INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Temporal Logic Patterns for Querying Dynamic Models of Cellular Interaction Networks

Pedro T. Monteiro — Delphine Ropers — Radu Mateescu — Ana T. Freitas — Hidde de Jong

N° 6470

Fevrier 2008

Thème BIO

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6470--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche GRENOBLE - RHÔNE-ALPES

Temporal Logic Patterns for Querying Dynamic Models of Cellular Interaction Networks

Pedro T. Monteiro [*†‡], Delphine Ropers[†], Radu Mateescu[†], Ana T. Freitas[‡], Hidde de Jong[†]

Thème BIO — Systèmes biologiques
Équipes-Projets HELIX et VASY

Rapport de recherche n° 6470 — Fevrier 2008 — 20 pages

Abstract: Models of the dynamics of cellular interaction networks have become increasingly larger in recent years. Formal verification based on model checking provides a powerful technology to keep up with this increase in scale and complexity. 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 that capture recurring biological questions and that can be automatically translated into temporal logic. The applicability of the developed set of patterns has been investigated by the analysis of an extended model of the network of global regulators controlling the carbon starvation response in *Escherichia coli*.

Key-words: Formal verification, Genetic regulatory networks, Systems biology, Qualitative Simulation, Temporal logic

[*] Université Claude Bernard Lyon 1, 43 Bvd. du 11 Nov. 1918, 69622 Villeurbanne Cedex, France
[†] INRIA Grenoble - Rhône-Alpes, 655 Av. de l'Europe, Montbonnot, 38334 St. Ismier Cedex, France
[‡] IST/INESC-ID, 9 Rua Alves Redol, 1000-029 Lisbon, Portugal

Patterns en Logique Temporelle pour faire des requêtes à modèles dynamiques de réseaux d'interaction cellulaires

Résumé : Les modèles de la dynamique des réseaux d'interaction cellulaires sont devenus de plus en plus grands au cours des dernières années. La vérification formelle basée sur le model checking fournit une technologie puissante pour faire face à cette augmentation en taille et en complexité. Malheureusement l'application de telles approches est limitée par la difficulté pour l'utilisateur non expérimenté de formuler des questions appropriées en logique temporelle. Pour faire face à ce problème, nous proposons l'utilisation de patterns, des squelettes de requêtes à haut niveau qui expriment des questions biologiques récurrentes et qui peuvent être automatiquement traduites en logique temporelle. L'applicabilité de l'ensemble de patterns développés a été étudiée par l'analyse d'une extension du modèle du réseau des régulateurs qui contrôle la réponse au manque de carbone chez *Escherichia coli*.

Mots-clés : Vérification formelle, Réseaux de régulation génique, Biologie des systèmes, Simulation qualitative, Logique temporelle

1 Introduction

Models of the dynamics of cellular interaction networks have become increasingly larger in recent years. While whole-cell models are not on the horizon yet, complex networks underlying specific cellular processes have been modeled in detail, such as the osmotic shock response in yeast [22], the yeast cell cycle [9], and signalling pathways involved in cancer [28]. The study of these models by means of analysis and simulation tools leads to large amounts of predictions, typically time-courses of the concentrations of several dozens of molecular components in a variety of physiological conditions and genetic backgrounds. This raises the question how to make sense of these data, that is, how to obtain an understanding of the way in which particular molecular mechanisms control the cellular process under study, and how to identify interesting predictions of novel phenomena that can be confronted with experimental data.

Methods from the field of formal verification provide a promising way to deal with the analysis of large and complex models of cellular interaction networks [13]. Generally speaking, formal verification proceeds by specifying dynamical properties of interest as statements in temporal logic. Efficient so-called model-checking algorithms, implemented in publicly-available computer tools, exist to determine whether the statements are true or false, and thus whether the dynamic properties are satisfied by the model. The methods are generally applicable to discrete models of cellular interaction networks, or continuous models that have been discretized under a suitable abstraction criterion. Several examples exist of the application of model-checking approaches in systems biology (*e.g.*, [2, 4, 5, 7, 8, 14]).

