From Simple Regulatory Motifs to Parallel Model Checking of Complex Transcriptional Networks

J. Barnat, L. Brim, I. Černá, S. Dražan, and D. Šafránek [1,2]

Faculty of Informatics, Masaryk University, Czech Republic

Abstract

In recent years, the previously reductionistic style of biological research has turned firmly towards systematic integrative paradigm, the so-called systems biology. In this novel paradigm, a functionality of a living cell is understood as a large set of complex biochemical reactions of several kinds running in parallel at different time-scales. The central mechanism which drives every living cell is protein synthesis, the so-called transcription, which is realized according to the genetic code. There are complex regulatory interactions that control transcription of genes to proteins. Owing to their inherent complexity, analysis of dynamical models of such interactions requires a scalable computational approach. In this paper we employ parallel LTL model checking for a case study of selected dynamic properties of an in sillico model of transcription in Bacillus subtilis, a bacterium living in soil. Moreover, we show the general fact that crucial LTL properties characterizing transcriptional dynamics can be inferred from network motifs commonly studied in systems biology.

Key words: tanscriptional regulatory networks, discrete simulation, parallel LTL model checking, Bacillus subtilis

1 Introduction

The traditional reductionistic style of biological research is nowadays turning towards systematic integrative paradigm, the so-called *systems biology*. There appear plenty of databases of biological knowledge keeping both structural and functional aspects of living organisms. Existence of such databases allows building of *in silico models* that predict functionality of living cells.

Each function of any living cell is driven by proteins. Proteins are synthesized according to the genetic code in the process of *transcription*. Transcription of genes to proteins is controlled by complex regulatory interactions.

[1] Email: safranek@fi.muni.cz

[2] This work has been supported by the FP6 project No. NEST-043235 (EC-MOAN).

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

These interactions are driven by specific proteins, so-called *transcription factors*, which collaborate on activation (increase) or repression (decrease) of particular gene transcription. For each cell type the transcription regulation is described by a *transcriptional regulatory network* (TRN) in which nodes represent proteins and genes interconnected by edges denote their mutual interaction. TRNs make the bottom level of complex biological networks and pathways. Even when taken separately, TRNs of most of pathways can be themselves very large (having around 10s of genes). A scheme of a TRN is depicted in Figure 1. There is a network of 8 genes from which m proteins (each denoted X_i) is produced. Additionally, the process of regulation is controlled by n external signals which determine interface for interaction with other functional layers in a cell.

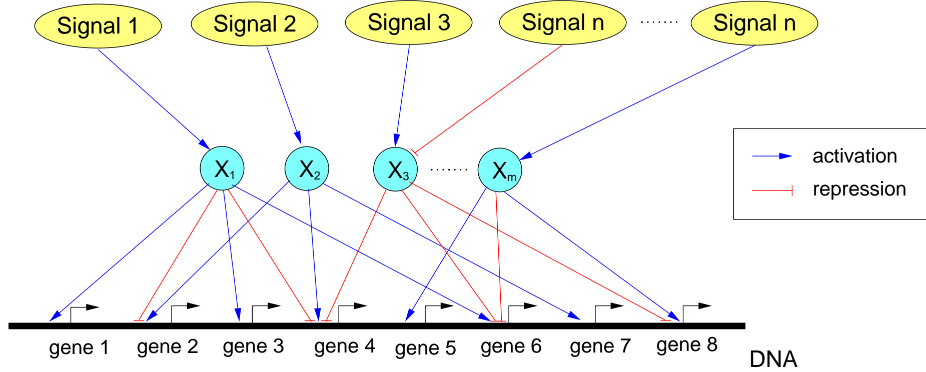

Fig. 1. Scheme of a transcriptional regulatory network

In order to deal with the complexity of transcriptional regulatory networks, experimental methods have to be supplemented with mathematical modelling and computer-supported analysis. One of the most critical limitations in applying current approaches to modelling and analysis is their pure scalability. Large models require powerful computational methods, the hardware infrastructure is available (clusters, GRID, multi-core computers), but the parallel (distributed) algorithms for model analysis are still under development.

