Temporal Logic Patterns for Querying Qualitative Models of Genetic Regulatory Networks*

Pedro T. Monteiro[1,2], Delphine Ropers[1], Radu Mateescu[1], Ana T. Freitas[2], and Hidde de Jong[1]

[1]INRIA Grenoble - Rhône-Alpes, 655 Av. de l'Europe, Montbonnot, 38334 St. Ismier Cedex, France
{Pedro.Monteiro, Delphine.Ropers, Radu.Mateescu, Hidde.de-Jong}@inrialpes.fr
[2]IST/INESC-ID, 9 Rua Alves Redol, 1000-029 Lisboa, Portugal
atf@inesc-id.pt

Abstract

Formal verification based on model checking provides a powerful technology to query qualitative models of dynamical systems. The application of model-checking approaches is hampered, however, by the difficulty for non-expert users to formulate appropriate questions in temporal logic. In order to deal with this problem, we propose the use of patterns, that is, high-level query templates capturing recurring questions which can be automatically translated to temporal logic. We develop a set of patterns for the analysis of qualitative models of genetic regulatory networks, which are sufficiently generic though to be useful in other application domains. The applicability of the patterns has been investigated by the analysis of a model of the network of global regulators controlling the carbon starvation response in *Escherichia coli*.

Introduction

Qualitative simulation provides predictions of the possible qualitative behavior of a dynamical system (Kuipers 1994). It is an attractive approach when little or no quantitative information on parameter values is available, or when one is interested in the range of possible qualitative behaviors compatible with the structure of the system. These conditions are often met in the analysis of biological systems, which explains the popularity of qualitative approaches in mathematical and theoretical biology (*e.g.*, (Batt et al. 2007; Bellazzi et al. 2001; King, Garrett, and Coghill 2005; Thomas, Thieffry, and Kaufman 1995)). An example is the method for the *qualitative simulation of genetic regulatory networks* described in (Batt et al. 2007). This approach is based on a class of piecewise-linear (PL) differential equation models to describe regulatory interactions between genes, and has been implemented in the computer tool Genetic Network Analyzer (GNA).

A problem with the use of qualitative simulation is the potential explosion of the number of qualitative behaviors when dealing with large and complex systems whose dynamics cannot be sufficiently constrained. In order to deal with this problem, the use of *model-checking techniques* has been proposed (Shults and Kuipers 1997). This approach was successfully explored for the validation of qualitative models of genetic regulatory networks, by coupling GNA to state-of-the-art model checkers (Batt et al. 2005). It allows model predictions to be verified by experimental observations expressed as statements in temporal logic.

Formal verification based on model checking provides a powerful technology to query qualitative models, but it raises new issues, notably that of formulating good questions when analyzing a large model. Posing relevant and interesting questions is critical in modeling in general, but even more so in the context of applying formal verification techniques, due to the fact that it is not easy for non-experts to formulate queries in temporal logic. The response to this problem proposed by the formal verification community is the use of *patterns*, that is, high-level query templates that capture recurring questions in a specific application domain and that can be automatically translated to temporal logic (Dwyer, Avrunin, and Corbett 1999). This approach does not seem to have received any attention in qualitative reasoning thus far.

The aim of this paper is to develop a set of patterns for the analysis of models of genetic regulatory networks. Its main contributions are twofold. First, we develop a set of generic query templates, based on a review of frequently-asked questions by modelers, and translate these templates to temporal logic formulas.Although the patterns have been formulated for the analysis of genetic regulatory networks, they are sufficiently generic to carry over to other application domains. Second, we show the interest of the patterns in a case-study, concerned with the analysis of a large and complex model of the *E. coli* carbon starvation response. This model extends a previous model (Ropers et al. 2006) by taking into account additional regulators of bacterial stress responses.

