

Via the examiner to the
Examination Office
Computer and Information Science
Attention of Mrs Rellstab
Box 188 / room PZ 807

Application received on:

Registration for the Master's Project
(according to the current examination regulations)

Last name: Klopper First name: Fabian
Student ID number: 01/956507

Hereby I register for the Master's Project.¹

Course title: Master's Project Modelling complex, self-organising Systems

Examiner: Prof. Dr. Tatjana Petrov

Examination number (see ZEuS): INF-14160

Topic: Minimization of Stochastic Dynamical Systems
- A Comparison of Models and Methods

Date of registration: 05.05.2020

Intended due date: 02.08.2020

The Master's Project has an extent of 9 ECTS-Credits, what corresponds to an average workload of 270 working hours (ca. 7 weeks full-time work).

☒ Enclosure: timetable/milestones

Konstanz, 27.04.2020

Konstanz, 27.04.2020

Signature, student

Signature, examiner

¹ The withdrawal from the examination is possible without consequences within two weeks after the date of registration. The withdrawal has to be declared to the examiner and the examination office in written form. After the two-week period the general not-pass- and cancellations rules are valid.

Minimization of Stochastic Dynamical Systems

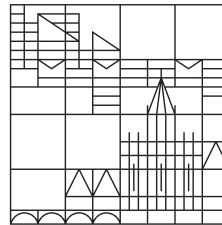
A Comparison of Models & Methods

M.Sc. Topic Proposal

by

Fabian Klopfer

Universität
Konstanz



Faculty of Sciences

Department of Computer and Information Science

Modelling of Complex Self-Organizing Systems Group

Supervisor: Dr. Stefano Tognazzi
Professor: Prof. Dr. Tatjana Petrov

Konstanz, 2020

1. Introduction

Stochastic dynamical systems are present in many natural phenomenon. Different methods of modelling these yield different advantages when it comes to composition, minimization, executability, expressiveness, succinctness and granularity.

The two representations under consideration in the proposed project — namely (weighted) automata and differential equations (DE) — may be subdivided resulting in distinct properties. For example an automaton-oriented approach allows for degrees of freedom when defining the transition function, while a differential equation-based approach differs in the constraints and assumptions the concrete form can make.

Chemical reaction networks [36], quantum computing [29] and brownian motion [15] are just some examples. The first one may be described using ordinary differential equations (ODEs), stochastic process algebras, stochastic Petri nets, stoichiometric equations, guarded commands and continuous-time Markov chains [21]. The second example is often modelled using differential equations [51], automata and grammars [29], whereas the third example was proven to exist as phenomenologically described using stochastic differential equations [15].

2. Problem Statement & Objectives of Study

The mentioned models may quickly reach the border to the computationally feasible, if one wants to extract exact results matching the real world. Especially required are not only careful implementations — no matter if it's numerical solvers, sampling techniques or even physical hardware — but also appropriate minimization methods with feasible computational complexity. Another desired trait is to reduce the system further to only answer the question under consideration exactly, ignoring other aspects that are irrelevant to the examiner.

During the project and the subsequent thesis those methods and their properties shall be the subject of my studies. Explicitly I want to investigate automata-based & differential equation-based modelling and the properties the different approaches have, the reduction or minimization of both models without and with simplifying assumptions (i.e. only a specific information is relevant), the connections and discords between the approaches, their properties and the results of reductions or minimizations of each under varying assumptions and conditions.

This induces the following goals:

1. Implement the minimization algorithm for weighted automata proposed by [26].
2. Implement the reduction algorithm for DEs proposed by [9].
3. Develop several reference examples serving as reproducible benchmarks in terms of qualitative and quantitative results regarding the reductions or minimization performable on each.
4. Write a report including a formal description of the models, implications by refinements and extensions, connection & discords, as well as comparisons of the results of the reductions or minimization.

3. Methods & Literature

As project and thesis contain mainly two aspects namely self-directed learning and some sort of contribution, the related literature may be divided into two categories: First educational textbooks & references containing general definitions, theorems and second application or research-oriented references & papers containing theorems and algorithms on the topic where the contribution shall be made. For this scope the literature from the first category is on differential equations, weighted

automata and additional topics that are relevant due to usage in the second category. The literature for the second category will be discussed in the next section.

3.1. Differential Equations

A differential equation is an equation relating functions with their derivatives.

Those equations are called ordinary, if the unknown functions are functions of one independent variable, while they're called partial if the function depends on more than one independent variable and contains partial derivatives.