The most widely-used modelling frameworks for the analysis of the dynamics of TRNs are based on ordinary differential equations [30] (ODE). The reduction of continuous models to discrete automata by a sequence of approximations and abstractions allows formal methods for the automated verification of properties of discrete transition systems to be applied [9]. One of methods which can be employed here is model checking. At the continuous level, interesting properties of TRNs can be characterized by certain parts of the interaction networks – so-called *network motifs*. Role of network motifs in dynamic behaviour can be mathematically analyzed [**?**]. However, such mathematical analysis is very complicated and cannot be done automatically. Therefore, model checking appears to be a suitable tool for algorithmical analysis of ODE models, in particular, their discrete abstractions. Properties which can be analysed by model checking include the behaviour specific for

known network motifs. Recent studies on biologically-relevant properties identified the need for both branching-time temporal operators, able to express multi-stability properties (reachability of several different equilibrium states) ([?,?]), and linear-time temporal operators, able to capture the oscillations of protein concentrations ([?,?]) as well as temporal ordering motifs ([?]).

When dealing with large TRNs, standard model-checking techniques do not provide acceptable response times for answering user queries and parallel model-checking algorithms are required. Owing to dynamical dependences among state variables, the state-space explosion arises during reduction to discrete automata. Even relatively small ODE models containing around 15 state variables lead to large automata having hundreds of thousands states. However, while substantial work on model-checking qualitative as well as quantitative properties of biochemical networks has been already achieved, to our best knowledge, no attempts to use parallel model checkers to analyse complex networks are known.

1.1 Our Contribution

In this paper we present a case study of applying parallel model checking to analysis of transcriptional regulatory networks using the extension GeNeSim of the parallel model-checker DiVinE [1]. We employ the piece-wise affine discrete abstraction method in which differential equations of the original ODE model are reduced to a system of piece-wise linear differantial equations (PLDE) as proposed in [19]. Our distributed state space generator is based on the respective qualitative simulation method implemented (as a sequential algorithm) in *Genetic Network Analyser (GNA)* [6].

The DiVinE distributed state space generator allows on-the-fly generation of the transition graph giving thus in many circumstances the possibility to analyse properties of even larger networks as opposed to the explicit representation as used in GNA simulator. This allows to check biologically interesting liveness properties on larger models than is possible with traditional sequential approach. In this case study we demonstrate that parallel model checking can extend the possibilities of qualititative analysis of Bacillus Subtilis bacteria presented in [16].

From our experiences on collaboration with biologists we have found it difficult to fully take advantage of the strong power of temporal logic. It is mainly because of the fact that thinking of biologists, based on experiments, principally differs from that of computer scientists. At present, it is far from reality that a temporal logic formalism can be directly used by biologists. To this end, we try to reveal biologically interesting LTL properties from the notion of network motifs which is well known in the biological community.

1.2 Related Work

The use of model checking for the analysis of biological networks has attracted much attention [9,32]. The individual approaches differ in models and model-checking tools used. Our approach is based on qualitative hybrid models as proposed by [19] and implemented in the GNA [17]. Besides GNA there are some other sequential approaches for model-checking of ODE models. The BIOCHAM workbench [12] provides an interface to symbolic model checker NuSMV and the enumerative CADP verification toolbox; the interface is based on a simple language for representing biochemical networks. The workbench provides mechanisms to reason about reachability, existence of partially described stable states, and some types of temporal behaviour. Another tool is the Robust Verification of Gene Networks (RoVerGeNe) [2]. To the best of our knowledge, none of the tools mentioned above employ the parallel approach.

This work extends our previous work [**?**] in two directions. At first, we introduce a pool of biologically relevant LTL properties which are systematically derived from transcriptional motifs which frequently appear in biological databases [**?**,**?**]. At second, the experiments of the Bacillus Subtilis case study presented in this paper are aimed at demonstrating how the motif-specific properties can be checked on complex networks that combine several motifs together.

2 Preliminaries

Before we start with stating the properties of individual transcriptional motifs, we briefly explain basic principles of PLDE systems and their qualitative simulation. The precise definition of these principles is given in [19]. Finally, we also give a syntax of the LTL logic used in this paper.

2.1 PLDE Systems

Assume we fix a network with n proteins. Let the concentration of ith protein in a time instant t, where $i \in \{1, ..., n\}$, be denoted by the variable $x_i(t)$. The respective PLDE system consists of a set of n equations. For each protein there is an equation describing how its concentration changes in time:

$$\frac{dx_i}{dt} = \sum_{l \in L} \kappa_{il} \varrho_{il}(\langle x_1, ..., x_n \rangle) - \gamma_i x_i$$

where

- L is a finite index set.
- For each $i \in \{1, .., n\}, l \in L$ κ_{il} is a constant expressing the rate of protein production.
- $\varrho_{il} : \mathcal{R}^n \to \{0, 1\}$ is a discrete input function.

- γ_i is a constant expressing the rate of exponential decay of protein i.