Patterns for querying qualitative models

Description of network dynamics

As a basic hypothesis, we assume that the dynamics of genetic regulatory networks can be modeled by means of *finite state transition systems* (FSTSs) (Clarke, Grumberg, and Peled 1999). The latter formalism provides a general description of a dynamical system that explicitly underlies GNA (Batt et al. 2007), but the predictions of other qualita-

*This paper also appears in the proceedings of the Eighteenth European Conference on Artificial Intelligence, ECAI'08.

tive simulators can also be mapped to FTSTs. The generality of the FSTS formalism is important for assuring the wide applicability of the patterns developed in this section. Moreover, statements in temporal logic are usually interpreted on FSTSs, so that the latter naturally connect qualitative models to model-checking tools.

A finite state transition system is formally defined as a tuple $\Sigma = \langle S, AP, L, T, S_0 \rangle$, where S is a set of states, AP is a set of atomic propositions, $L : S \rightarrow 2^{AP}$ is a labeling function that associates to a state $s \in S$ the set of atomic propositions satisfied by s, $T \subseteq S \times S$ is a relation defining transitions between states, and $S_0 \subseteq S$ is a set of initial states. For our purpose, S describes the possible states of the genetic regulatory network, each of which is characterized by a set of atomic propositions, such as that the concentration of protein P is above a threshold and increasing.

Identification of patterns

The notion of patterns was introduced in the domain of software engineering as a means to capture expert solutions to recurring problems in program design. In the formal verification domain they have been introduced in an influential paper (Dwyer, Avrunin, and Corbett 1999), to help non-expert users formulate their temporal-logic queries. In the latter context, patterns are high-level descriptions of frequently asked questions in an application domain that are formulated in structured natural language rather than temporal logic. The aim of the patterns is not to cover all possible questions an expert can think of, but rather to simplify the formulation of those that are primary.

The difficulty of proposing patterns is to come up with a limited number of query schemas that are sufficiently generic to be applicable in a variety of situations, and at the same time sufficiently concrete to be comprehensible for the non-expert user. Moreover, the overlap between the patterns should be minimal. We analyzed a large number of modeling studies in systems biology (starting from the references in (Szallazi, Periwal, and Stelling 2006)), as well as lists of temporal logic queries (*e.g.*, (Chabrier-Rivier et al. 2004)). This bibliographic research allowed us to identify an open-ended list of questions on the dynamics of genetic, metabolic, and signal transduction networks. For instance, "Is the basal glycerol production level combined with rapid closure of Fps1 sufficient to explain an initial glycerol accumulation after osmotic shock?" (Klipp et al. 2005).

The identified questions were grouped into four categories, depending on whether they concerned the *occurrence/exclusion*, *consequence*, *sequence*, and *invariance* of cellular events. For each of these, we developed an appropriate pattern, capturing the essence of the question and the most relevant variants.

Description of patterns

The patterns consist of structured natural language phrases, represented in schematic form, with placeholders for so-called *state descriptors*. A state descriptor is a statement expressing a state property, and takes the form of (a Boolean combination of) atomic propositions. Let ϕ, ψ be state descriptors, then

$$\begin{aligned} \phi, \psi ::= & \; p_1 \in AP \mid p_2 \in AP \mid \ldots \\ ::= & \; \neg\phi \mid \phi \wedge \psi \mid \phi \Rightarrow \psi \mid \ldots \end{aligned}$$

The state descriptors are interpreted on the FSTS, in the sense that their meaning is formally defined as the set of states $S_1 \subseteq S$ satisfying the state descriptor. In addition to (Boolean combinations of) atomic propositions, the state descriptors may be temporal-logic formulas defined on the atomic propositions AP. However, the precise definition of the state descriptors depends on the particular type of FSTS that is used, as the latter determines AP.

Definition 1 (Occurrence/exclusion pattern)

It	is possible	for a state	ϕ	to occur
	is not possible			

This pattern represents the concepts of *occurrence* and its negation, *exclusion* (to capture safety properties). It will often be used during the development of a model to check for the presence or absence of some property that was experimentally observed. For instance, "It is possible for a state with a high concentration of protein P_1 to occur". Using this pattern, we can also check for *mutual exclusion*, by using the pattern negative form in combination with a conjunctive state descriptor. For instance, "It is not possible for a state to occur in which genes g_1 and g_2 are highly expressed".