A differential equation is called linear, if it is linear in the unknown function and non-linear if the equation is not expressible as linear in the unknown function. The coefficients of differential equations may be arbitrary, i.e. non-linear even when the differential equation is linear.

Differential equations can also be homogenous or heterogenous. They're called homogenous, if all functions in the equation are of the same degree and homogenous, that is if you multiply all parameters of the function by a scaling factor, the value of the function is also multiplied by the scaling factor to the power of the function degree. If a DE is not homogenous it is called heterogenous.

There are some extensions to the framework: Fractional differential equations generalize to non-integer derivatives, integro-differential equations is a generalization to relate functions not only to their derivatives but also to integrals of themselves, functional differential equations allow to consider the derivatives with varying argument values (e.g. $x + 1$, $x - 3$ instead of x) what is widely used in electrical engineering [40]. Differential-algebraic equations allow defining constraints on the solution space by introducing algebraic terms in the differential equation e.g. in electrodynamics when Kirchhoff's law is to be preserved. Stochastic differential equations enable modelling stochastic processes using differential equations with additional stochastic terms. This list is not complete not comprehensive.

As the field is rather broad, a lot of different references are required, which are mentioned here and listed in the bibliography: [27, 37, 33, 20, 54, 50, 48, 44, 31].

3.2. Automata

Automata are defined by tuples of different length depending on the concrete subtype but always consist of a set of states, and a transition function. Those models differ in the type of memory, the transition and acceptance semantics and define implicitly languages that follow certain construction rules called grammar.

Automata theory is closely related to Algebraic structures: The set of all input characters including the empty string ϵ can be seen as a Monoid with the concatenation operator. Transition systems are sometimes specified using matrix notations, often a sparse representation is more succinct. For instance for Finite State Automata one matrix would be necessary for each input character, specifying whether there is a transition between states for this character.

The expressiveness & complexity of grammars and the automaton model are directly coupled as the transitions and productions of both models are. Automata may either be deterministic or non-deterministic, meaning that there is only one or many different choices per input respectively when making a transition. If the environment decides what transition is taken — e.g. when making a typo and the spell checker marks it so that you correct it — it is called external non-determinism. If the automaton decides which transition is taken — e.g. when you make a typo, notice it by visual inspection and correct it — it's called internal non-determinism. Important is that for the same input character you have different transitions and that either factors outside of the system or inside the system decide what transitions are to be taken.

Finite State automata are memoryless and are equivalent to regular languages. Pushdown automata have a stack and are equivalent to context-free languages while Turing machines have unlimited, addressable memory and can express context-sensitive or unrestricted languages.

An extension to this basic framework are probabilistic automata where certain transitions are assigned weights that have to fulfill the Kolmogorov axioms. The concrete semantics depend on the model that is chosen and there is quite a zoo of models [45]. There are also probabilistic extensions of automata with memory but overall there is no clear hierarchy like in the non-probabilistic case. [45] attempted to order the chaos a bit but only covered memoryless automata and ordered them by the generality of the transition function. The implications of different memory models in the probabilistic context has not been studied yet to the best of the authors knowledge.

A generalization of classical and probabilistic automata are weighted automata (WA), introduced by Schützenberger [42] in 1961. Each transition is associated with a certain weight, cost or more abstract quantity, thus assigning words quantities when the WA is executed with the word as input. Weights are often modelled using a semi-ring $SR = (S, +, \cdot, 0, 1)$, that is a ring without additive inverse elements (or simply put without subtraction), while words are described by the monoid Σ^* . A function $\Sigma^* \rightarrow SR$ is called a formal power series. Classical automata define a power series over Σ^* and the boolean semiring, that is a series of 1 either terminated by a zero if there is no more transition that fulfills the word structure or by a 1 implying acceptance. An important aspect of weighted automata is that they generalize the classical automata and therefore formal languages to formal power series, which are seen as weighted languages. This means that for example in contrast to finite state automata on the boolean semi ring where an event either happens or it doesn't, it's possible to model probabilities of events, costs of error correction if an event deviates from the base case and the like.

As the field of automata — especially weighted and probabilistic ones — is relatively new compared to the field of DEs — Probabilistic Automata were first defined in 1956 [43] in contrast to DEs which were developed in the 17th century by Newton [30] — there are less sophisticated textbooks to rely on, so that more papers are used. Namely the references are [45, 43, 16, 39, 14, 35, 21, 47, 8, 46, 18, 1, 11, 12, 38, 41].