In general, each equation of the system consists of two terms – the positive production term and the negative degradation term. The latter term is naturally nonzero and describes instability of proteins among other species in the cell. The former term describes the transcriptional regulation, in particular, the intensity of protein production w.r.t. the current conditions in the cell. More precisely, the maximal production is given by the production constant κ and it is additionally regulated by the input function ϱ.

The input function ϱ in general depends on the current concentration of all the proteins in the system. In piece-wise linear approximation the (multidimensional) input function has a discrete range and is given by a product of elementary (one dimensional) step functions. These step functions qualitatively characterize edges in TRNs, in particular, the transcriptional activation or repression of the target gene by a certain transcriptional factor w.r.t. its given threshold concentration. The input function has the form:

$$\varrho_{il}(\langle x_1, ..., x_n \rangle) = \prod_{j=1}^{n} s^*(x_j, \theta_j^i)$$

where $s^* : \mathcal{R} \rightarrow \{0, 1\}$, $* \in \{+, -\}$, denotes a so-called step function defined for the protein concentration x_j and its threshold θ_j^i by the expression:

$$\text{activation}: \quad s^+(x_j, \theta_j^i) = \begin{cases} 1, \text{ if } x_j > \theta_j^i, \\ 0, \text{ if } x_j < \theta_j^i, \end{cases}$$

$$\text{repression}: \quad s^-(x_j, \theta_j^i) = 1 - s^+(x_j, \theta_j^i)$$

2.2 Qualitative Simulation

For given initial conditions (initial concentrations of protein species), the possible evolution of protein concentrations in time can be predicted by simulation. The algorithm of qualitative simulation for PLDE models relies on the fact that input functions given as products of step functions have discrete ranges. Up-to this approximation, the concentration values of proteins are discretely abstracted into several open intervals between respective thresholds, and additionally, the discrete points equal just to the threshold values.

For each variable there is a finite number of regulation configurations determined by particular values of the step functions. In each such configuration the transcriptional regulation tends towards an equilibrium state (a so-called *local equilibrium*) determined for some $L' \subseteq L$ by the equation:

$$\frac{dx_i}{dt} = 0 \Leftrightarrow x_i = \frac{\sum_{l \in L'} \kappa_{il}}{\gamma_i}$$

If we consider a variable x_i then for each particular configuration of the binary values returned by step functions appearing in the respective equation the equilibrium concentration value is given just by the expression above. In order to simulate the dynamics of the PLDE system, we set for each variable the inequality condition which mutually relates all the thresholds defined for that variable, and moreover, we also put each possible equilibrium concentration value between some two succseeding thresholds.

Evolution of protein concentrations starting from the given initial values is then given by the state transition system in which each state is characterized by the vector of current discrete concentration levels of the proteins (open intervals or just the thresholds). Because each protein is naturally degraded (γ_i is always nonzero) values of each variable are bounded by some maximal level and hence the system has finite number of states [19].

The outgoing transitions are computed differently for states where all variables are set to values approximated by an open interval (so-called regulatory states) and for states where at least one variable has a value equal to some threshold (so-called switching states). Each transition means visiting some of the succeeding states in the global phase space. In the former case, the transitions are determined directly by the direction vector leading from the current state to its respective local equilibrium state. In the latter case, the direction vector is computed by an intricated algorithm which relies on analysis of the system dynamics in neighbouring states. In general, time complexity of computing outgoing transitions for a given state is exponential in number of variables in the system. This is an imposition due to the accuracy of the overapproximative simulation and also a motivation for employing the parallel model checking method for analysis of PLDE systems.

3 PLDE and LTL Approach to Transcriptional Motifs

In this section, we characterize the basic transcriptional motifs modeled as PLDE systems. For each motif we state a set of LTL properties which describe qualitatively its significant dynamical properties. We focus on motifs that embody qualitatively exceptional properties, in particular, we do not include motifs that manipulate timing aspects of transcriptional interactions (e.g., single input modules, feedforward loops,... [?]).

3.1 Autoregulation Motifs

Autoregulation is the most simplest motif which appears very frequently in all TRNs [?]. It deals with one variable (one protein) which regulates its production from its own gene. Such a regulation determines the resulting amount of the protein concentration which is achieved when the system reaches a global equilibrium state. In particular, the regulation is controlled by a step function which depending on the current amount of the protein concentration

returns its increment. Depending on the character of the step function there are two types of autoregulation – negative and positive.