Definition 2 (Consequence pattern)

If a state	ϕ	occurs,

then it is	possibly	followed by a state	ψ
	necessarily		

The *consequence* pattern relates two events separated in time. More precisely, it expresses that if the first state occurs, then it is possibly or necessarily followed by the second state. If the latter state necessarily follows, then the consequence pattern expresses a form of causal relation. An instance of this pattern is, for example, "If a state occurs in which the concentration of protein P is below 5 μM, then it is necessarily followed by a state in which the expression of gene *g* is at its basal level".

Definition 3 (Sequence pattern)

A state	ψ	is reachable and

is	possibly	preceded	at some time	by a state	ϕ
	necessarily		all the time		

The *sequence* pattern represents an ordering relation between two events. It ought not to be confused with the *consequence* pattern, since the conditional occurrence of the second state which characterizes the latter is absent in the *sequence* pattern. It must be possible to observe both the first and the second state, in that order, for an instance of the *sequence* pattern to be true.

Four variants of the pattern are distinguished, depending on whether the second state follows possibly or necessarily

Occurrence/Exclusion pattern	**CTL**	**μ-calculus**
It **is possible** for a state ϕ to occur	$EF\ (\phi)$	$\mu X.(\phi \vee \Diamond X)$
It **is not possible** for a state ϕ to occur	$\neg EF\ (\phi)$	$\neg\mu X.(\phi \vee \Diamond X)$
Consequence pattern		
If a state ϕ occurs, then it is **possibly** followed by a state ψ	$AG\ (\phi \Rightarrow EF\ (\psi))$	$\nu X.((\phi \Rightarrow \mu Y.(\psi \vee \Diamond Y)) \wedge \Box X)$
If a state ϕ occurs, then it is **necessarily** followed by a state ψ	$AG\ (\phi \Rightarrow AF\ (\psi))$	$\nu X.((\phi \Rightarrow \mu Y.(\psi \vee \Box Y)) \wedge \Box X)$
Sequence pattern		
A state ψ is reachable and is **possibly** preceded **at some time** by a state ϕ	$EF\ (\phi \wedge EF\ (\psi))$	$\mu X.((\phi \wedge \mu Y.(\psi \vee \Diamond Y)) \vee \Diamond X)$
A state ψ is reachable and is **possibly** preceded **all the time** by a state ϕ	$E\ (\phi\ U\ \psi)$	$\mu X.(\psi \vee (\phi \wedge \Diamond X))$
A state ψ is reachable and is **necessarily** preceded **at some time** by a state ϕ	$EF\ (\psi) \wedge \neg E\ (\neg\phi\ U\ \psi)$	$\mu X.(\psi \vee \Diamond X) \wedge \neg\mu Y.(\psi \vee (\neg\phi \wedge \Diamond Y))$
A state ψ is reachable and is **necessarily** preceded **all the time** by a state ϕ	$EF\ (\psi) \wedge AG\ (\neg\phi \Rightarrow AG\ (\neg\psi))$	$\mu X.(\psi \vee \Diamond X)\ \wedge \nu Y.((\phi \vee \nu Z.(\neg\psi \wedge \Box Z)) \wedge \Box Y)$
Invariance pattern		
A state ϕ **can** persist indefinitely	$EG\ (\phi)$	$\nu X.(\phi \wedge \Diamond X)$
A state ϕ **must** persist indefinitely	$AG\ (\phi)$	$\nu X.(\phi \wedge \Box X)$

Table 1: Rules for the translation of the patterns into CTL and μ-calculus. For each of the four patterns, the translation of all variants is shown. We use the version of μ-calculus presented in (Kupferman, Vardi, and Wolper 2000), which is interpreted on classical Kripke structures. The symbol T stands for True.

after the first state, and whether the system is in the first state all the time or only at some time before the occurrence of the second state. An instance of this pattern is "A steady state is reachable and is necessarily preceded all the time by a state in which nutrient N is absent".

Definition 4 (Invariance pattern)

A state	ϕ	can	persist indefinitely
		must	

The *invariance* pattern is used to check if the system can or must remain indefinitely in a state. In contrast with the *occurrence/exclusion* pattern, the question is not whether a particular state can be reached, but rather whether a particular state is invariable. An instance of this pattern is "A state with a basal expression of gene g must persist indefinitely".

