

Language-based Abstractions for Dynamical Systems

Andrea Vandin

IMT School for Advanced Studies Lucca
Lucca, Italy

`andrea.vandin@imtlucca.it`

Ordinary differential equations (ODEs) are the primary means to modelling dynamical systems in many natural and engineering sciences. The number of equations required to describe a system with high heterogeneity limits our capability of effectively performing analyses. This has motivated a large body of research, across many disciplines, into abstraction techniques that provide smaller ODE systems while preserving the original dynamics in some appropriate sense. In this paper we give an overview of a recently proposed computer-science perspective to this problem, where ODE reduction is recast to finding an appropriate equivalence relation over ODE variables, akin to classical models of computation based on labelled transition systems.

1 Introduction

Ordinary differential equations (ODEs) are the primary means to modelling dynamical systems in a wide range of natural and engineering sciences. When the complexity of the considered system is high, the number of equations required limits our capability of performing effective analyses. This has motivated a large body of research, across many disciplines, into abstraction techniques that provide smaller ODE systems preserving the original dynamics in some appropriate sense (e.g., [1, 40, 49, 15]). In particular, we refer to a number of techniques whose basic idea is to build a smaller ODE system with less variables and/or parameters, while preserving most of the original dynamics. Depending on the specific technique used, the obtained reduced model may be used, e.g., to provide a more compact description that abstracts from low-level details, or to improve the performance and scalability of required analyses. The development of abstraction techniques is widely investigated in many disciplines in the science and engineering domain that deal with dynamical systems, for instance:

- Ecological systems typically regard massive populations of entities that interact with the environment, and adapt to it. For this reason, quantitative abstractions have been widely studied in ecology (e.g., [40]). In this domain, abstraction techniques mainly exploit the fact that entities can range from molecules to whole organisms in their ecosystem (hence they act at different time scales), and the fact that there is often a high heterogeneity among the entities, e.g., they can have different age and metabolism, or can be located in different points in space. In particular, many techniques focus on the aggregation of large-scale models of ecosystems based on ODEs or difference equations (the time-discrete homologous of ODEs). Aggregations are obtained as a coarsening of the state space in terms of *macro-variables*, each describing a group of original variables (i.e., of entities in the original ecosystem). Aggregations have been developed to ignore, e.g., age differences in population models [40] or to exploit the fact that entities act at different time scales (see [3] for a survey).

- Cellular automata are a classic example of models used in physics to describe dynamical systems consisting of simple agents with small state spaces, whose interactions result in complex phenomena. Abstraction methods identify sets of neighbouring cells for which a coarse-grained description can still describe the overall behaviour of the original model [39].
- In computational systems biology, many abstraction techniques have been developed to cope with the combinatorial species and chemical reaction networks (CRN) that arise from the interactions of basic molecules which can undergo internal changes, or that can bind each other to form complex species. In particular, we refer to techniques developed for systems of ODEs arising from protein interaction networks (e.g. [10, 28]). Here, methods have been proposed to consider: a covering of the state space (where a variable may appear in more than one group) [28]; quotienting, induced by a partition of the ODE variables (e.g., [29]); and aggregations exploiting time-scale separation (e.g., [47]).
- Large-scale dynamical systems are common also in control engineering. Here, starting from the seminal paper [1], many approaches have been proposed to reduce the original model in a way that preserves controllability, i.e., the capability of driving the system to a desired state by using appropriate control inputs [51].

More recently, in the computer science community there has been an increasing interest towards quantitative models of computation based on ODEs, for example to use formal languages to describe biochemical models [48, 23, 9, 12, 13, 42, 22, 52] or as a deterministic approximation for languages with stochastic semantics [22, 35, 59].

Our own line of research [21, 56, 18, 20, 65, 19, 17, 38] consists of a computer-science perspective to the abstraction problem, borrowing ideas from the concurrency theory community. We recast the ODE reduction problem to that of finding an appropriate equivalence relation over ODE variables, akin to classical models of computation based on labelled transition systems. We studied such *differential equivalences* for two basic intermediate languages, trading expressivity for efficiency:

- i) *IDOL* (Intermediate Drift-Oriented Language) [20] covers a general class of non-linear ODEs with derivatives containing polynomials, rationals, minima/maxima, and absolute values. This is enough, e.g., to capture the existing ODE semantics of stochastic process algebras [38, 61, 60, 13]. The largest equivalences of IDOL terms are computed using a symbolic partition-refinement algorithm that exploits an encoding into a satisfiability modulo theories (SMT) problem;
- ii) *Reaction networks* [17, 19], a slight generalization of chemical reaction networks, characterise ODEs with polynomial derivatives. In this case, the partition refinement is based on Paige and Tarjan's seminal proposal [50], giving an efficient algorithm that runs in polynomial time.

Our framework for ODE reduction has been implemented in the tool ERODE [21] (<http://sysma.imtlucca.it/tools/erode/>), allowing us to provide evidence of effective reductions in realistic models from the literature.