4. Related Work on Minimization

4.1. Automata

Minimization of DFAs is solvable in $\mathcal{O}(ns \log n)$ where n is the number of states and s the number of characters in the input alphabet using Hopcroft's algorithm [22] which is an extensions to a previously developed algorithm by Moore [43]. Another method reverses the edges to get an NFA, which can be converted to a DFA using the power set construction. Reversing the edges again yields the minimal DFA for the language [5].

NFAs can be converted to DFAs and then minimized using the algorithms referenced above, but the standard conversion algorithm — namely power set construction — has exponential runtime and is thus infeasible for large state spaces. There are algorithms for minimizing NFAs without conversions to DFAs. An early approach was exponential in runtime just like the conversion but provided a basis for future work that improved significantly [24]. Jiang and Ravikumar prove that explicit NFA minimization is PSPACE-complete by extending Kameda's work [23]. Tarjan provides an efficient bi-simulation algorithm for minimizing NFAs resulting in $\mathcal{O}(m \log n)$ runtime complexity.

Regarding probabilistic automata, Kiefer and Wachter provide an algorithm with polynomial runtime ($\mathcal{O}(\sum |n^3|)$). For more specific probabilistic automata models for example Markov chains there are various exact [36, 7, 25] and approximate algorithms [17, 2] of different complexity. The most efficient procedure to the best of the authors knowledge is in $\mathcal{O}(m \log n)$ with m the number of edges and n the number of states [49, 53]. Buchholz proposes methods for forward and backward bisimulation also resulting in $\mathcal{O}(m \log n)$ runtime [6].

A promising direction of research with respect to minimization of automata is the minimization of weighted automata. Kiefer et al. provide an algorithm that is in NC complexity (i.e. decidable in polylogarithmic time with a polynomial number of processors) [26]. Mohri combines moving the weights close to the initial states with the Hopcroft algorithm for deterministic underlying automata

and multiplicatively cancellative semirings to achieve $\mathcal{O}(m \log n)$ runtime.

Last but not least Wissmann et al. provide a generic method that minimizes various different automata types with a maximal runtime of $\mathcal{O}(m \log m \log n)$ (preprint uploaded on April 2nd, 2020) [53]. The reported runtimes are equivalent to the state-of-the-art methods for all types of automata.

4.2. Differential Equations

Unlike with automata, there are many points and aspects where one can start thinking about model reduction for differential equations. An example of a current approach in chemical reaction networks aggregates variables whose dynamics can still explicitly be described when aggregated and variables with equivalent dynamics [9]. Another approach is to separate fast from slow dynamical components, calling the fast components radical, making their dynamics immediate and thus forming a differential algebraic equation with only slow dynamics modelled as differentials while the fast ones are modelled by algebraic constraints. This method is called Quasi-Steady State Approximation (QSSA) [4]. Several concrete algorithms to perform improved versions of QSSA [3] and theoretically extended approximations, which diminish the error, have been developed [13]. A widely-used technique for reducing the complexity of differential equations is numerical solving, i.e. bounded-error approximation of computationally untractable DEs [10, 28, 34].

4.3. Combinations

An open question is how the two models relate and how the minimization or reduction in one model affects the properties in the other. [36] has shown that differential equation based fragments provide more comprehensive minimization techniques, is this always the case? Which of the models has advantages in reduction, which one in execution? Are the two equivalent or is there a special subtype unifying both frameworks, like (probabilistic) cellular automata [52, 32, 19]?

References

- [1] Steven Abney, David McAllester, and Fernando Pereira. "Relating probabilistic grammars and automata". In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. 1999, pp. 542–549.
- [2] Giovanni Bacci et al. "On the metric-based approximate minimization of Markov Chains". In: *Journal of Logical and Algebraic Methods in Programming* 100 (2018), pp. 36–56.
- [3] François Boulier et al. "Towards an automated reduction method for polynomial ODE models of biochemical reaction systems". In: *Mathematics in Computer Science* 2.3 (2009), pp. 443–464.
- [4] George Edward Briggs and John Burdon Sanderson Haldane. "A note on the kinetics of enzyme action". In: *Biochemical journal* 19.2 (1925), p. 338.
- [5] Janusz A Brzozowski. "Canonical regular expressions and minimal state graphs for definite events". In: *Mathematical theory of Automata* 12.6 (1962), pp. 529–561.
- [6] Peter Buchholz. "Bisimulation relations for weighted automata". In: *Theoretical Computer Science* 393.1-3 (2008), pp. 109–123.
- [7] Peter Buchholz. "EXACT AND ORDINARY LUMPABILITY IN FINITE MARKOV CHAINS". In: 1994.
- [8] RG Bukharaev. "Probabilistic automata". In: *Journal of Soviet Mathematics* 13.3 (1980), pp. 359–386.