Translation to temporal logic

By defining a temporal-logic translation of the patterns, the user queries can be automatically cast in a form that allows the verification of the specified property by means of model-checking tools. The patterns defined above are independent of a particular temporal logic, which allows the same high-level specification of a user query to be verified by means of different approaches and tools. It is worth noticing though that some of the patterns we propose have a branching-time nature (*e.g.*, the *consequence* and the *sequence* patterns), and therefore these are not translatable into a linear-time formalism, such as LTL (Clarke, Grumberg, and Peled 1999).

Two examples of translations of the previously defined patterns are shown in tabular form: the Computational Tree Logic (CTL) translation and the μ-calculus translation (Table 1). In both CTL and μ-calculus, formulas are built upon atomic propositions. Also, the usual connectors of propositional logic, such as negation ($\neg$), logical or ($\vee$), logical and ($\wedge$) and implication ($\Rightarrow$), can be used in both logics. In addition, CTL provides two types of operators: *path quantifiers*, **E** and **A**, and *temporal operators*, such as **F** and **G**. Path quantifiers are used to specify that a property p is satisfied by some (**E** p) or every (**A** p) path starting from a given state. Temporal operators are used to specify that, given a state and a path starting from that state, a property p holds for some (**F** p) or for every (**G** p) state of the path. Each path quantifier must be paired with a temporal operator. In the case of μ-calculus, two types of operators are provided: the least (μ) and greatest (ν) *fixed points*, and the *modal operators* possibility ($\Diamond$) and necessity ($\Box$). Least and greatest fixed points specify finite and infinite recursive applications of a formula, respectively. For instance, given a state and a path starting from that state, the fact that a property p holds for some state or for all states of the path is expressed using a least (μ) or a greatest (ν) fixed point, respectively. Modal operators are used to specify that, given a state, p possibly ($\Diamond$ p) or necessarily ($\Box$ p) holds on some or all of its outgoing states.