This paper briefly reviews our framework for ODE reduction. A more detailed tutorial-like presentation unifying the two approaches can be found in [65], while [18, 56] address the more general problem of computing all differential equivalences of a model.

2 Framework Overview

Differential equivalences. Differential equivalences induce a quotienting of the ODE variables. Two distinct notions of equivalence for ODEs have been provided in [20]. *Forward differential equivalence*

(FDE) allows us to abstract from single variables, and to consider only cumulative information on each equivalence class. In particular, it guarantees that the original ODE system can be described in terms of a smaller ODE system having only one macro-variable per equivalence class that describes the sum of the variables within the equivalence class. For instance, consider the following ODE system with variables x_1 , x_2 , and x_3 :

$$\dot{x}_1 = -x_1, \quad \dot{x}_2 = k_1 \cdot x_1 - x_2, \quad \dot{x}_3 = k_2 \cdot x_1 - x_3, \quad (1)$$

where k_1 and k_2 are constants and the dot' operator denotes the derivative operator (with respect to time). It can be shown that the partition of variables $H = \{\{x_1\}, \{x_2, x_3\}\}$ is an FDE because

$$\dot{x}_1 = -x_1, \quad (\dot{x}_2 + \dot{x}_3) = \dot{x}_2 + \dot{x}_3 = (k_1 + k_2) \cdot x_1 - (x_2 + x_3). \quad (2)$$

In fact, by applying the simple change of variable $x_{2,3} = x_2 + x_3$, this is equivalent to writing

$$\dot{x}_1 = -x_1 \quad \dot{x}_{2,3} = (k_1 + k_2) \cdot x_1 - x_{2,3}.$$

This reduced ODE system contains one variable per equivalence class in H , each of which describes the sum of the solutions of the variables in the corresponding equivalence class. Thus, setting the *initial condition* $x_{2,3}(0) = x_2(0) + x_3(0)$ yields that the solution satisfies $x_{2,3}(t) = x_2(t) + x_3(t)$ at all time points t . As discussed in [17], the notion of FDE has proved to be particularly useful in the context of computational systems biology, where modellers are interested in a few observations of interest, like the evolution of the concentration of few species only, or of the sums of certain species that represent different configurations of the same molecule.

Backward differential equivalence (BDE) instead allows us to identify redundant dynamics in the original system. In fact, it equates variables that have the same solutions at all time points, if initialized equally. In the ODE system in Equation (1), if $k_1 = k_2$, we have that $H = \{\{x_1\}, \{x_2, x_3\}\}$ is also a BDE. Hence, we can reduce the original ODE system by removing either equation between x_2 and x_3 , say x_3 , and by rewriting every remaining occurrence of x_3 as x_2 :

$$\dot{x}_1 = -x_1 \quad \dot{x}_2 = k_1 x_1 - x_2.$$

The notion of BDE has proved to be useful in the context of evolutionary biology, where one is interested in understanding whether a system has evolved into another one preserving the original functionality [14, 15, 16]. In fact, it is well known that using larger networks instead of smaller ones of *equal functionality* is beneficial for enhanced stability with respect to stochastic noise [15]. In particular, in [20, 18] we have shown that BDE succeeded in tracing a certain functionality across two CRNs (by taking the CRN whose species and reactions are the disjoint union of those from the source and target one).

Differential equivalences guarantee that the relationship between the original model and the abstract one is exact. However, FDE leads to a loss of information, because information on the individual variables within an equivalence class may not be recovered in general. Instead, BDE preserves all information, however it can be applied only when the initial conditions are coherent with the considered equivalence classes: i.e., when all backward equivalent variables are initialized equally. Forward differential equivalence is closely related to the notion of exact ODE lumpability, thoroughly investigated in the chemistry domain (e.g., [57, 49, 46]). However no automated procedure exists to reduce an ODE system using this approach (e.g., [63]). To cope with this, i.e., to guarantee that our framework can be instantiated in a family of fully automatic reduction techniques, restrictions are imposed to be able to develop minimisation algorithms.

Symbolic minimisation. In [20] each ODE variable is treated explicitly as a real function and a differential equivalence is encoded in a logical formula over ODE variables. Thus, checking whether a candidate partition is BDE/FDE can be done symbolically using an encoding into satisfiability modulo theories (SMT) [5]. In fact, differential equivalences belong to the quantifier-free fragment of first-order logic. It is possible to restrict the admissible ODE systems to those for which an SMT solver for non-linear real arithmetic e.g. Z3, [24] is a decision procedure. This can be done by, roughly speaking, excluding trigonometric functions. Hence, e.g., we support rational functions, necessary to encode sophisticated biological kinetics like Hill’s one [5], and minima/maxima, necessary to capture the ODEs arising from terms of stochastic process algebras [35, 59, 36], or from queuing networks [58]. Hence, the technique is generic enough to be applied to a wide range of dynamical systems.