3.1.1 *Negative Autoregulation*

In the case of negative autoregulation, the input function is given by a single step function denoting repression. In particular, this means that the protein represses production of its own. The PLDE system consists of a single equation having the following form:

$$\frac{dx}{dt} = \kappa s^-(x,\theta) - \gamma x$$

The only local equilibrium state of this system is characterised by the concentration value $\frac{\kappa}{\gamma}$. By positioning this value above or below the threshold, we get different dynamic behaviour. In Table 1 there is given a characteristic LTL property for each of these positions.

The first line shows the fact that the production rate is relatively low and hence the autoregulation does not affect the transcription at the concentration level below the threshold, whereas for the values above the threshold the autoregulation switches off the production. Both aspects lead to the observation that this autoregulation motif forces the resulting global equilibrium state to appear at levels below the threshold.

$0 < \frac{\kappa}{\gamma} < \theta$	$\mathbf{FG}(x \leq \theta)$
$\theta < \frac{\kappa}{\gamma} < max$	$\mathbf{FG}(x = \theta)$

Table 1
Characteristic Properties of Negative Autoregulation

The second line represents the situation when the local equilibrium concentration is set to a higher value than the threshold. In this case, the autoregulation affects all concentration levels and leads the system to the global equilibrium in which the concentration value is equal just to the threshold level.

3.1.2 *Positive Autoregulation*

Positive autoregulation is determined by a single activation step function. Here the protein at certain concentration level activates its own production. The respective PLDE system is analogous to the previous one:

$$\frac{dx}{dt} = \kappa s^+(x,\theta) - \gamma x$$

Similarly to the case of negative autoregulation there are two different local equilibrium positions. They are listed in Table 2. In the first case, the behaviour is just the same as in the previous case. However, the case when the local equilibrium concentration is set above the threshold behaves

differently. In particular, a kind of bistability arises. In other words, there are two different global equilibrium states to which the system can lead. In such a setting, the positive autoregulation behaves like a switch that desides whether the system stabilizes at the maximal concentration level or at the zero level. Both situations are possible. The LTL property (1) describes existence of the two global equilibrium states — a necessary and sufficient condition of bistability. In PLDE approximation, both states are reachable from a single specific qualitative state – a so-called bistable switch. However, existence of such a state cannot be directly expressed in LTL. Therefore we state at the least a formula (2) which contradicts the existence of a bistable switch.

$0 < \frac{\kappa}{\gamma} < \theta$	$\mathbf{FG}(x \leq \theta)$
$\theta < \frac{\kappa}{\gamma} < max$	1. $((x < \theta) \Rightarrow \mathbf{G}(x < \theta)) \wedge ((x > \theta) \Rightarrow \mathbf{G}(x > \theta))$
	2. $(x = \theta) \Rightarrow (\mathbf{GF}(x > \theta) \wedge \mathbf{GF}(x < \theta))$

Table 2
Characteristic Properties of Positive Autoregulation

3.2 Feedback Loop Motifs

Another significant cathegory of transcriptional motifs is represented by a group of at least two different transcriptional factors which mutually interact in a circular manner. In particular, the respective network is a cyclic graph. Such motifs, depending on the type of individual interactions in the circle, present behavior similar to the autoregulation (imposing stability or multi-stability), or they can lead to concentration oscilations [?] (e.g., the well-known cyrcadian rhythm).

In this section we state the properties of the most significant representants of feedback loops, in particular, minimal loops made of just two proteins.

3.2.1 Double-negative feedback loop

In this kind of a feedback loop each of the two proteins represses the other one. If we denote the proteins X and Y, respectively, the relevant PLDE system is defined by two equations of the following form:

$$\frac{dx}{dt} = \kappa_x s^-(y, \theta_y^x) - \gamma_x x \qquad \frac{dy}{dt} = \kappa_y s^-(x, \theta_x^y) - \gamma_y y$$

$\theta_x^y < \frac{\kappa_x}{\gamma_x} < max_x$	1. $((x > \theta_x^y \wedge y < \theta_y^x) \Rightarrow \mathbf{G}(x > \theta_x^y \wedge y < \theta_y^x)) \wedge ((x < \theta_x^y \wedge y > \theta_y^x) \Rightarrow \mathbf{G}(x < \theta_x^y \wedge y > \theta_y^x))$
$\theta_y^x < \frac{\kappa_y}{\gamma_y} < max_y$	2. $(x = \theta_x^y \wedge y = \theta_y^x) \Rightarrow ((\mathbf{GF}(x > \theta_x^y) \wedge \mathbf{GF}(x < \theta_x^y)) \vee (\mathbf{GF}(y > \theta_y^x) \wedge \mathbf{GF}(y < \theta_x^y)))$

Table 3
Characteristic properties of double-negative feedback loop

From the possible 4 different local equilibria configurations we consider the most representative one, given in Table 3. In such a setting, the system is

bistable, and moreover, there is a bistable switch in the state characterised by the property that concentration value of each protein is equal to the respective threshold. The formula (1) states the property that two different global equilibrium states exist in the system. Similarly as in the case of bistability in positive autoregulation, the property of bistable switch is not expressible in LTL. However, we state a formula (2) which involves just counterexamples of bistable switch.