Formal verification based on model checking provides a powerful technology to query models of cellular interaction networks. It raises a number of new issues though, notably that of formulating good questions when analyzing a huge network model. The problem of 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. For instance, the property "Gene g is eventually expressed, and necessarily preceded over the whole duration of the experiment by a concentration larger than 0.9 μM of the transcription factor P" corresponds to the following CTL formula, where exp_g denotes expression of g:

$$EF(exp_g) \wedge \neg E(True\ U\ (\neg([P] > 0.9\mu M) \wedge E(True\ U\ exp_g))) \qquad (1)$$

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 [12]. Apart from lists of example queries [8], the systematic definition of queries has not received any attention in systems biology thus far.

The aim of this paper is to develop a set of patterns for the analysis of dynamic models of cellular interaction networks. Its main contributions lie, first, in the development of generic query templates, based on a review of frequently-asked questions by modelers, and the translation of these templates into temporal logic formulas (Sec. 2). Second, we apply the patterns for analyzing the qualitative dynamics of a large and complex model of the *E. coli* carbon starvation response (Sec. 3). This model extends a previous model [27] by taking into account additional regulators of the bacterium, notably a module centered around the general stress response factor RpoS. We verify the control the latter is predicted to exert on the DNA supercoiling level in the cell.

2 Patterns of biological queries

2.1 Description of network dynamics

As a basic hypothesis, we assume that the dynamics of molecular interaction networks can be modeled by means of *finite state transition systems* (FSTSs) [11]. The latter formalism provides a general description of a dynamical system that implicitly or explicitly underlies many of the existing discrete formalisms used to model cellular interaction networks, such as Boolean networks and their generalizations, Petri nets, and process algebras. In addition, by defining appropriate discrete abstractions, continuous models of cellular interaction networks can also be mapped to FSTSs. The generality of the FSTS formalism is important for assuring the wide applicability of the patterns developed in this section. Moreover, statements in temporal logics are usually interpreted on FSTSs, so that the latter naturally connect network 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 cellular interaction network, each of which is characterized by a set of atomic propositions, such as that the concentration of protein P is increasing, or that the concentration of metabolite M is smaller than 5 mM.

2.2 Identification of patterns

The notion of patterns originates in architecture [1] and was introduced in the domain of software engineering as a means to capture expert solutions to recurring problems in program design [15]. In the formal verification domain they have been introduced in an influential paper by [12], 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 [29]), as well as previous applications of model checking and temporal logic (*e.g.*, [2, 4, 5, 7, 8, 14]). 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? [22]
- Once a cell has executed START, does it slip back into G1 phase and repeat START? Or rather, must it execute a FINISH to return to G1? [9].
- Does Shc phosphorylation exhibit a relative acceleration with decreasing EGF concentration and show a decline over time? [28]

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.

2.3 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

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

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. We will return to this generalization in Sec. 5.

It is often convenient to introduce predefined state descriptors that capture Boolean combinations of atomic propositions that are recurrently used. Some examples of predefined state descriptors that we found useful are the following:

- $Increases_i/Decreases_i$: the concentration of molecular component i increases/decreases in this state;
- *IsSteadyState*: the concentrations of all molecular components are steady in this state;
- *IsOscillatoryState*: the concentrations of some molecular components oscillate in this state.

Notice that the precise definition of the state descriptors depends on the particular type of FSTS that is used, as the latter determines the set of atomic propositions 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*. 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 high expression 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 protein P_1 and protein P_2 are highly expressed".

More generally, the *exclusion* pattern captures the *safety properties* used in the domain of concurrent systems. A *safety* property (of which *mutual exclusion* is a typical example) expresses that "something bad never happens" during the execution of the system, in our example a bad state violating mutual exclusion.

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 occurrence of the second state. If the latter state necessarily follows, then the consequence pattern expresses a form of causal relation. Instances of this pattern are, for instance, "If a state occurs in which protein P is phosphorylated, then it is possibly followed by a state in which the expression of gene g decreases", or "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 reached 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. Both the first and the second state, in that order, have to be observed 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 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. Instances of this pattern are "A state in which reactions R_1 and R_2 occur at a high rate is reached after 2 min, and is possibly preceded at some time by a state in which the transcription factor P is phosphorylated" or "A steady state is reached 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. Instances of the pattern are "A state in which reaction R occurs at a high rate can persist indefinitely" and "A state with a basal expression of gene g must persist indefinitely".

2.4 Translation to temporal logic

By defining a translation into temporal logic 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 [11].