Let us consider the example in Equation (1), assuming $k_1 = k_2 = 1$, for which case we have shown that $H = \{\{x_1\}, \{x_2, x_3\}\}$ is a BDE. The condition for H to be a BDE has been shown to correspond to requiring that related variables with equal assignments always have equal derivatives [20]. This can be encoded in a quantifier-free first-order logical formula, which in the specific case of H is:

$$\phi_H := (x_2 = x_3) \implies (k_1 \cdot x_1 - x_2 = k_2 \cdot x_1 - x_3)$$

The SMT check $\text{sat}(\neg\phi_H)$ searches for an assignment of the variables x_1 , x_2 , and x_3 such that $\neg\phi$ holds. Hence, the partition is a BDE if and only if the procedure returns “unsat”, meaning that ϕ_H can not be falsified. This is the case for our example. More interestingly, in case the solver returns an assignment which satisfies $\neg\phi_H$, a *witness*, then we can use it within an iterative algorithm which computes the coarsest BDE partition that refines the initial one (i.e., that can be obtained by splitting the blocks of the initial one). Assume for example that we start from the candidate BDE partition $H_2 = \{\{x_1, x_2, x_3\}\}$. It can be shown that H_2 is not a BDE, and hence the SMT check $\text{sat}(\neg\phi_{H_2})$ is satisfiable. For instance, a witness for the satisfiability of $\neg\phi_{H_2}$ is $(x_1 = 1, x_2 = 1, x_3 = 1)$, which yields to different evaluations for the three derivatives:

$$\dot{x}_1 = -x_1 = 1, \quad \dot{x}_2 = k_1 \cdot x_1 - x_2 = 0, \quad \dot{x}_3 = k_2 \cdot x_1 - x_3 = 0$$

This suggests the implementation of a partition-refinement algorithm that *splits* the blocks of the current partition in sub-blocks of variables which cannot be distinguished using the returned witness. In other words, at the next iteration the candidate BDE partition would be $\{\{x_1\}, \{x_2, x_3\}\}$. A similar algorithm has been provided also for the FDE case. Note that this algorithm meets an important property often required for reduction techniques: the reduced model preserves user-defined observables of the original system. Indeed, each variable of interest for the modeller can be put in a singleton initial block. In the backward case, this allows us to provide an initial partition coherent with the initial conditions of the original model (that is, two variables are in the same initial block if their initial conditions are the same), a necessary condition for the reduced model to faithfully represent the original dynamics.

Syntactic minimisation. More efficient partition refinement algorithms can be provided for ODEs with derivatives that contain only multivariate polynomials of degree at most two [19]. This is quite a general class of ODEs, covering linear systems as well as chemical reaction networks. This approach is based on a syntactic representation of an ODE system in terms of a so-called *reaction network* (RN), which slightly generalizes the notion of chemical reaction network by allowing reaction rates to be also negative. Therefore, one can readily encode any polynomial ODE system as an RN [19]. In particular, an RN consists of a set of species/variables interacting by means of reactions parameterised by a real value. Intuitively, we use reaction networks to generalize labelled transition systems (LTS) so to consider

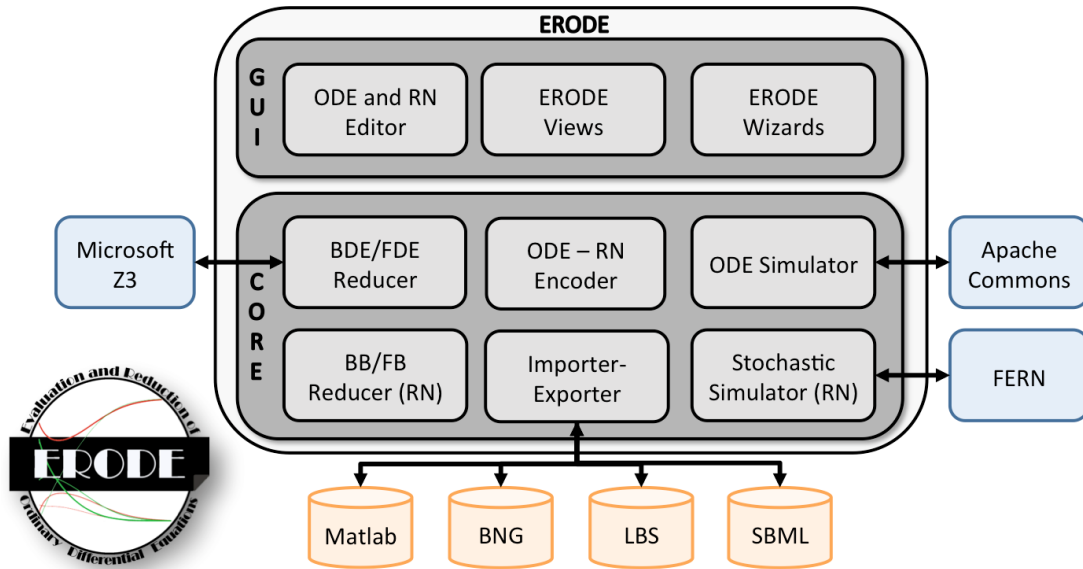


Figure 1: ERODE’s Architecture.

transitions among multisets of nodes rather than among individual nodes. Hence, there exists a loose connection between species and nodes of an LTS, and between reactions and transitions of an LTS.

