BioDiVinE: A Tool for Parallel Analysis of Multi-Affine ODE Models*

J. Barnat, L. Brim, I. Černá, S. Dražan, J. Fabriková, J. Láník, and D. Šafránek

Department of Computer Science, Faculty of Informatics
Masaryk University Brno, Czech Republic
safranek@fi.muni.cz

1 Introduction

The most widely-used modelling frameworks for the analysis of the dynamics of biological systems are based on the deterministic continuous approach of ordinary differential equations (ODE). The reduction of continuous models to discrete automata by a sequence of reductions, approximations, and abstractions allows formal methods for the automated analysis of temporal properties to be applied [6, 7, 4]. When dealing with large models from systems biology, standard discrete state-space exploration techniques do not provide acceptable response times for answering user queries and high-performance parallel algorithms are required. Owing to dynamical dependencies among state variables, the *state-space explosion problem* arises during reduction to discrete automata.

In the poster we present a prototype tool BioDiVinE [9] for parallel analysis of biological models based on mass action kinetics. In particular, the tool adapts the rectangular abstraction approach of multi-affine ODEs mathematically introduced in [8] and algorithmically tackled in [14, 6]. We contribute to the domain by means of a scalable algorithm. In particular, the contribution of BioDiVinE is three-fold. First, the tool provides a parallel on-the-fly state space generator for the rectangular abstraction (RATS). Second, the state space generation algorithm employs several heuristics [2] for reducing the extent of approximation by guiding the state generator to avoid spurious simulations. Finally, the embedded enumerative on-the-fly LTL model checker allows direct application of efficient parallel model checking algorithms to analysis of biological models. Our experiments [2] show that ODE models involving up-to 20 variables resulting in reachable state spaces having around 10^7 states can be sufficiently analysed (with responses in the order of tens of seconds) on a common cluster. BioDiVinE also provides a graphical module that allows two-dimensional visualisation of reachable state spaces.

2 Related Work

In our previous work [3] we have dealt with parallel model checking analysis of piece-wise affine ODE models [12]. The method allows fully qualitative analysis, since in the piece-wise affine approximation generating of the state space does not require to numerically enumerate the equations. Therefore that approach, in contrast to this one, is primarily devoted for models with unknown kinetic parameters. The price for this feature is higher time complexity of the state space generation. In particular, time appears there more critical than space while causing the parallel algorithms not to scale well.

In the current version of BioDiVinE all the kinetic parameters are required to be numerically specified. In such a situation there is an alternative possibility to do LTL model checking directly on numerical simulations [15, 10]. However, in the case of unknown initial conditions there appears the need to provide large-scale parameter scans resulting in huge number of simulations. On the contrary, the analysis conducted with BioDiVinE can be naturally generalised to arbitrary intervals of initial conditions by means of rectangular abstraction.

* This work has been partially supported by the Academy of Sciences of CR grant No. 1ET408050503 and the FP6 project No. NEST-043235 (EC-MOAN).

3 Toolset Description

BioDiVinE employs aggregate power of network-interconnected workstations (nodes) to analyse large-scale state transition systems whose exploration is beyond capabilities of sequential tools. System properties can be specified either directly in Linear Temporal Logic (LTL) or alternatively as processes describing undesired behaviour of systems under consideration (negative claim automata). From the algorithmic point of view, the tool implements a variety of novel parallel algorithms [11, 1] for cycle detection (LTL model checking). By these algorithms, the entire state space is uniformly split into partitions and every partition is distributed to a particular computing node. Each node is responsible for generating the respective state-space partition on-the-fly while storing visited states into the local memory.

The state space generator constructs the rectangular abstraction transition system for a given multi-affine system. The scheme of the tool architecture is provided in Figure 1. Library-level components are responsible for constructing, managing and distributing the state space. They form the core of the tool. The tool provides two graphical user interface components *SpecAff* — allowing editing of biological models in terms of chemical reactions, and *SimAff* — allowing visualisation of the simulation results.

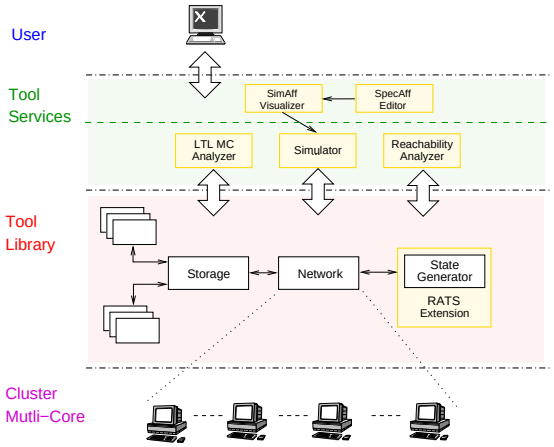

Fig. 1. BioDiVinE Toolset Architecture

The input (biochemical) model is specified by the following data:

- list of chemical species,
- list of partitioning thresholds given for each species,
- list of chemical reactions.