Two examples of translations of the patterns in Sec. 2.3 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 ($\mathbf{E}\ p$) or every ($\mathbf{A}\ p$) 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 ($\mathbf{F}\ p$) or for every ($\mathbf{G}\ p$) 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: *fixed points*, the least (μ) and greatest (ν), and *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, a property p possibly ($\Diamond\ p$) or necessarily ($\Box\ p$) holds on some or all of its outgoing states.

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 reached 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 reached 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 reached 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 reached and is **necessarily** preceded **all the time** by a state ϕ	$EF\ (\psi)\ \wedge \neg E(T\ U$ $(\neg \phi \wedge E(T\ U\ \psi)))$	$\mu X.(\psi \vee \Diamond X)\ \wedge \neg \mu Y.(\neg \phi \wedge$ $\mu Z.(\psi \vee (T \wedge \Diamond Z)) \vee (T \wedge \Diamond Y))$
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 [23], which is interpreted on classical Kripke structures. The symbol T stands for True.

3 Carbon Starvation Response in *E. coli*

3.1 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 [21], and controlled on the molecular level by a complex genetic regulatory network integrating various environmental signals.

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 [19, 20]. However, notwithstanding the enormous amount of information accumulated on the genes, proteins, and other molecules known to be involved in the stress adaptation process, it is currently not understood how the response of the cell emerges from the network of molecular interactions. Moreover, with some exceptions [6], numerical values for the kinetic parameters and the molecular concentrations are absent, 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 network using a class of *piecewise-linear (PL) differential equations*. The PL models, originally introduced by [18], 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 [4, 3]. 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 ([27]; 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.

3.2 Qualitative simulation of carbon starvation response

The mathematical properties of the class of PL models used for modeling the stress response network have been well-studied [18]. We have previously shown how discrete abstractions can be used to convert the continuous dynamics of the PL systems into a FSTS [3]. The states S of the FSTS correspond to hyperrectangular regions in the concentration space, while the transitions T arise from trajectories that enter 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 Genetic Network Analyzer (GNA) [4]. GNA is able to export the FSTS to standard model checkers like NuSMV [10] and CADP [16].

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 purpose, *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.

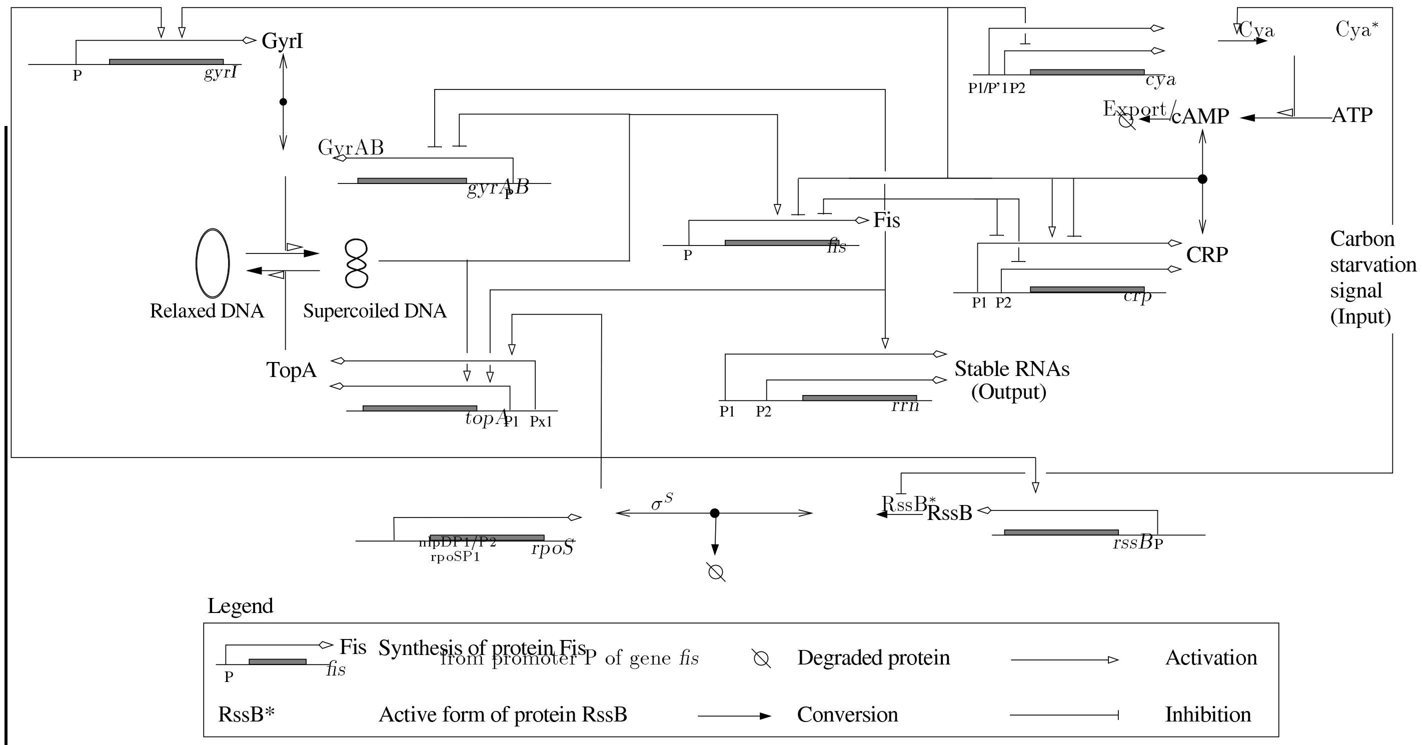
In the next section we show how the patterns defined in Sec. 2.3 can speed up the querying of these FSTSs, by simplifying the formulation of relevant properties to be tested. We are particularly interested in the question how the extension of the model with RpoS affects the predicted dynamics of the system. The instances of the patterns were translated into the temporal logic CTL following the translation rules of Table 1, and then verified using the model-checker NuSMV.