We provide two bisimulation equivalences for such syntactic representation: the forward RN bisimulation (FB), and the backward RN bisimulation (BB), which are related to FDE and BDE, respectively [17]. Continuing the parallel between reaction networks and LTSs, our bisimulations are similar in spirit to quantitative bisimulations on LTSs, e.g., Larsen and Skou’s probabilistic bisimulation [44]. In particular, the largest RN bisimulations that refine a given input partition can be computed by generalizing Paige and Tarjan’s famous algorithm [50]. In [19] we propose a partition refinement algorithm along the lines of the best-performing analogues for Markov chain lumping such as [26] and [64], and for probabilistic transition systems [4]. The algorithm computes the largest FB/BB refining a given input partition of variables in $O(mn \log n)$ time, where m is the number of monomials in the ODE system and n is the number of variables.

Tool support: ERODE. We implemented our minimisation techniques in ERODE [21], a mature tool featuring a modern integrated development environment for the evaluation and reduction of ordinary differential equations. ERODE is on the Eclipse framework, it is multi-platform and does not require any installation process. The tool is available together with a manual and sample models at <http://sysma.imtlucca.it/tools/erode>.

Fig. 1 depicts ERODE’s architecture. It is organized in the GUI layer, and the core layer. Fig. 2 depicts the main components of the GUI layer, including a fully-featured text editor based on the *XTEXT* framework which supports syntax highlighting, content assist, error detection, and fix suggestions (top-middle of Fig. 2). Additionally, this layer offers a number of views, including a *project explorer* to navigate among different ERODE files (top-left of Fig. 2); an *outline* to navigate the parts of the currently open ERODE file (bottom-left of Fig. 2); a *plot view* to display ODE solutions (top-right of Fig. 2); and a *console view* to display diagnostic information (bottom-right of Fig. 2).

The core layer implements the minimization algorithms and related data structures for FDE, BDE,

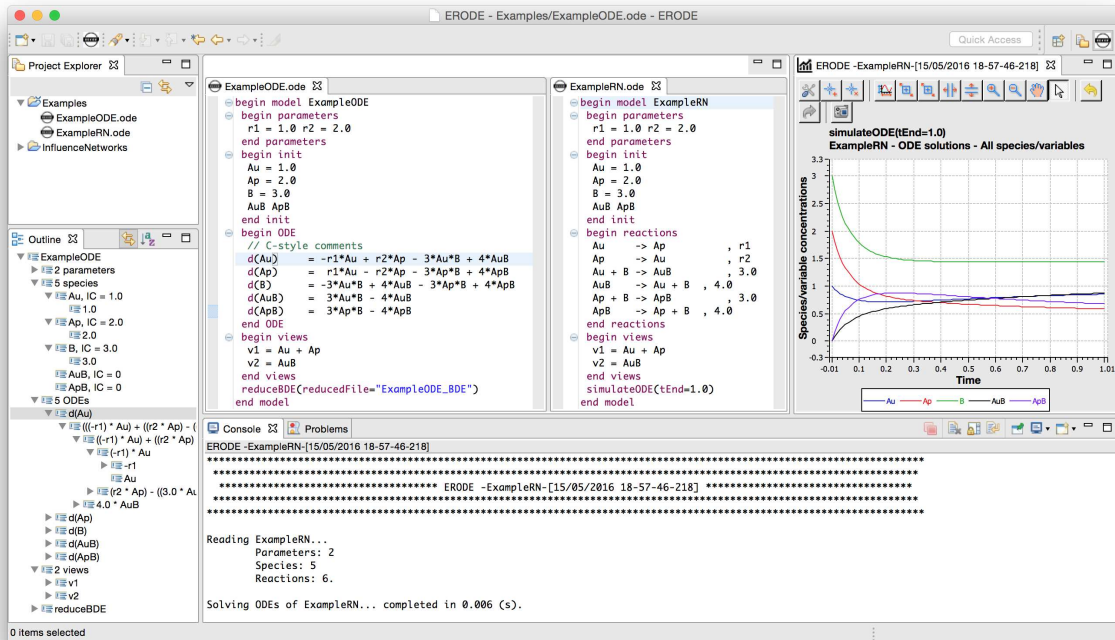


Figure 2: A screenshot of ERODE.