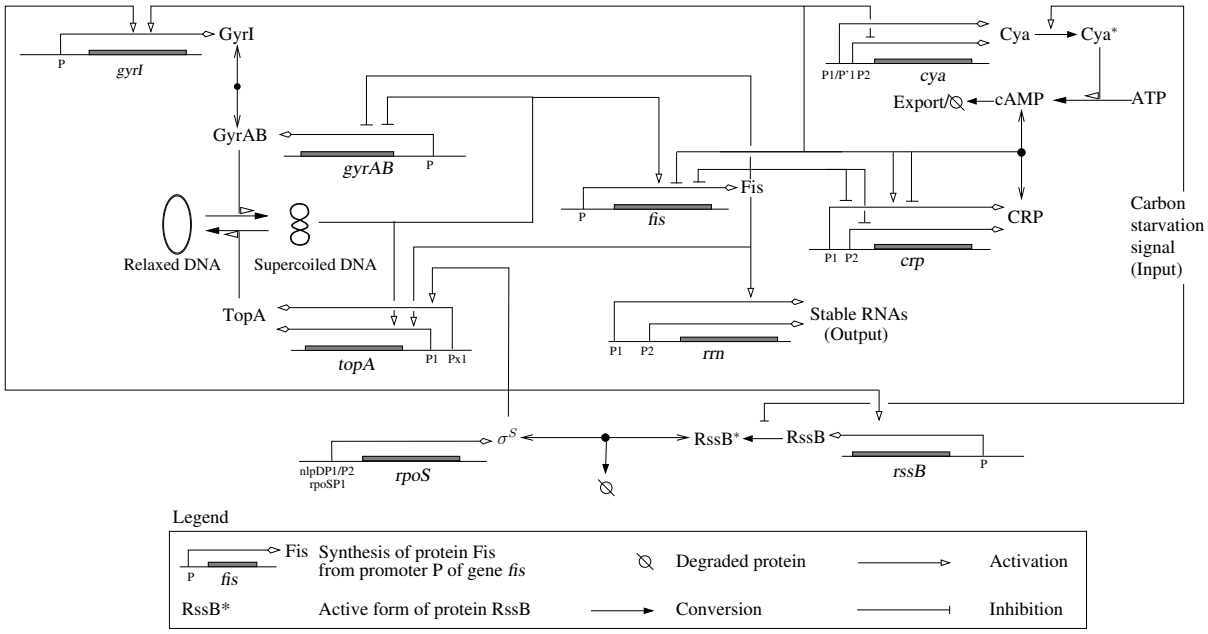
Carbon starvation response in *E. coli*

Model of carbon starvation response

To test the applicability of the temporal logic patterns, we have used our approach for the analysis of a model of the carbon starvation response in the bacterium *E. coli*. In the absence of essential carbon sources in its growth environment, 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, and controlled on the molecular level by a complex genetic regulatory network.

The molecular basis of the adaptation of the growth of *E. coli* to the nutritional conditions has been the focus of extensive studies for decades (Gutierrez-Ríos et al. 2007; Hengge-Aronis 1996). However, notwithstanding the enormous amount of information accumulated on the genes, proteins, and other molecules, kinetic parameters and the molecular concentrations are absent, with some exceptions, which makes it difficult to apply traditional methods for the dynamical modeling of genetic regulatory networks.

These circumstances have motivated the development of a qualitative model of the carbon starvation response net-



(a)

$$\dot{x}_{gyrAB} = \kappa_{gyrAB}\,(1 - s^{+}(x_{gyrAB}, \theta^{2}_{gyrAB})\; s^{-}(x_{gyrI}, \theta^{1}_{gyrI})\; s^{-}(x_{topA}, \theta^{1}_{topA}))\; s^{-}(x_{fis}, \theta^{4}_{fis}) - \gamma_{gyrAB}\; x_{gyrAB}$$

$$0 < \theta^{1}_{gyrAB} < \theta^{2}_{gyrAB} < \kappa_{gyrAB}/\gamma_{gyrAB} < max_{gyrAB} \quad \text{(b)}$$

Figure 1: (a) Network of key genes, proteins and regulatory interactions involved in the carbon starvation response network in *E. coli*. (b) PL differential equation and parameter inequality constraints for the gyrase GyrAB. The variable x_{gyrAB} denotes the concentration of GyrAB. The protein is produced at a rate κ_{gyrAB} if the DNA supercoiling level is not high, that is, if the concentration of GyrAB itself is below the threshold θ^{2}_{gyrAB}, and the concentrations of the topoisomerase TopA and the gyrase inhibitor GyrI are above the thresholds θ^{1}_{topA} and θ^{1}_{gyrI}, respectively. The regulatory logic of *gyrAB* expression is modeled by means of step functions. For instance, $s^{+}(x_{gyrAB}, \theta^{2}_{gyrAB})$ evaluates to 1, if $x_{gyrAB} > \theta^{2}_{gyrAB}$ (and to 0 otherwise). The protein is degraded at a rate proportional to its own concentration, $\gamma_{gyrAB}\; x_{gyrAB}$. The constraint $\theta^{2}_{gyrAB} < \kappa_{gyrAB}/\gamma_{gyrAB} < max_{gyrAB}$ express that the derepression of the *gyrAB* promoter allows the concentration of GyrAB to reach a high level, above the threshold θ^{2}_{gyrAB}. Instead of numerical values, the qualitative simulator uses such inequality constraints to infer behavior predictions (Batt et al. 2007; 2005).

work using a class of *piecewise-linear (PL) differential equations*. The PL models, originally introduced on (Glass and Kauffman 1973), provide a coarse-grained picture of the dynamics of genetic regulatory networks. They associate a protein concentration variable to each of the genes in the network, and capture the switch-like character of gene regulation by means of step functions that change their value at a threshold concentration of the proteins. The advantage of using PL models is that the qualitative dynamics of the high-dimensional systems are relatively simple to analyze, using inequality constraints on the parameters rather than exact numerical values (Batt et al. 2005; 2007). This makes the PL models a valuable tool for the analysis of the carbon starvation network.