- [9] Luca Cardelli et al. "Maximal aggregation of polynomial dynamical systems". In: *Proceedings of the National Academy of Sciences of the United States of America* 114 38 (2017), pp. 10029–10034.
- [10] Charles Francis Curtiss and Joseph O Hirschfelder. "Integration of stiff equations". In: *Proceedings of the National Academy of Sciences of the United States of America* 38.3 (1952), p. 235.
- [11] Pedro R DArgenio and Joost-Pieter Katoen. "A theory of stochastic systems part I: Stochastic automata". In: *Information and computation* 203.1 (2005), pp. 1–38.
- [12] Pedro R DArgenio and Joost-Pieter Katoen. "A theory of stochastic systems. Part II: Process algebra". In: *Information and Computation* 203.1 (2005), pp. 39–74.
- [13] Peter Deuflhard and Jörg Heroth. "Dynamic dimension reduction in ODE models". In: *Scientific computing in chemical engineering*. Springer, 1996, pp. 29–43.
- [14] Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of weighted automata*. Springer Science & Business Media, 2009.
- [15] Albert Einstein. "On the theory of the Brownian movement". In: *Ann. Phys* 19.4 (1906), pp. 371–381.
- [16] Kousha Etesami. "Analysis of probabilistic processes and automata theory". In: *Automata: from Mathematics to Applications* (2016).
- [17] Bernhard C Geiger et al. "Optimal kullback–leibler aggregation via information bottleneck". In: *IEEE Transactions on Automatic Control* 60.4 (2014), pp. 1010–1022.
- [18] Stuart Geman and Mark Johnson. "Probabilistic grammars and their applications". In: *International Encyclopedia of the Social & Behavioral Sciences* 2002 (2002), pp. 12075–12082.
- [19] G Grinstein, C Jayaprakash, and Yu He. "Statistical mechanics of probabilistic cellular automata". In: *Physical review letters* 55.23 (1985), p. 2527.
- [20] Jack K Hale et al. *Introduction to functional differential equations*. Vol. 99. Springer Science & Business Media, 1993.
- [21] Thomas Henzinger, Barbara Jobstmann, and Verena Wolf. "Formalisms for Specifying Markovian Population Models". In: vol. 22. Sept. 2009. DOI: 10.1007/978-3-642-04420-5_2.
- [22] John Hopcroft. "An $n \log n$ algorithm for minimizing states in a finite automaton". In: *Theory of machines and computations*. Elsevier, 1971, pp. 189–196.
- [23] Tao Jiang and Bala Ravikumar. "Minimal NFA problems are hard". In: *SIAM Journal on Computing* 22.6 (1993), pp. 1117–1141.
- [24] Tsunehiko Kameda and Peter Weiner. "On the state minimization of nondeterministic finite automata". In: *IEEE Transactions on Computers* 100.7 (1970), pp. 617–627.
- [25] Michael N Katehakis and Laurens C Smit. "A successive lumping procedure for a class of Markov chains". In: *Probability in the Engineering and Informational Sciences* 26.4 (2012), pp. 483–508.
- [26] Stefan Kiefer et al. "On the Complexity of Equivalence and Minimisation for Q-weighted Automata". In: *Logical Methods in Computer Science* 9 (2013).
- [27] Peter Kunkel and Volker Mehrmann. *Differential-algebraic equations: analysis and numerical solution*. Vol. 2. European Mathematical Society, 2006.
- [28] Wilhelm Kutta. "Beitrag zur naherungsweise integration totaler differentialgleichungen". In: *Z. Math. Phys.* 46 (1901), pp. 435–453.
- [29] Cristopher Moore and James P Crutchfield. "Quantum automata and quantum grammars". In: *Theoretical Computer Science* 237.1-2 (2000), pp. 275–306.