FB and BB. The algorithms for FB and BB reductions have been provided entirely in Java, while those for FDE and BDE reductions resort on the Z3 SMT solver, accessed via its Java APIs. The core layer also features encoding capabilities from RN to ODE (or IDOL) representation, and vice versa, as well as export/import functionalities for third-party formats, like biochemical models for BioNetGen [9] and Microsoft GEC [31], or ODEs defined in MATLAB. Also, models can be exported in the SBML format, an XML-based language for describing biological systems. Finally, this layer provides support for the numerical solutions of ODEs by means of the Apache Commons Maths library [2]. When the input is a CRN (i.e. an RN with only positive rates) it can also be interpreted as a CTMC, following an established approach [32]. Using the *FERN* library [30], ERODE features CTMC simulation.

By using a standard laptop machine, in [21] we have shown that our syntactic minimisation algorithms can scale to models with up to 2.5 million variables and 25 million monomials, with runtimes of a few minutes only. Instead, the symbolic minimisation scales up to 70 thousand variables in the backward case, and to 1 thousand variables in the forward case. As discussed in [20], the difference in performance and scalability between the backward and forward symbolic minimisation techniques are due to the fact that combinatorially many SMT checks among each pair of related variables are required in order to establish if a partition is FDE, while the BDE case requires only one check.

3 Conclusions

This paper presented two equivalences for ordinary differential equations (ODEs), forward and backward differential equivalences. In case the ODEs can be described in terms of the IDOL language, which

essentially restricts to ODEs whose derivatives do not contain exponential and trigonometric functions, then such ODE systems can be minimized up to these notions of equivalence using symbolic reduction techniques based on SMT solving.

If instead the derivatives of the considered ODE system are multivariate polynomials, then more efficient syntactic minimization techniques can be used, based on an encoding of the notions of differential equivalence in terms of two probabilistic bisimulations, forward and backward RN bisimulations. In [21] we have shown that the syntactic minimisation algorithms can scale to systems with millions of variables and monomials, terminating in a few minutes also in some challenging models. There are, however, further challenges ahead which we wish to tackle.

Forward bisimulation is only a sufficient condition for FDE; while it provides significant reductions in practice [17, 19], some other examples from the literature demonstrate that the algorithm may miss some FDE reductions (see [20]). In ongoing work we are relaxing the notion of forward bisimulation so to fully characterize FDE (for multivariate polynomial derivatives of degree at most two). Instead, backward bisimulation characterises BDE, and hence the corresponding syntactic minimisation technique from [19] should be used when dealing with polynomial ODEs of degree at most two. Also, we plan to extend the RN representation and the two notions of forward and backward RN bisimulations in order to be able to apply them to polynomial derivatives with arbitrary degree.

As regards our symbolic reduction techniques, we used SMT only in a black box fashion. It could be interesting to study ad-hoc heuristics for SMT solving that could lead to better performance or scalability. For the same reason, it would be interesting to consider parallelised versions of such symbolic reduction techniques. Also, another interesting research line related to our symbolic reduction techniques is that of further exploiting the generality of SMT in order to handle models with uncertainties in rates (a well-known issue in mathematical biology): here the SMT framework can already be easily extended to compute partitions that are differential equivalences under all possible assignments of such uncertain parameters, left as free variables in the satisfiability problem (similarly to the SMT-based parametric minimisation approach of [25] for probabilistic models written in PRISM [41]).

Lastly, we remark that our techniques regard exact aggregations. In some cases, however, one might be interested in more permissive, approximate notions that do not discriminate ODE variables with nearby trajectories (e.g., [54, 11, 34, 62, 37, 27, 43]). Such approximate variants can be considered as weaker notions that, e.g., allow variability in the parameters, considering the exact versions as a degenerate case in which no such variability is needed. We remark that, at least for the numerical benchmarks considered so far with realistic models, the exact reductions can already be quite effective. Of course, approximate ones might be able to provide even coarser descriptions. In this case, however, the main challenge is to be able to relate the variability in the parameters tolerated by the coarsening procedure with the error incurred when considering an approximate, smaller model, instead of the original one. In an ongoing research we are developing approximate variants of our differential equivalences, aiming at maintaining computational tractability, and certified error bounds that do not grow fast with time.

As regards our tool ERODE, we also plan to enrich its family of offered analysis techniques. For example, we plan to extend ERODE, and in particular its stochastic simulator, with statistical model checking [45] capabilities. This can be obtained upon integration with the statistical analyser MultiVeStA [55], which has already proven useful in the analysis of a wide range of scenarios, including software product lines [6, 7], crowd-steering [53], public transportation systems [33], and swarm robotics [8].

Acknowledgement. The author is indebted to Luca Cardelli, Mirco Tribastone, and Max Tschaiowski for the collaboration on the development of the presented framework, and for many fruitful discussions.