3.2.2 Double-positive feedback loop

This kind of two-protein feedback loop has the similar properties as the negative loop. The PLDE system differs only in signs of step functions used:

$$\frac{dx}{dt} = \kappa_x s^+(y, \theta_y^x) - \gamma_x x \qquad \frac{dy}{dt} = \kappa_y s^+(x, \theta_x^y) - \gamma_y y$$

The most representative configuration and the respective bistability properties are stated in Table 4. The property (1) differs from the respective property of the negative loop only in atomic propositions. The bistable switch contradiction (2) is just the same formula.

$\theta_x^y < \frac{\kappa_x}{\gamma_x} < max_x$	1. $((x > \theta_x^y \wedge y > \theta_y^x) \Rightarrow \mathbf{G}(x > \theta_x^y \wedge y > \theta_y^x)) \wedge ((x < \theta_x^y \wedge y < \theta_y^x) \Rightarrow \mathbf{G}(x < \theta_x^y \wedge y < \theta_y^x))$
$\theta_y^x < \frac{\kappa_y}{\gamma_y} < max_y$	2. $(x = \theta_x^y \wedge y = \theta_y^x) \Rightarrow ((\mathbf{GF}(x > \theta_x^y) \wedge \mathbf{GF}(x < \theta_x^y)) \vee (\mathbf{GF}(y > \theta_y^x) \wedge \mathbf{GF}(y < \theta_x^y)))$

Table 4
Characteristic properties of double-positive feedback loop

3.2.3 Incoherent feedback loop

The most interesting two-protein feedback motif is based on circular interaction in which the first of the two proteins acts as a repressor of the second protein production while the second protein acts as an activator of the first protein (or vice-versa). Formally, the corresponding PLDE system for such a situation can be of the following form:

$$\frac{dx}{dt} = \kappa_x s^-(y, \theta_y^x) - \gamma_x x \qquad \frac{dy}{dt} = \kappa_y s^+(x, \theta_x^y) - \gamma_y y$$

The representative configuration of local equilibria is determined by the same inequalities as in the case of the previous loop motifs. In such a situation, the motif leads to oscillatory behaviour (when initial concentrations are suitably set). The oscillation means that each of both species periodically changes its concentration value from the level below the threshold to the level above the threshold and vice-versa. Moreover, the oscillatory behavior is in this case kept on for the entire time-live of the cell. LTL properties which guarantee the oscillatory behaviour for each of the proteins are given in Table 5 as properties (1) and (2).

$\theta_x^y < \frac{\kappa_x}{\gamma_x} < max_x$	1. $(\mathbf{G}((x_a \leq \theta_a^b) \Rightarrow \mathbf{F}(x_a > \theta_a^b))) \wedge (\mathbf{G}((x_a \geq \theta_a^b) \Rightarrow \mathbf{F}(x_a < \theta_a^b)))$
$\theta_y^x < \frac{\kappa_y}{\gamma_y} < max_y$	2. $(\mathbf{G}((x_b \leq \theta_b^a) \Rightarrow \mathbf{F}(x_b > \theta_b^a))) \wedge (\mathbf{G}((x_b \geq \theta_b^a) \Rightarrow \mathbf{F}(x_b < \theta_b^a)))$

Table 5
Characteristic properties of incoherent feedback loop

4 GeNeSim: Parallel Model Checker for PLDE Models

GeNeSim is build on the top of the DiVinE library that offers common functions needed to develop a parallel or distributed enumerative model checker. The only extension to the library that was necessary, was the extension of the state generator to a state generator tailored for the specific input provided by GeNeSim GUI [?]. For the structure of GeNeSim implementation and connection to DiVinE see Figure 2.