The biochemical model is then automatically translated into a multi-affine system of ODEs forming the mathematical model that can be analysed by BioDiVinE algorithms. The mathematical model consists of the following data:

- list of variables,
- list of (multi-affine) ODEs,
- list of partitioning thresholds given for each species,
- list of initial rectangular subspaces (the union of these subspaces forms the initial condition),
- Büchi automaton representing an LTL property (this data is not needed for simulation).

An example of a simple three-species model representing a single biochemical reaction $A + B \rightarrow C$ performed with rate 0.5 $M^{-1}s^{-1}$ is showed in Figure 2. The respective mathematical model is

showed in Figure 3 on the left in the textual `.bio` format. For each variable there is specified the equation as well as the list of real values representing individual threshold positions. The initial condition is defined in this particular case by a single rectangular subspace: $A \in \langle 6, 10\rangle, B \in \langle 4, 6\rangle$, and $C \in \langle 0.0001, 2\rangle$. The state space generated for this setting is depicted in Figure 3 on the right. Figure 4 demonstrates visualisation features of BioDiVinE.

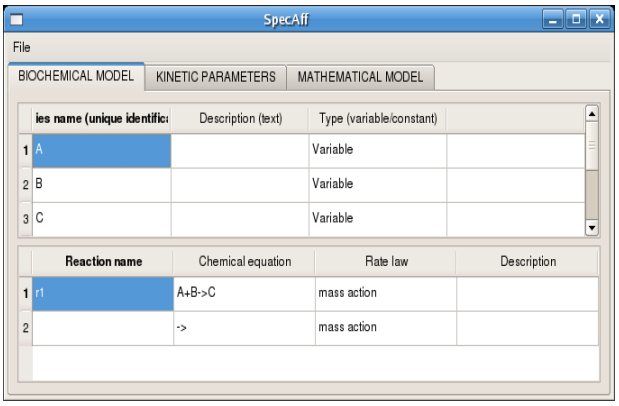

Fig. 2. A biochemical model specified in BioDiVinE GUI

```
VARS:A,B,C

EQ:dA = (-0.5)*A*B
EQ:dB = (-0.5)*A*B
EQ:dC = 0.5*A*B

TRES:A: 0.0001, 6, 10
TRES:B: 0.0001, 4, 6
TRES:C: 0.0001, 2, 4, 6

INIT: 6:10, 4:6, 0.0001:2
```

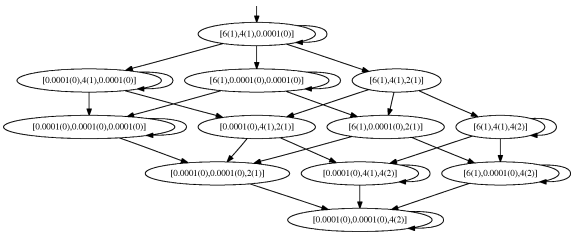

Fig. 3. A multi-affine ODE model and its state space generated by BioDiVinE

For model checking analysis, BioDiVinE relies on the parallel LTL model checking algorithms of the underlying DiVinE library [5]. A given LTL formula is translated into a Büchi automaton which represents its negation. That way the automaton represents the never claim property. The automaton is automatically generated for an LTL formula and merged with the mathematical model by `divine.combine` utility. An example of a model extended with a never claim property is showed in Figure 5. In particular, the automaton specified in terms of DiVinE language represents a never claim for the safety LTL formula $\mathbf{G}(A \leq 10)$ expressing that concentration of species A keeps under the given level.

For any multi-affine model extended with a never claim automaton as showed in Figure 5, the parallel model checking algorithms can be directly called. We have performed several experiments [2] in order to show scaling of the algorithms when distributed on several cluster nodes. Figure 6 shows scaling of model checking conducted on a simple model of a reaction network representing a catalytic reaction scaled for different numbers of intermediate products.

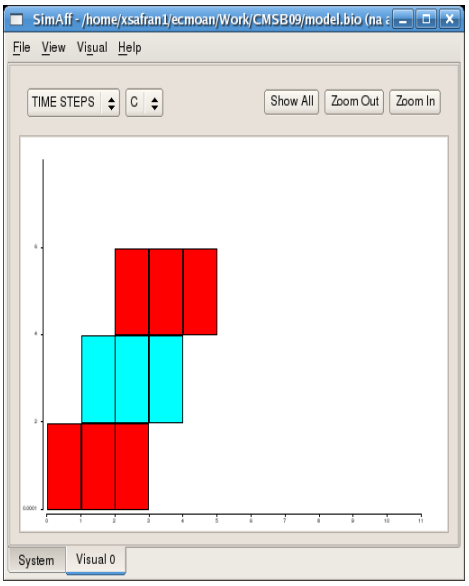

Fig. 4. A visualisation of the state space in BioDiVinE GUI

```
VARS:A,B,C

EQ:dA = (-0.5)*A*B
EQ:dB = (-0.5)*A*B
EQ:dC = 0.5*A*B

TRES:A: 0.0001, 6, 10
TRES:B: 0.0001, 4, 6
TRES:C: 0.0001, 2, 4, 6

INIT: 6:10, 4:6, 0.0001:2
```

```
process LTL_property {
state q1, q2;
init q1;
accept q2;
trans
q1 -> q2 { guard A>10; },
q1 -> q1 {},
q2 -> q2 {};
}

system sync property LTL_property;
```