References

- [1] M. Aoki (1968): *Control of large-scale dynamic systems by aggregation*. *IEEE Trans. Autom. Control* 13(3), pp. 246–253, doi:10.1109/TAC.1968.1098900.
- [2] *Apache Commons Mathematics Library*, <http://commons.apache.org/proper/commons-math/>.
- [3] P. Auger, R. Bravo de la Parra, J. C. Poggiale, E. Sánchez & T. Nguyen-Huu (2008): *Aggregation of Variables and Applications to Population Dynamics*, pp. 209–263. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-540-78273-5_5.
- [4] C. Baier, B. Engelen & M. E. Majster-Cederbaum (2000): *Deciding Bisimilarity and Similarity for Probabilistic Processes*. *J. Comput. Syst. Sci.* 60(1), pp. 187–231, doi:10.1006/jcss.1999.1683.
- [5] C.W. Barrett, R. Sebastiani, S.A. Seshia & C. Tinelli (2009): *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*, chapter Satisfiability Modulo Theories. IOS Press, doi:10.3233/978-1-58603-929-5-825.
- [6] M. H. ter Beek, A. Legay, A. Lluch-Lafuente & A. Vandin (2015): *Statistical analysis of probabilistic models of software product lines with quantitative constraints*. In: *SPLC 2015*, ACM, pp. 11–15, doi:10.1145/2791060.2791087.
- [7] M. H. ter Beek, A. Legay, A. Lluch-Lafuente & A. Vandin (2016): *Statistical Model Checking for Product Lines*. In: *ISoLA 2016, LNCS 9952*, pp. 114–133, doi:10.1007/978-3-319-47166-2_8.
- [8] L. Belzner, R. De Nicola, A. Vandin & M. Wirsing (2014): *Reasoning (on) Service Component Ensembles in Rewriting Logic*. In: *Specification, Algebra, and Software, LNCS 8373*, Springer, pp. 188–211, doi:10.1007/978-3-642-54624-2.
- [9] M. L. Blinov, J. R. Faeder, B. Goldstein & W. S. Hlavacek (2004): *BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains*. *Bioinformatics* 20(17), pp. 3289–3291, doi:10.1093/bioinformatics/bth378.
- [10] N. M. Borisov, N. I. Markevich, J. B. Hoek & B. N. Kholodenko (2005): *Signaling through Receptors and Scaffolds: Independent Interactions Reduce Combinatorial Complexity*. *Biophysical Journal* 89(2), pp. 951–966, doi:10.1529/biophysj.105.060533.
- [11] F. van Breugel & J. Worrell (2006): *Approximating and computing behavioural distances in probabilistic transition systems*. *TCS* 360(1–3), pp. 373–385, doi:10.1016/j.tcs.2006.05.021.
- [12] L. Calzone, F. Fages & S. Soliman (2006): *BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge*. *Bioinformatics* 22(14), pp. 1805–1807, doi:10.1093/bioinformatics/btl1172.
- [13] L. Cardelli (2008): *On process rate semantics*. *TCS* 391(3), pp. 190–215, doi:10.1016/j.tcs.2007.11.012.
- [14] L. Cardelli (2014): *Morphisms of reaction networks that couple structure to function*. *BMC Systems Biology* 8(1), doi:10.1186/1752-0509-8-84.
- [15] L. Cardelli, A. Csikász-Nagy, N. Dalchau, M. Tribastone & M. Tschaikowski (2016): *Noise Reduction in Complex Biological Switches*. *Scientific Reports* 6, p. 20214, doi:10.1038/srep20214.
- [16] L. Cardelli, R. D. Hernansaiz-Ballesteros, N. Dalchau & A. Csikász-Nagy (2017): *Efficient Switches in Biology and Computer Science*. *PLOS Computational Biology* 13(1), pp. 1–16, doi:10.1371/journal.pcbi.1005100.
- [17] L. Cardelli, M. Tribastone, M. Tschaikowski & A. Vandin (2015): *Forward and Backward Bisimulations for Chemical Reaction Networks*. In: *CONCUR*, pp. 226–239, doi:10.4230/LIPIcs.CONCUR.2015.226.
- [18] L. Cardelli, M. Tribastone, M. Tschaikowski & A. Vandin (2016): *Comparing Chemical Reaction Networks: A Categorical and Algorithmic Perspective*. In: *LICS 2016*, pp. 485–494, doi:10.1145/2933575.2935318.
- [19] L. Cardelli, M. Tribastone, M. Tschaikowski & A. Vandin (2016): *Efficient Syntax-Driven Lumping of Differential Equations*. In: *TACAS 2016*, pp. 93–111, doi:10.1007/978-3-662-49674-9_6.