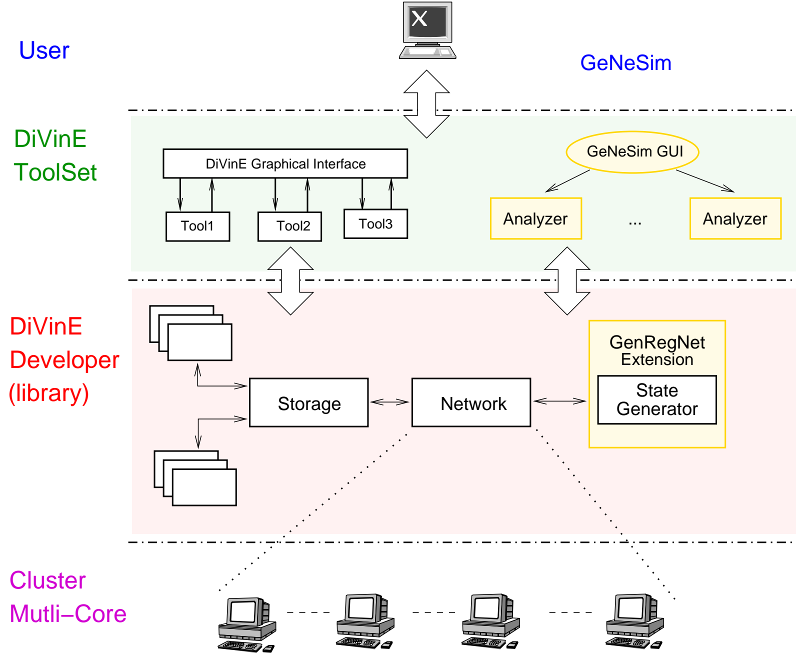

Fig. 2. How is GeNeSim embodied into DiVinE.

4.1 PLDE model representation and state generator

The central component of GeNeSim is the state generator. The hierarchy of DiVinE state space representation classes is extended with new structures that represent symbolically the PLDE model in C++. A PLDE model (Genesim System) is represented as a container of variables which occur in respective PLDEs. Each state variable contains a set of production rates (may be empty) and a set of degradation rates. With respect to the mathematical specification of PLDEs, at least one degradation rate constant must be always defined. Each rate constant is defined as a container of *regulation terms*. A regulation term represents a particular subterm of the equation which is relevant to

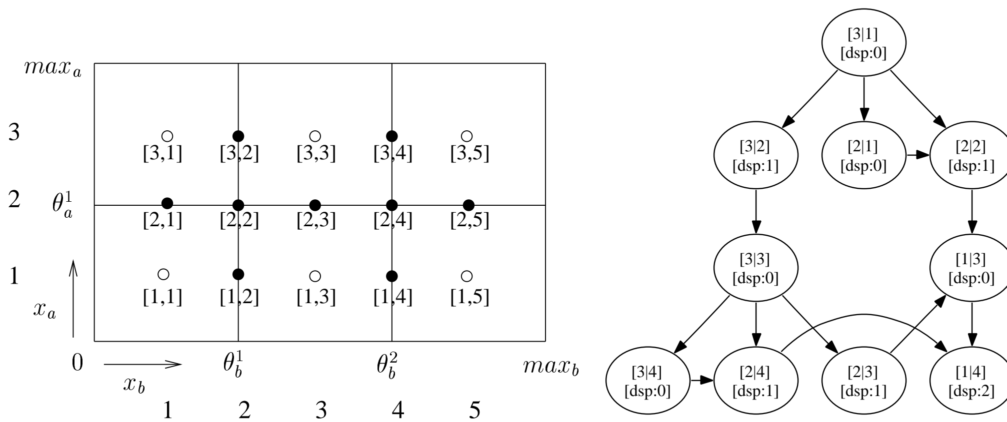

Fig. 3. A qualitative state space and a transition system generated by GeNeSim

the activation and deactivation of the respective reaction (represented by the relevant production/degradation rate constant). We support two forms of regulation terms – negative and positive. The positive term is defined as a direct product of step functions whereas the negative term is defined as a negation of a product of step functions. Both kinds of regulation terms allow all possible types of PLDE models to be encoded in GeNeSim. Negative terms are necessary for modeling of transcription factors which have the form of multi-protein complexes, as is showed in [19].

The GeNeSim state generator implements the DiVinE methods `get_initial()` for determining the initial state of the system and `get_succs()` that computes a container of successors of a given state. By these two methods the implicit representation of the qualitative simulation is implemented. Such a representation allows integration of GeNeSim with DiVinE distributed on-the-fly LTL model checking algorithm.