4 Analysis of Carbon Starvation Response Model using query patterns

4.1 Mutual inhibition of Fis and CRP

The proteins Fis and CRP mutually inhibit their expression (Fig. 1). The regulatory protein CRP is the target of a signal-transduction pathway, which activates the adenylate cyclase Cya in case of carbon starvation. In turn, the latter synthesizes a small molecule, cAMP, which binds to CRP. This active form of CRP is able to regulate the expression of a large number of genes. In particular, CRP·cAMP binds to the promoter region of the gene *fis*, thereby preventing synthesis of new Fis proteins. Fis is an important regulator of genes involved in the cellular metabolism but it also inhibits *crp* expression, by binding to multiple sites in the two promoter regions of the gene, P1 and P2. The regulatory interactions between genes *fis* and *crp* form a positive feedback loop, a motif often found in the genetic regulatory networks. When present in isolation, this kind of motif has been shown to lead to bistability [17].

The question can be asked whether the motif is also functional in the context of the carbon starvation response network. For instance, the expression of *fis* is not only controlled by CRP·cAMP, but also by the DNA supercoiling level and Fis itself. To check whether the bistability property is preserved in the larger network, we used two instances of the *occurrence/exclusion* pattern to express that it is impossible that proteins Fis and CRP be simultaneously present at high and at low concentration in the cells (Table 2). This property was shown to be true by the NuSMV model checker. We conclude that the positive feedback loop involving *fis* and *crp* is functional.



(a)

$$\dot{x}_{gyrAB} = \kappa_{gyrAB}\ (1 - s^{+}(x_{gyrAB}, \theta^2_{gyrAB})\ s^{-}(x_{gyrI}, \theta^1_{gyrI})\ s^{-}(x_{topA}, \theta^1_{topA}))\ s^{-}(x_{fis}, \theta^4_{fis}) - \gamma_{gyrAB}\ x_{gyrAB}$$
$$0 < \theta^1_{gyrAB} < \theta^2_{gyrAB} < \kappa_{gyrAB}/\gamma_{gyrAB} < max_{gyrAB}$$

(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}.

4.2 Damped oscillations after carbon upshift

The carbon starvation response network also contains a negative feedback loop, involving the genes *gyrAB*, *topA*, and *fis* (Fig. 1). GyrAB is a gyrase protein which supercoils the DNA structure, whereas the topoisomerase TopA relaxes it. An increase of the DNA supercoiling level stimulates expression of Fis, which in turn decreases the supercoiling level, by stimulating *topA* expression and inhibiting *gyrAB* expression. The resulting negative feedback loop was predicted to give rise to (damped) oscillations of Fis and GyrAB concentrations after a carbon upshift [27].