- [20] L. Cardelli, M. Tribastone, M. Tschaikowski & A. Vandin (2016): *Symbolic computation of differential equivalences*. In: *POPL 2016*, pp. 137–150, doi:10.1145/2837614.2837649.
- [21] L. Cardelli, M. Tribastone, M. Tschaikowski & A. Vandin (2017): *ERODE: A Tool for the Evaluation and Reduction of Ordinary Differential Equations*. In: *TACAS 2017*, pp. 310–328, doi:10.1007/978-3-662-54580-5_19.
- [22] F. Ciocchetta & J. Hillston (2009): *Bio-PEPA: A framework for the modelling and analysis of biological systems*. *TCS* 410(33-34), pp. 3065–3084, doi:10.1016/j.tcs.2009.02.037.
- [23] V. Danos & C. Laneve (2004): *Formal molecular biology*. *TCS* 325(1), pp. 69–110, doi:10.1016/j.tcs.2004.03.065.
- [24] L. De Moura & N. Bjørner (2008): *Z3: An Efficient SMT Solver*. In: *TACAS*, pp. 337–340, doi:10.1007/978-3-540-78800-3_24.
- [25] C. Dehnert, J.-P. Katoen & D. Parker (2013): *SMT-Based Bisimulation Minimisation of Markov Models*. In: *VMCAI'13, LNCS 7737*, Springer, pp. 28–47, doi:10.1007/978-3-642-35873-9_5.
- [26] S. Derisavi, H. Hermanns & W.H. Sanders (2003): *Optimal state-space lumping in Markov chains*. *Information Processing Letters* 87(6), pp. 309–315, doi:10.1016/S0020-0190(03)00343-0.
- [27] J. Desharnais, V. Gupta, R. Jagadeesan & P. Panangaden (2004): *Metrics for labelled Markov processes*. *Theor. Comput. Sci.* 318(3), pp. 323–354, doi:10.1016/j.tcs.2003.09.013.
- [28] J. Feret, V. Danos, J. Krivine, R. Harmer & W. Fontana (2009): *Internal coarse-graining of molecular systems*. *PNAS* 106(16), pp. 6453–6458, doi:10.1073/pnas.0809908106.
- [29] J. Feret, T. Henzinger, H. Koepl & T. Petrov (2012): *Lumpability abstractions of rule-based systems*. *TCS* 431(0), pp. 137–164, doi:10.1016/j.tcs.2011.12.059.
- [30] E. Florian, C. C. Friedel & R. Zimmer (2008): *FERN - a Java framework for stochastic simulation and evaluation of reaction networks*. *BMC Bioinformatics* 9(1), p. 356, doi:10.1186/1471-2105-9-356.
- [31] *Microsoft GEC*, <http://research.microsoft.com/en-us/projects/gec/>.
- [32] D. T. Gillespie (1977): *Exact Stochastic Simulation of Coupled Chemical Reactions*. *Journal of Physical Chemistry* 81(25), pp. 2340–2361, doi:10.1021/j100540a008.
- [33] S. Gilmore, M. Tribastone & A. Vandin (2014): *An Analysis Pathway for the Quantitative Evaluation of Public Transport Systems*. In: *IFM 2014*, pp. 71–86, doi:10.1007/978-3-319-10181-1_5.
- [34] V. Gupta, R. Jagadeesan & P. Panangaden (2006): *Approximate reasoning for real-time probabilistic processes*. *LMCS* 2(1), doi:10.2168/LMCS-2(1:4)2006.
- [35] R. A. Hayden & J. T. Bradley (2010): *A fluid analysis framework for a Markovian process algebra*. *TCS* 411(22-24), pp. 2260–2297, doi:10.1016/j.tcs.2010.02.001.
- [36] J. Hillston, M. Tribastone & S. Gilmore (2011): *Stochastic Process Algebras: From Individuals to Populations*. *The Computer Journal*, doi:10.1093/comjnl/bxr094.
- [37] G. Iacobelli & M. Tribastone (2013): *Lumpability of fluid models with heterogeneous agent types*. In: *DSN*, pp. 1–11, doi:10.1109/DSN.2013.6575346.
- [38] G. Iacobelli, M. Tribastone & A. Vandin (2015): *Differential Bisimulation for a Markovian Process Algebra*. In: *MFCS 2015*, pp. 293–306, doi:10.1007/978-3-662-48057-1_23.
- [39] N. Israeli & N. Goldenfeld (2006): *Coarse-graining of cellular automata, emergence, and the predictability of complex systems*. *Phys. Rev. E* 73, p. 026203, doi:10.1103/PhysRevE.73.026203.
- [40] Y. Iwasa, V. Andreasen & S. Levin (1987): *Aggregation in model ecosystems. I. Perfect aggregation*. *Ecological Modelling* 37(3-4), pp. 287–302, doi:10.1016/0304-3800(87)90030-5.
- [41] M.Z. Kwiatkowska, G. Norman & D. Parker (2011): *PRISM 4.0: Verification of Probabilistic Real-Time Systems*. In: *CAV*, pp. 585–591, doi:10.1007/978-3-642-22110-1_47.
- [42] M. Kwiatkowski & I. Stark (2008): *The Continuous pi-Calculus: A Process Algebra for Biochemical Modelling*. In: *CMSB*, pp. 103–122, doi:10.1007/978-3-540-88562-7_11.