Fig. 5. A multi-affine model extended with a never claim automaton

$$S + E \rightleftharpoons ES_1 \rightleftharpoons ES_2 \rightleftharpoons \cdots \rightleftharpoons ES_k \rightarrow P + E$$

$$\begin{aligned}
\dot{S} &= ES_1 - 0.01 \cdot E \cdot S \\
\dot{E} &= ES_1 - 0.01 \cdot S \cdot E + ES_1 \\
\dot{ES_1} &= 0.01 \cdot E \cdot S - ES_1 - ES_1 \\
\dot{ES_2} &= \cdot ES_1 - 2 \cdot ES_2 + ES_3 \\
&\vdots \\
\dot{ES_{k-1}} &= ES_{k-2} - 2 \cdot ES_{k-1} + ES_k \\
\dot{ES_k} &= ES_{k-1} - 2 \cdot ES_k
\end{aligned}$$

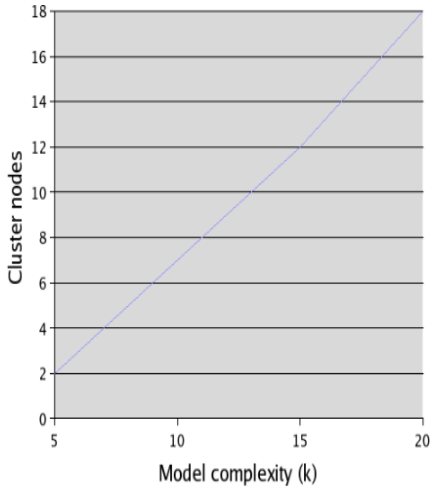

Fig. 6. Scaling of model checking algorithms on a homogeneous cluster

4 Conclusion

In this poster abstract we have presented the tool BioDiVinE for parallel model checking analysis of multi-affine ODE models. The tool currently supports rectangular abstraction of multi-affine systems providing discrete (over)approximation of the continuous state space. Properties of the model are specified in terms of LTL formulae. Parallel model checking algorithms can be used to either find an example of a particular behaviour or to decide that certain property is satisfied by all trajectories of the system starting at states given by particular initial conditions. The practicability of model checking is naturally limited by the level of overapproximation involved. Current applications of BioDiVinE show its usage for analysis of safety properties.

For future work we aim to employ BioDiVinE for analysis of biological models developed in the EC-MOAN project [13]. We also plan to improve the GUI in order to bring the tool closer to the community of biologists.

References

1. J. Barnat, L. Brim, and J. Chaloupka. From Distributed Memory Cycle Detection to Parallel LTL Model Checking. *Electronic Notes in Theoretical Computer Science*, 133(1):21–39, May 2005.
2. J. Barnat, L. Brim, I. Černá, S. Dražan, J. Fabriková, and D. Šafránek. Computational Analysis of Large-Scale Multi-Affine ODE Models. 2009. Accepted to High Performance Computational Systems Biology (HIBI) 2009.
3. J. Barnat, L. Brim, I. Černá, S. Dražan, J. Fabriková, and D. Šafránek. On Algorithmic Analysis of Transcriptional Regulation by LTL Model Checking. *Theoretical Computer Science*, 2009. in press.
4. J. Barnat, L. Brim, I. Černá, S. Dražan, and D. Šafránek. Parallel Model Checking Large-Scale Genetic Regulatory Networks with DiVinE. In *Proc. of From Biology to Concurrency and Back*, volume 194 of *ENTCS*, pages 35–50. Elsevier, 2007.
5. 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.
6. G. Batt, C. Belta, and R. Weiss. Model checking liveness properties of genetic regulatory networks. In *Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 4424 of *LNCS*, pages 323–338. Springer, 2007.
7. 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.
8. C. Belta and L.C.G.J.M. Habets. Controlling a class of nonlinear systems on rectangles. *IEEE Transactions on Automatic Control*, 51(11):1749–1759, 2006.
9. BioDiVinE Toolset. Masaryk University, 2009. http://divine.fi.muni.cz.
10. L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.
11. I. Černá and R. Pelánek. Distributed explicit fair cycle detection (set based approach). In *Model Checking Software. 10th International SPIN Workshop*, volume 2648 of *LNCS*, pages 49 – 73. Springer, 2003.
12. 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.
13. *FP6 EU Project EC-MOAN (Escherichia Coli-Modelling and Analysis)*, 2007. `http://www.ec-moan.org`.
14. M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. *Transactions of the Institute of Measurement and Control*, 2008. in press.
15. X. Liu, J. Jiang, O. Ajayi, X. Gu, D. Gilbert, and R. Sinnott. BioNessie. *Global Healthgrid: e-Science Meets Biomedical Informatics - Proceedings of HealthGrid 2008*, (138):147–157, 2008.