it does not admit any CTL equivalent [Clarke and Draghicescu, 1988]. However, the property can be expressed in μ-calculus and evaluated using XTL, a component of the CADP toolbox [Mateescu and Garavel, 1998]. The capability of GNA to generate export files for different model checkers, allows one to take advantage from the specific strengths of each of these.

References

[Alur *et al.*, 2000] R. Alur, T.A. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proc. IEEE*, 88(7):971–84, 2000.

[Appleman *et al.*, 1998] J.A. Appleman, W. Ross, J. Salomon, and R.L. Gourse. Activation of *E. coli* rRNA transcription by Fis during a growth cycle. *J. Bacteriol.*, 180(6):1525–32, 1998.

[Azam *et al.*, 1999] T. Ali Azam, A. Iwata., A. Nishimura, S. Ueda, and A. Ishihama. Growth phase-dependent variation in protein composition of the *E. coli* nucleoid. *J. Bacteriol.*, 181(20):6361–70, 1999.

[Batt *et al.*, 2003] G. Batt, H. de Jong, J. Geiselmann, and M. Page. Analysis of genetic regulatory networks: A model-checking approach. *QR'03*, 31–38. 2003.

[Batt *et al.*, 2005] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, M. Page, and D. Schneider. Qualitative analysis and verification of hybrid models of genetic regulatory networks. *HSCC'05*, *LNCS* 3414, 134–50. Springer, 2005.

[Bernot *et al.*, 2004] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. A fruitful application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. *J. Theor. Biol.*, 229(3):339–47, 2004.

[Brajnik and Clancy, 1998] G. Brajnik and D.J. Clancy. Focusing qualitative simulation using temporal logic. *Ann. Math. Artif. Intell.*, 22(1-2):59–86, 1998.

[Chabrier-Rivier *et al.*, 2004] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theor. Comput. Sci.*, 325(1):25–44, 2004.

[Cimatti *et al.*, 2002] A. Cimatti *et al.* NuSMV2: An open-source tool for symbolic model checking. *CAV'02*, *LNCS* 2404, 359–64. Springer, 2002.

[Clarke and Draghicescu, 1988] E.M. Clarke and I.A. Draghicescu. Expressibility results for linear-time and branching-time logics. *REX Workshop*, *LNCS* 354, 428–37. Springer, 1988.

[Clarke *et al.*, 1999] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 1999.

[de Jong *et al.*, 2003] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic Network Analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–44, 2003.

[de Jong *et al.*, 2004a] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in *B. subtilis*. *Bull. Math. Biol.*, 66(2):261–99, 2004.

[de Jong *et al.*, 2004b] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.*, 66(2):301–40, 2004.

[Glass and Kauffman, 1973] L. Glass and S.A. Kauffman. The logical analysis of continuous non-linear biochemical control networks. *J. Theor. Biol.*, 39(1):103–29, 1973.

[Gouzé and Sari, 2002] J.-L. Gouzé and T. Sari. A class of piecewise-linear differential equations arising in biological models. *Dyn. Syst.*, 17(4):299–316, 2002.

[Grumberg and Long, 1994] O. Grumberg and D.E. Long. Model checking and modular verification. *ACM Trans. Program. Lang. and Syst.*, 16(3):1512–42, 1994.

[Huisman *et al.*, 1996] G.W. Huisman, D.A. Siegele, M.M. Zambrano, and R. Kolter. Morphological and physiological changes during stationary phase. In F.C. Neidhardt *et al.*, eds, *E. coli and Salmonella: Cellular and Molecular Biology*, 1672–82. ASM Press, 1996.

[Kawamukai *et al.*, 1985] M. Kawamukai, J. Kishimoto, R. Utsumi, M. Himeno, T. Komano, and H. Aiba. Negative regulation of adenylate cyclase gene (*cya*) expression by cyclic AMP-cyclic AMP receptor protein in *E. coli*. *J. Bacteriol.*, 164(2):872–77, 1985.

[Kuipers, 1994] B.J. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, 1994.

[Mateescu and Garavel, 1998] R. Mateescu and H. Garavel. XTL: A meta-language and tool for temporal logic model-checking. *STTT'98*, 33–42, 1998.

[Mateescu and Sighireanu, 2003] R. Mateescu and M. Sighireanu. Efficient on-the-fly model-checking for regular alternation-free mu-calculus. *Sci. Comput. Program.*, 46(3):255–81, 2003.

[Ropers *et al.*, 2004] D. Ropers, H. de Jong, M. Page, D. Schneider, and J. Geiselmann. Qualitative simulation of nutritional stress response in *E. coli*. Technical Report RR-5412, INRIA, 2004.

[Shults and Kuipers, 1997] B. Shults and B.J. Kuipers. Proving properties of continuous systems: Qualitative simulation and temporal logic. *Artif. Intell.*, 92:91–130, 1997.