- [43] K.G. Larsen, R. Mardare & P. Panangaden (2012): *Taking It to the Limit: Approximate Reasoning for Markov Processes*. In: *MFCs*, pp. 681–692, doi:10.1007/978-3-642-32589-2_59.
- [44] K.G. Larsen & A. Skou (1991): *Bisimulation through probabilistic testing*. *Information and Computation* 94(1), pp. 1–28, doi:10.1016/0890-5401(91)90030-6.
- [45] A. Legay, B. Delahaye & S. Bensalem (2010): *Statistical model checking: An overview*. In: *RV 2010*, Springer, pp. 122–135, doi:10.1007/978-3-642-16612-9_11.
- [46] G. Li & H. Rabitz (1989): *A general analysis of exact lumping in chemical kinetics*. *Chemical Engineering Science* 44(6), pp. 1413–1430, doi:10.1016/0009-2509(89)85014-6.
- [47] J. D. Murray (2002): *Mathematical Biology I: An Introduction, 3rd edition*. Springer, doi:10.1007/b98868.
- [48] M. Nagasaki, S. Onami, S. Miyano & H. Kitano (1999): *Bio-calculus: Its Concept and Molecular Interaction*. *Genome Informatics* 10, pp. 133–143, doi:10.11234/gi1990.10.133.
- [49] M. S. Okino & M. L. Mavrouniotis (1998): *Simplification of Mathematical Models of Chemical Reaction Systems*. *Chemical Reviews* 2(98), pp. 391–408, doi:10.1021/cr9502231.
- [50] R. Paige & R. Tarjan (1987): *Three Partition Refinement Algorithms*. *SIAM Journal on Computing* 16(6), pp. 973–989, doi:10.1137/0216062.
- [51] G. J. Pappas (2003): *Bisimilar linear systems*. *Automatica* 39(12), pp. 2035–2047, doi:10.1016/j.automatica.2003.07.003.
- [52] M. Pedersen & G. D. Plotkin (2010): *A Language for Biochemical Systems: Design and Formal Specification*. In: *Trans. Comp. Systems Biology XII*, Springer, pp. 77–145, doi:10.1007/978-3-642-11712-1_3.
- [53] D. Pianini, S. Sebastio & A. Vandin (2014): *Distributed statistical analysis of complex systems modeled through a chemical metaphor*. In: *HPCS 2014, IEEE*, pp. 416–423, doi:10.1109/HPCSim.2014.6903715.
- [54] A. Di Pierro, C. Hankin & H. Wiklicky (2003): *Quantitative Relations and Approximate Process Equivalences*. In: *CONCUR*, pp. 498–512, doi:10.1007/978-3-540-45187-7_33.
- [55] S. Sebastio & A. Vandin (2013): *MultiVeStA: Statistical model checking for discrete event simulators*. In: *Valuetools 2013, ACM*, pp. 310–315, doi:10.4108/icst.valuetools.2013.254377.
- [56] S. Tognazzi, M. Tribastone, M. Tschaikowski & A. Vandin (2017): *EGAC: A Genetic Algorithm to Compare Chemical Reaction Networks*. In: *GECCO 2017 (to appear)*, doi:10.1145/3071178.3071265.
- [57] J. Toth, G. Li, H. Rabitz & A.S. Tomlin (1997): *The Effect of Lumping and Expanding on Kinetic Differential Equations*. *SIAM JAM* 57(6), pp. 1531–1556, doi:10.1137/S0036139995293294.
- [58] M. Tribastone (2013): *A Fluid Model for Layered Queueing Networks*. *IEEE Transactions on Software Engineering* 39(6), pp. 744–756, doi:10.1109/TSE.2012.66.
- [59] M. Tribastone, S. Gilmore & J. Hillston (2012): *Scalable Differential Analysis of Process Algebra Models*. *IEEE Trans. Software Eng.* 38(1), pp. 205–219, doi:10.1109/TSE.2010.82.
- [60] M. Tschaikowski & M. Tribastone (2012): *Exact fluid lumpability for Markovian process algebra*. In: *CONCUR, LNCS*, pp. 380–394, doi:10.1007/978-3-642-32940-1_27.
- [61] M. Tschaikowski & M. Tribastone (2014): *Exact fluid lumpability in Markovian process algebra*. *TCS* 538, pp. 140–166, doi:10.1016/j.tcs.2013.07.029.
- [62] M. Tschaikowski & M. Tribastone (2016): *Approximate Reduction of Heterogenous Nonlinear Models With Differential Hulls*. *IEEE Trans. Automat. Contr.* 61(4), pp. 1099–1104, doi:10.1109/TAC.2015.2457172.
- [63] T. Turanyi & A. Tomlin (2014): *Analysis of Kinetic Reaction Mechanisms*. Springer, doi:10.1007/978-3-662-44562-4.
- [64] A. Valmari & G. Franceschinis (2010): *Simple $O(m \log n)$ Time Markov Chain Lumping*. In: *TACAS*, pp. 38–52, doi:10.1007/978-3-642-12002-2_4.
- [65] A. Vandin & M. Tribastone (2016): *Quantitative Abstractions for Collective Adaptive Systems*. In: *SFM 2016, Bertinoro Summer School*, pp. 202–232, doi:10.1007/978-3-319-34096-8_7.