In previous work we developed a PL model that we extend here by the general stress response factor RpoS and related regulators (Ropers et al. 2006; Ropers et al., in preparation). The dynamics of this system are described by nine coupled PL differential equations, and fifty inequality constraints on the parameter values.

Qualitative simulation of starvation response

The mathematical properties of the class of PL models used for modeling the stress response network have been well-studied (Glass and Kauffman 1973). It was previously shown how discrete abstractions can be used to convert the continuous dynamics of the PL system into a FSTS (Batt et al. 2007). The states S of the FSTS correspond to hyperrectangular regions in the concentration space, while the transitions T arise from trajectories entering one region from another. The atomic propositions AP describe, among other things, the concentration bounds of the regions and the trend of the variables inside a region (increasing, decreasing, or steady). The generation of the FSTS from the PL model has been implemented in the computer tool GNA (Batt et al. 2005). GNA is able to export the FSTS to standard model checkers like NuSMV (Cimatti et al. 2002) and CADP (Garavel, Lang, and Mateescu 2007), supporting the use of CTL and μ-calculus, respectively.

The application of this approach to the model of the *E. coli* carbon starvation network generates a huge FSTS. The entire state set consists of approximately $\mathcal{O}(10^{10})$ states, while the subset of states that is most relevant for our pur-

Properties	Response
Occurrence/exclusion pattern: Mutual inhibition of Fis and CRP	True
| It | is not possible | for a state | $x_{crp} \geq \frac{k^1_{crp}+k^2_{crp}+k^3_{crp}}{\gamma_{crp}} \wedge x_{fis} \geq \theta^4_{fis}$ | to occur | and | It | is not possible | for a state | $x_{crp} \leq \frac{k^1_{crp}}{\gamma_{crp}} \wedge x_{fis} \leq \theta^1_{fis}$ | to occur |	
CTL: $\neg EF(x_{crp} \geq \frac{k^1_{crp}+k^2_{crp}+k^3_{crp}}{\gamma_{crp}} \wedge x_{fis} \geq \theta^4_{fis}) \wedge \neg EF(x_{crp} \leq \frac{k^1_{crp}}{\gamma_{crp}} \wedge x_{fis} \leq \theta^1_{fis})$ μ-calculus: $\neg\mu X.((x_{crp} \geq \frac{k^1_{crp}+k^2_{crp}+k^3_{crp}}{\gamma_{crp}} \wedge x_{fis} \geq \theta^4_{fis}) \vee \Diamond X) \wedge \neg\mu X.((x_{crp} \leq \frac{k^1_{crp}}{\gamma_{crp}} \wedge x_{fis} \leq \theta^1_{fis}) \vee \Diamond X)$	
Consequence pattern: Damped oscillations after nutrient upshift	True
| If a state | $x_{signal} < \theta_{signal}$ | occurs, then it is | necessarily | followed by a state | $isOscillatoryState$ |	
CTL: $AG\ ((x_{signal} < \theta_{signal}) \Rightarrow AF\ (isOscillatoryState))$ μ-calculus: $\nu X.(((x_{signal} < \theta_{signal}) \Rightarrow \mu Y.(isOscillatoryState \vee \Box Y)) \wedge \Box X)$	
Sequence pattern: Control of entry into stationary phase by RpoS	True
| A state | $x_{rrn} < \theta_{rrn}$ | is reachable and is | necessarily | preceded | at some time | by a state | $x_{rpoS} \geq \theta^1_{rpoS}$ |	
CTL: $EF\ (x_{rrn} < \theta_{rrn}) \wedge \neg E\ (\neg(x_{rpoS} \geq \theta^1_{rpoS})\ U\ (x_{rrn} < \theta_{rrn}))$ μ-calculus: $\mu X.((x_{rrn} < \theta_{rrn}) \vee \Diamond X)\ \wedge \neg\mu Y.((x_{rrn} < \theta_{rrn}) \vee (\neg(x_{rpoS} \geq \theta^1_{rpoS}) \wedge \Diamond Y))$	
Invariance pattern: Expression of *topA* during growth-phase transitions	False
| A state | $x_{topA} < \theta^1_{topA}$ | can | persist indefinitely |	
CTL: $EG\ (x_{topA} < \theta^1_{topA})$ μ-calculus: $\nu X.((x_{topA} < \theta^1_{topA}) \wedge \Diamond X)$	