In the present version of our model, additional interactions contribute to controlling the DNA supercoiling level. Hence, the gyrase inhibitor GyrI represses the activity of GyrAB by forming a complex with the protein. The expression of *gyrI* is notably stimulated by RpoS [25]. We formulated a *consequence* pattern to verify whether this affects the functioning of the negative feedback loop. In particular, we checked whether the carbon upshift is still a necessary condition for the occurrence of damped oscillations, as it was in the previous model (Table 2). In the pattern we made use of the state descriptor *isOscillatoryState*, which labels states as belonging to a (terminal) cycle in the FSTS. The occurrence of an oscillatory state could alternatively be expressed using temporal logic formulas (Sec. 5). The model-checker returned true for the query, meaning that the damped oscillations still occur following a carbon upshift.

4.3 Control of entry into stationary phase by RpoS

RpoS is a stress sigma factor that allows cells to adapt to and survive under harmful conditions by entering stationary phase [20]. 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.

4.4 Expression of *topA* during growth-phase transitions

Our previous model was incapable of accounting for the control of DNA supercoiling during growth-phase transitions. In particular, TopA was predicted to be never expressed, which is consistent with published data [26]. The extension of the model with RpoS makes it possible to refine the description of the control of the DNA supercoiling level. On the one hand, GyrAB activity is regulated by GyrI, as mentioned previously, and on the other hand, the *topA* promoter is activated by RpoS.

In order to know whether *topA* is expressed in response to the carbon source availability, we used an *invariance* pattern to check if the absence of *topA* expression persists indefinitely (Table 2). The corresponding temporal logic formula is false and the diagnostic of the model-checker shows that expression of *topA* is stimulated at the entry into stationary phase, most likely under the influence of RpoS. Indeed, following carbon starvation, the protein RssB is inactivated, which leads to the accumulation of RpoS at high levels. RpoS in turn stimulates the expression of *topA*.

Occurrence/exclusion pattern: Mutual inhibition of Fis and CRP
| 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 | | 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
| 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
| A state | $x_{rpoS} \geq \theta^1_{rpoS}$ | is reached and is | necessarily | preceded | at some time | by a state | $x_{rrn} > \theta_{rrn}$ |
CTL: $EF\ (x_{rpoS} \geq \theta^1_{rpoS}) \wedge \neg E\ (\neg(x_{rrn} > \theta_{rrn})\ U\ (x_{rpoS} \geq \theta^1_{rpoS}))$ μ-calculus: $\mu X.((x_{rpoS} \geq \theta^1_{rpoS}) \vee \Diamond X) \wedge \neg\mu Y.((x_{rpoS} \geq \theta^1_{rpoS}) \vee (\neg(x_{rrn} > \theta_{rrn}) \wedge \Diamond Y))$
Invariance pattern: Expression of *topA* during growth-phase transitions
| 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 the instances of the patterns used in the analysis of the *E. coli* carbon starvation response into CTL and μ-calculus, following the translation rules in Table 1.

5 Discussion

Formal verification techniques are promising tools for upscaling the analysis of cellular interaction networks. 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 ([12], see also [24]), 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. The patterns capture a large number of frequently-asked questions by modelers in systems biology, but they are not restricted to a particular type of network or a particular biological system. In addition, we have provided translations of the patterns to two different temporal logics, CTL and μ-calculus.

The second contribution of the paper is the use of these patterns for the analysis of the complex genetic regulatory network involved in the carbon starvation response in *E. coli*. We have modeled this network by means of PL differential equations and simulated the qualitative dynamics of the system in response to carbon starvation and carbon upshift. Our model extends a previous model [27] with additional global regulators, notably the sigma factor RpoS, to better account for the control of DNA supercoiling during the growth transitions of the bacteria. The patterns are instantiated to verify the effect of this addition to the predicted network dynamics.

The patterns proposed in this paper are globally consistent with those discussed in [12], but there are differences due to the specific application domain for which our patterns were developed. For instance, the notion of scope used by [12] is not commonly defined for all the patterns, but implicitly present through the use of specific variants for each pattern. Also, we have not explicitly included patterns that can be obtained by the recursive application of other patterns, such as the *chain response* pattern defined in [12]. While patterns have not been used for the querying of cellular interaction networks thus far, some papers list example questions. It is reassuring to observe that all questions in the list of [8] can be expressed by means of the patterns in Sec. 2.3.