An example of a simulation transition system generated by GeNeSim is depicted in Fig. 3 (on the right). It represents a qualitative simulation of the PLDE model of the double-negative loop motif presented in the previous section. The information included in the individual states denotes the address of the domain and the so-called *direction set property*. The address of each domain is given as a vector of discrete concentration levels of respective proteins in the symbolic concentration space (see Fig. 3). Direction set property (dsp) expresses the information concerning the potential phases in the respective domain computed by approximation. On the one hand, the dsp information is used as a key resource for generating successors of switching domains. On the other hand, computation of the dsp information requires exploration of all the potential neighbouring states. Such exploration takes an indispensable amount of time. Therefore saving of the dsp information into states accelerates generation of the simulation state space. For a particular domain D it

can gather the following values:

$$dsp(D) = \begin{cases} 2, \text{ if } D \text{ represents a steady state,} \\ 1, \text{ if the set of phases is empty (only if } D \text{ is switching),} \\ 0, \text{ otherwise.} \end{cases} \quad (1)$$

The value 1 has sense only for states which represent switching domains. The empty phase set signalises that the respective switching domain must be immediately left after it is entered. It symbolises the fact that in such a domain no substrate can keep its concentration constant.

In general, to minimise the memory needed for allocation of states, we have decided to save into states only the mentioned information. All other information is computed on-the-fly whenever it is needed for state space generation, and consequently, during a particular analysis.

5 Experiments

6 Conclusions

Results presented in the previous section show that the parallel approach accelerates simulation and model-checking of genetic regulatory networks. Average maximal rate of acceleration achieved by our experiments makes the parallel analysis 7.5 times faster than the sequential analysis with GNA. In particular, the parallel approach enables queries for models having up to 10 state variables to be answered in terms of minutes. Moreover, also larger models (more than 10 variables), which are not satisfactorily tractable by the explicit sequential approach, can be still analysed by the implicit parallel approach on suitably large clusters.

To summarise, our contribution is a demonstration of the use of parallel model-checking for biological systems. In particular, we provide a translation of a piecewise-linear model of a genetic regulatory network into a discrete transition system which serves as an input for the parallel model-checker DiVinE. The approach allows for parallel on-the-fly model-checking of larger networks than is possible by sequential algorithms. The preliminary experiments conducted with the tool confirm good scalability. Though we have focused on qualitative analysis, the DiVinE tool is also able to analyse some stochastic and quantitative properties as well. These extensions together with improvements of the GeNeSim implementation in speeding-up the state space exploration, and that way reaching practicable results for extremely large networks (having around 100 variables), remain for our future work.

References

[1] J. Barnat, L. Brim, I. Černá, P. Moravec, P. Ročkai, and P. Šimeček. DiVinE – A Tool for Distributed Verification (Tool Paper). In *Computer Aided Verification*, volume 4144/2006 of *LNCS*, pages 278–281. Springer, 2006.

[2] G. Batt, C. Belta, and R. Weiss. Model checking genetic regulatory networks with parameter uncertainty. In A. Bemporad, A. Bicchi, and G. Butazzo, editors, *Tenth International Workshop on Hybrid Systems: Computation and Control (HSCC'07)*, volume 4416 of *LNCS*, pages 61–75. Springer, 2007.

[3] G. Batt, C. Belta, and R. Weiss. Model checking liveness properties of genetic regulatory networks. In O. Grumberg and M. Huth, editors, *Thirteenth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07)*, volume 4424 of *LNCS*, pages 323–338. Springer, 2007.

[4] G. Batt, D. Bergamini, H. de Jong, H. Garavel, and R. Mateescu. Model checking genetic regulatory networks using GNA and CADP. In *Eleventh International SPIN Workshop on Model Checking Software (SPIN'04)*, volume 2989 of *LNCS*, pages 158–163. Springer, 2004.

[5] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Analysis and verification of qualitative models of genetic regulatory networks: A model-checking approach. In L. P. Kaelbling and A. Saffiotti, editors, *Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 370–375, 2005.

[6] 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. In *Thirteenth International Conference on Intelligent Systems for Molecular Biology*, pages 19–28, 2005.

[7] 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 : Nutritional stress response in escherichia coli. In *Eighth International Workshop on Hybrid Systems : Computation and Control (HSCC'05)*, volume 3414 of *LNCS*, pages 134–150. Springer, 2005.

[8] G. Batt, D. Ropers, H. de Jong, M. Page, and J. Geiselmann. Symbolic Reachability Analysis of Genetic Regulatory Networks using Qualitative Abstractions. Technical Report RR-6136, INRIA, 2007.