- [30] Isaac Newton. "Methodus fluxionum et serierum infinitarum". In: *Opuscula mathematica, philosophica et philologica* 1 (1774), p. 1774.
- [31] Bernt Øksendal. "Stochastic differential equations". In: *Stochastic differential equations*. Springer, 2003, pp. 65–84.
- [32] Stephan Olariu and Albert Y Zomaya. "Cellular Automata, PDEs, and Pattern Formation". In: *Handbook of Bioinspired Algorithms and Applications*. Chapman and Hall/CRC, 2005, pp. 291–302.
- [33] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.
- [34] Steven A Orszag. "Numerical methods for the simulation of turbulence". In: *The Physics of Fluids* 12.12 (1969), pp. II–250.
- [35] Azaria Paz. *Introduction to probabilistic automata*. Academic Press, 2014.
- [36] Tatjana Petrov. "Markov chain aggregation and its application to rule-based modelling". In: *CoRR* abs/1812.09774 (2018). arXiv: 1812.09774. URL: <http://arxiv.org/abs/1812.09774>.
- [37] Linda Petzold. "Differential/Algebraic Equations Are Not ODEs". In: *SIAM J. Sci. Stat. Comput.* 3.3 (Sept. 1982), pp. 367–384. ISSN: 0196-5204. DOI: 10.1137/0903023. URL: <https://doi.org/10.1137/0903023>.
- [38] Michael O Rabin. "Probabilistic automata". In: *Information and control* 6.3 (1963), pp. 230–245.
- [39] John Rhodes, Chrystopher L Nehaniv, and Morris W Hirsch. *Applications of automata theory and algebra: via the mathematical theory of complexity to biology, physics, psychology, philosophy, and games*. World Scientific, 2010.
- [40] Jean-Pierre Richard. "Time-delay systems: an overview of some recent advances and open problems". In: *Automatica* 39.10 (2003), pp. 1667–1694. ISSN: 0005-1098. DOI: [https://doi.org/10.1016/S0005-1098\(03\)00167-5](https://doi.org/10.1016/S0005-1098(03)00167-5). URL: <http://www.sciencedirect.com/science/article/pii/S0005109803001675>.
- [41] Uwe Schöning. *Theoretische Informatik-kurz gefasst*. BI-Wiss.-Verlag, 1992.
- [42] Marcel Paul Schützenberger. "On the definition of a family of automata". In: *Information and control* 4.2-3 (1961), pp. 245–270.
- [43] C. E. Shannon and J. McCarthy. *Automata Studies*. (AM-34). Vol. 34. Princeton University Press, 1956.
- [44] George F Simmons. *Differential equations with applications and historical notes*. CRC Press, 2016.
- [45] Ana Sokolova and Erik P De Vink. "Probabilistic automata: system types, parallel composition and comparison". In: *Validation of Stochastic Systems*. Springer, 2004, pp. 1–43.
- [46] Ana Sokolova and Erik P De Vink. "Probabilistic automata: system types, parallel composition and comparison". In: *Validation of Stochastic Systems*. Springer, 2004, pp. 1–43.
- [47] Mariëlle Stoelinga. "An introduction to probabilistic automata". In: *Bulletin of the EATCS* 78.176-198 (2002), p. 2.
- [48] Stephen Strogatz. "Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering (studies in nonlinearity)". In: (2001).
- [49] Antti Valmari and Giuliana Franceschinis. "Simple O (m log n) time Markov chain lumping". In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2010, pp. 38–52.
- [50] Vito Volterra. "Theory of functionals and of integral and integro-differential equations". In: (1959).

- [51] John Von Neumann. *Mathematical foundations of quantum mechanics: New edition*. Princeton university press, 2018.
- [52] John Von Neumann, Arthur W Burks, et al. "Theory of self-reproducing automata". In: *IEEE Transactions on Neural Networks* 5.1 (1966), pp. 3–14.
- [53] Thorsten WiSSmann et al. "From Generic Partition Refinement to Weighted Tree Automata Minimization". In: *arXiv preprint arXiv:2004.01250* (2020).
- [54] Daniel Zwillinger. *Handbook of differential equations*. Vol. 1. Gulf Professional Publishing, 1998.

A. Roadmap

| Time frame | Task | To be submitted |
|------------------|--|--|
| Until May 1st | Forms & registration, Requirements and Design | Software Requirements & Design specification, filled forms to M. Hesse |
| Until June 1st | Implementation of WA minimization | Source code with tests |
| Until July 1st | Implementation of Differential Equivalences | Source code with tests |
| Until July 15th | Benchmark: scalable examples of equivalent WAs & DEs | Benchmark specification incl. Proof of equivalence |
| Until August 1st | Final Report | Report + Source with tests + Benchmark specification |