An obvious generalization of the patterns proposed in this paper, already briefly mentioned in Sec. 2, would be to allow state descriptors that are formulas in temporal logic. For instance, instead of using atomic propositions to label states belonging to a (terminal) cycle in the FSTS, which requires the preliminary detection of strongly connected components in the state transition graph, we could use temporal logic formulas [4]. The introduction of temporal logic formulas as state descriptors makes the patterns more general, but also potentially more complicated to formulate and dependent on a particular temporal logic. A compromise trading some expressive power for user-friendliness would be to restrict the possible temporal logic formulas to simple forms and introduce these as predefined state descriptors (Sec. 2.3). This is consistent with the main idea underlying the use of patterns, namely that they cannot be expected to cover all possible queries, but rather should allow users to formulate their frequent questions without worrying about the translation to temporal logic.

References

[1] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language*. Oxford University Press, New York, NY, 1977.

[2] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. *Cell Biochem. Biophys.*, 38(3):271–86, 2003.

[3] G. Batt, H. de Jong, M. Page, and J. Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, in press, 2007.

[4] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics*, 21(Suppl 1):i19–i28, 2005.

[5] G. Bernot, J.-P. Comet, A. Richard, and Guespin J. Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. *J. Theor. Biol.*, 229(3):339–48, 2004.

[6] K. Bettenbrock, S. Fischer, A. Kremling, K. Jahreis, T. Sauter, and E.-D. Gilles. A quantitative approach to catabolite repression in *Escherichia coli*. *J. Biol. Chem.*, 281(5):2578–84, 2006.

[7] M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using the PRISM model checker. In G. Plotkin, editor, *Proc. of CMSB*, pages 179–90, 2005.

[8] 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.

[9] K.C. Chen, L. Calzone, A. Csikasz-Nagy, F.R. Cross, B. Novak, and J.J. Tyson. Integrative analysis of cell cycle control in budding yeast. *Mol. Biol. Cell*, 15(8):3841–62, 2004.

[10] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In D. Brinksma and K.G. Larsen, editors, *Proc. 14th CAV*, volume 2404 of *LNCS*, pages 359–64, Berlin, 2002. Springer-Verlag.

[11] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, Cambridge, MA, 1999.

[12] M.B. Dwyer, G.S. Avrunin, and J.C. Corbett. Patterns in property specifications for finite-state verification. In *Proc. 21st Intl. Conf. Software Engineering*, pages 411–20, 1999.

[13] J. Fisher and T.A. Henzinger. Executable cell biology. *Nat. Biotechnol.*, 25(11):1239–50, 2007.

[14] J. Fisher, N. Piterman, A. Hajnal, and T.A. Henzinger. Predictive modeling of signaling crosstalk during *C. elegans* vulval development. *PLoS Comput. Biol.*, 3(5):e92, 2007.

[15] E. Gamma, R. Helm, R. Johnson, and J.M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, New York, NY, 1994.

[16] H. Garavel, F. Lang, and R. Mateescu. CADP 2006: A toolbox for the construction and analysis of distributed processes. In W. Damm and H. Hermanns, editors, *Proc. 19th CAV*, volume 4590 of *LNCS*, pages 158–63, Berlin, 2007. Springer-Verlag.

[17] T.S. Gardner, C.R. Cantor, and J.J. Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403(6767):339–42, 2000.

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

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

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

[21] G.W. Huisman, D.A. Siegele, M.M. Zambrano, and R. Kolter. Morphological and physiological changes during stationary phase. In F.C. Neidhardt, editor, *Escherichia coli and Salmonella: Cellular and Molecular Biology*, pages 1672–82, Washington DC, 1996. ASM Press.

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

[23] O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–60, 2000.

[24] Z. Manna and A. Pnueli. Tools and rules for the practicing verifier. In R. Rashid, editor, *Carnegie Mellon Computer Science: A 25th Anniversary Commemorative*, pages 125–159, New York, NY, 1991. ACM Press.

[25] T.J. Oh, I.L. Jung, and I.G. Kim. The *Escherichia coli* SOS gene *sbmC* is regulated by H-NS and RpoS during the SOS induction and stationary growth phase. *Biochem. Biophys. Res. Commun.*, 288(4):1052–8, 2001.

[26] H. Qi, R. Menzel, and Y.C. Tse-Dinh. Regulation of *Escherichia coli topA* gene transcription: involvement of a *sigmaS*-dependent promoter. *J. Mol. Biol.*, 267(3):481–9, 1997.

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

[28] B. Schoeberl, C. Eichler-Jonsson, E.-D. Gilles, and G. Müller. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat. Biotechnol.*, 20(4):370–75, 2002.

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

Contents

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399