Table 2: Translation of properties used in the analysis of the *E. coli* carbon starvation response, following the translation rules in Table 1. The symbol $isOscillatoryState$ is a predicate attributed by the qualitative simulator to a state and indicating that the state is part of a cycle in the state transition graph.

pose, *i.e.* the states that are reachable from an initial state corresponding to a particular growth state of the bacteria, still consists of $\mathcal{O}(10^3)$ states. It is clear that FSTSs of this size cannot be analyzed by visual inspection, and that formal verification techniques are needed.

In the next section we show how the previously defined patterns can speed up the querying of these FSTSs, by simplifying the formulation of relevant properties to be tested.

Analysis of carbon starvation response model using query patterns

Four relevant properties were studied to analyze the *E.coli* carbon starvation response model (Table 2). The properties correspond to the following questions:

- Does the mutual inhibition motif of Fis and CRP (Fis inhibits the expression of gene *crp*, and CRP inhibits the expression of gene *fis*) have an effect on the dynamics of the carbon starvation response network?
- Is a carbon upshift a necessary condition for the occurrence of damped oscillations in the concentration of the regulators of the DNA supercoiling level?
- Is the entry into stationary phase always preceded by the accumulation of the stress response regulator RpoS?
- Is gene *topA* expressed in response to carbon source availability?

The instances of the patterns were translated into CTL following the translation rules of Table 1, and then verified using the model-checker NuSMV. The results are shown in the Table 2. By way of illustration we develop the formulation of the pattern for the third question and interpret the results of the verification process.

RpoS is a general stress response factor that allows cells to adapt to and survive under harmful conditions by entering stationary phase (Hengge-Aronis 1996). Due to its key role, the concentration of RpoS is tightly regulated, at the transcriptional, translational, and post-translational levels. The stability of the protein is mainly controlled in our conditions: while cells grow on a carbon source, RpoS is actively degraded through the protein RssB, which binds to RpoS and targets the factor to an intracellular protease. However, the depletion of the carbon source inactivates RssB, thus allowing RpoS to accumulate at a high concentration.

Given the importance of RpoS for cell survival, one may ask whether the entry into stationary phase is always preceded by the accumulation of RpoS in the cell. We formulated this question using a *sequence* pattern, where the stationary phase is represented by a low level of stable RNAs *rrn* (Table 2). The latter indicator is motivated by the fact that stationary-phase cells do not need high levels of these RNAs, which are necessary for the high translational activity of the exponential phase. The property is true, which indicates that the entry into stationary phase cannot occur before RpoS has accumulated. This points at the central role of RpoS in the growth adaptation of the bacteria.

Discussion

Formal verification techniques are promising tools for upscaling the analysis of qualitative models of genetic regulatory networks and other dynamical systems. The widespread adoption of model-checking approaches is restrained, however, by the difficulty for non-expert users to formulate appropriate questions in temporal logics. Inspired by work in the formal verification community (Dwyer, Avrunin, and

Corbett 1999), the first contribution of the paper consists in the formulation of a set of patterns in the form of query templates in structured natural language. In addition, we have provided translations of the patterns to two different temporal logics, CTL and μ-calculus. The patterns capture a large number of frequently-asked questions by modelers of regulatory networks, as for example listed in (Chabrier-Rivier et al. 2004). The second contribution of the paper concerns the instantiation of the patterns for the analysis of the complex genetic regulatory network involved in the carbon starvation response in *E. coli*. We have extended an existing model of the network with additional global regulators and verified the effect of the extensions on the predicted network dynamics.

The paper addresses issues we were confronted with when applying qualitative simulation techniques to a real-world problem in biology. We have proposed a solution, temporal logic query patterns for the analysis of large FSTSs, that has turned out to be useful in our application. However, we also expect this approach to carry over to other qualitative reasoning applications, where similar problems arise. Model checking is a promising way to analyze the large FSTSs arising in qualitative simulation (Shults and Kuipers 1997), but most modelers are not familiar with temporal logics and have difficulty in expressing their questions by means of these formalisms. Although meant to capture frequently-asked questions in biology, the patterns introduced in this paper are defined for FSTSs in general and seem sufficiently generic to apply to other problems as well. At the very least, they form a good starting-point for the formulation of a new set of query templates, tailored to the specificities of qualitative applications in other domains.