[9] G. Bernot, J-P. Comet, A. Richard, and J. Guespin. Application of Formal Methods to Biological Regulatory Networks: Extending Thomas' Asynchronous Logical Approach with Temporal Logic. *Journal of Theoretical Biology*, 229(3):339–347, 2004.

[10] James M. Bower and Hamid Bolouri. *Computational modeling of genetic and biochemical networks*. Cambridge : Bradford Book, 2001.

[11] L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman. Machine Learning Biochemical Networks from Temporal Logic Properties. *Transactions on Computational Systems Biology VI*, 4220:68–94, 2006.

[12] L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[13] I. Černá and R. Pelánek. Distributed explicit fair cycle detection (set based approach). In T. Ball and S.K. Rajamani, editors, *Model Checking Software. 10th International SPIN Workshop*, volume 2648 of *Lecture Notes in Computer Science*, pages 49 – 73. Springer Verlag, 2003.

[14] N. Chabrier and F. Fages. Symbolic Model Checking of Biochemical Networks. In N. Chabrier and F. Fages, editors, *Computational Methods in Systems Biology (CMSB'03)*, volume 2602 of *LNCS*, pages 149–162. Springer-Verlag, 2003.

[15] H. de Jong. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[16] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in Bacillus subtilis. *Bulletin of Mathematical Biology*, 66(2):261–300, 2004.

[17] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.

[18] H. de Jong, J-L. Gouzé, C. Hernandez, M. Page, T. Sari, and Geiselmann J. "hybrid modeling and simulation of genetic regulatory networks: A qualitative approach". In Oded Maler and Amir Pnueli, editors, *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003*, volume 2623 of *LNCS*, pages 267–282. Springer, 2003.

[19] H. de Jong, J-L. Gouz, C. Hernandez, T. Sari, M. Page, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. Technical Report RR-4407, INRIA-Helix, 2002.

[20] S. Williams et.al. The Potential of the Cell Processor for Scientific Computing. In *Proc. of 3rd Conference on Computing Frontiers*, pages 9–20. ACM Press, 2006.

[21] K. Fisler, R. Fraer, G. Kamhi, M. Y. Vardi, and Z. Yang. Is there a best symbolic cycle-detection algorithm? In *Proc. Tools and Algorithms for the Construction and Analysis of Systems*, volume 2031 of *LNCS*, pages 420–434. Springer-Verlag, 2001.

[22] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(No. 25):2340–2381, 1977.

[23] L. Glass. Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, 54(1):85–107, 1975.

[24] L. Glass. Global analysis of nonlinear kinectics. *Statistical Mechanics, Part B: Time Dependent Processes*, pages 311–349, 1977.

[25] L. Glass, C. Hill, and T. Mestl. Nonlinear dynamics of gene and neural networks. In *Nonlinear Phenomena in Biological and Physical Systems*, pages 249–262. Indian National Science Academy, 2000.

[26] Jean-Luc Gouzé and Tewfik Sari. A class of piecewise linear differential equations arising in biological models. Technical Report RR-4207, INRIA, 2001.

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

[28] J.M. McCollum, G.D.Peterson, Ch.D. Cox, and M.L.Simpson. Accelerating Gene Regulatory Network Modeling Using Grid-Based Simulation. In *SIMULATION 2004*, volume 80, pages 231–241. SAGE, 2004.

[29] J.S. Meredith, S.R. Alam, and J.S. Vetter. Analysis of a Computational Biology Simulation Technique on Emerging Processing Architectures. In *Proc. of Parallel and Distributed Processing Symposium*, pages 1–8. IEEE, 2007.

[30] T. Mestl, E. Plahte, and S.W.Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology*, 176(2):291–300, 1995.

[31] L. Popova-Zeugmann, M. Heiner, and I. Koch. Time petri nets for modelling and analysis of biochemical networks. *Fundam. Inform.*, 67(1-3):149–162, 2005.

[32] M. Schaub, Th. Henzinger, and J. Fisher. Qualitative networks: a symbolic approach to analyze biological signaling networks. *BMC Systems Biology*, 1:4, 2007.

[33] C. P. Sosa, T. Milledge, J. McAllister, and G. Z. Milledge. Life Sciences Molecular Dynamics Applications on the IBM System Blue Gene Solution: Performance Overview. Technical report, IBM, 2006.

[34] D. Thieffry and R. Thomas. Dynamical Behavior of Biological Regulatory Networks. *Bulletin of Mathematical Biology*, 57(2):277–297, 1995.