Acknowledgments

This work was partially supported by FCT program (PhD grant SFRH/BD/32965/2006 to PTM) and PDCT program (project PTDC/EIA/71587/2006). DR, RM, and HdJ are supported by the European Commission under project EC-MOAN (FP6-2005-NEST-PATH-COM/043235).

References

Batt, G.; Ropers, D.; de Jong, H.; Geiselmann, J.; Mateescu, R.; Page, M.; and Schneider, D. 2005. Analysis and verification of qualitative models of genetic regulatory networks: A model-checking approach. In Kaelbling, L., ed., *Proceedings of the Intl. Joint Conf. on Artif. Intel.*, 370–375.

Batt, G.; de Jong, H.; Page, M.; and Geiselmann, J. 2007. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica* 44(4):982–989.

Bellazzi, R.; R., G.; Ironi, L.; and Patrini, C. 2001. A hybrid input-output approach to model metabolic systems: an application to intracellular thiamine kinetics. *J. Biomed. Inform.* 34(4):221–48.

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

Cimatti, A.; Clarke, E.; Giunchiglia, E.; Giunchiglia, F.; Pistore, M.; Roveri, M.; Sebastiani, R.; and Tacchella, A. 2002. NuSMV 2: An opensource tool for symbolic model checking. In Brinksma, D., and Larsen, K., eds., *Proceedings of the 14th Intl. Conf. on Comp. Aided Verif.*, volume 2404 of *LNCS*, 359–64. Berlin: Springer-Verlag.

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

Dwyer, M.; Avrunin, G.; and Corbett, J. 1999. Patterns in property specifications for finite-state verification. In *Proceedings of the 21st Intl. Conf. on Soft. Eng.*, 411–20.

Garavel, H.; Lang, F.; and Mateescu, R. 2007. CADP 2006: A toolbox for the construction and analysis of distributed processes. In Damm, W., and Hermanns, H., eds., *Proceedings of the 19th Intl. Conf. on Comp. Aided Verif.*, volume 4590 of *LNCS*, 158–63. Berlin: Springer-Verlag.

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

Gutierrez-Ríos, R.; Freyre-Gonzalez, J.; Resendis, O.; Collado-Vides, J.; Saier, M.; and Gosset, G. 2007. Identification of regulatory network topological units coordinating the genome-wide transcriptional response to glucose in *Escherichia coli*. *BMC Microbiol.* 7(1):53.

Hengge-Aronis, R. 1996. Regulation of gene expression during entry into stationary phase. In F.C. Neidhardt, *et al*., ed., *Escherichia coli and Salmonella: Cellular and Molecular Biology*, 1497–512. Washington DC: ASM Press.

King, R.; Garrett, S.; and Coghill, G. 2005. On the use of qualitative reasoning to simulate and identify metabolic pathways. *Bioinformatics* 21(9):2017–26.

Klipp, E.; Nordlander, B.; Krüger, R.; Gennemark, P.; and Hohmann, S. 2005. Integrative model of the response of yeast to osmotic shock. *Nat. Biotechnol.* 23(8):975–82.

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

Kupferman, O.; Vardi, M.; and Wolper, P. 2000. An automata-theoretic approach to branching-time model checking. *J. ACM* 47(2):312–60.

Ropers, D.; de Jong, H.; Page, M.; Schneider, D.; and Geiselmann, J. 2006. Qualitative simulation of the carbon starvation response in *Escherichia coli*. *Biosystems* 84(2):124–52.

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

Szallazi, Z.; Periwal, V.; and Stelling, J. 2006. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. Cambridge, MA: MIT Press.

Thomas, R.; Thieffry, D.; and Kaufman, M. 1995. Dynamical behaviour of biological regulatory networks. *Bull. Math. Biol.* 57(2):247–276.