

Minimization of Stochastic Dynamical Systems

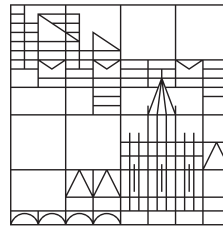
A Comparison of Models and Methods

M.Sc. Thesis Proposal, Draft v0.1

by

Fabian Klopfer

Universität
Konstanz



Faculty of Sciences

Department of Computer and Information Science

Modelling of Complex Self-Organizing Systems Group

Advisor: Prof. Dr. Tatjana Petrov

Konstanz, 2020

@Tanja/Stefano:

- What would be a more suitable title?
- Are introduction and problem statement okay like that?
- Wrt. Methods and literature: Is it okay as is or should I look for/add additional relevant literature?
- shall I add something on how the presentations and meetings are planned? I wouldnt add a roadmap as last time as my roadmap was completely off and overly optimistic
- Is the depiction of the models okay like this for the proposal? I'd include a more comprehensive/formal version in the thesis itself in the background section. Also include the MC as labeled version, WLTS, LTS, CRNs, CTMCs, formal power series, formal laurent series, ...
- Comments/Feedback?

1 Introduction

In the course of the project, two different approaches for minimizing dynamical systems in different modelling frameworks were considered:

On the one hand a method by Kiefer et al. [9], that uses the framework of weighted automata and is based on the work by Schützenberger[15]. It constructs the space of all words that can be reached from the starting state in n steps where n is the number of states. Those vectors are generated by multiplying the initial vector with the transition matrices corresponding to the considered word. Then a random vector of size $n \times |\text{Alphabet}|$ is generated and each word vector is multiplied by a random factor that is determined from the random vector and the word structure. The word vectors multiplied by the factors are then added up. This is done n times, each of these so generated n vectors is stacked as row vector into a matrix. Finally the initial vector is stacked atop of this matrix forming a $n + 1 \times n$ matrix. To complete the construction the maximally linearly independent subset of rows is chosen as basis of the prefix space. Then Schuetzenberger construction is applied using this matrix. Schuetzenberger construction uses the just constructed basis to transform the transition matrix, the initial and the final vector to a new lower-dimensional representation following the new basis. Thus the matrix constructed is some sort of projection from n to m dimensions with included reweighting and is thus a bisimulation following the definitions by Buchholz [3]. The method of Kiefer continues by applying the same steps using the postfix space instead of the prefix space.

This procedure has a worst case complexity that is exponential as the powerset of words with up to n letters has to be constructed: $|\text{Alphabet}|^n$. If we are assuming a sparse prefix space, that is only a few of all possible words are a possible prefix of the language accepted by the automaton, we get the average runtime. As Schuetzenberger construction uses matrix multiplication, which is in $\mathcal{O}(n^3)$ the algorithm also has an asymptotic average runtime of $\mathcal{O}(n^3)$. The algorithm is guaranteed to yield a minimal representation with respect to the number of states but produces often more non zero entries in the transition matrix.

On the other hand, a method by Tribastone et al.[5], that builds upon the algorithm of Valmari for Markov Chains[17] — thus also Paige and Tarjan[12] — extending it to the framework of chemical reaction networks (CRNs) or to be formally more strictly stochastic rewriting systems. The algorithm is actually targets polynomial ordinary differential equations (ODEs), but the authors provide a conversion from ODEs to CRNs. The algorithm then proceeds by calculating a quantity for each species and possible left-handside of the rule set and use this to split the species into partitions using the standard partition refinement approach that corresponds closely to [17]. The quantity to compute is the contribution by the authors and comes in two flavors: Forward differential equivalence uses stoichiometry per partition to find species that behave the same when “reacting” to other species, while backward differential equivalence tries to find species that behave

the same with respect to the reagents they are “reacted” from. According to the authors this shall be in correspondence with ordinary lumpability for forward differential equivalence (FDE) and exact lumpability for backward differential equivalence (BDE). While the first correspondence seems to be quite close, the latter assumes uniform initial values which is an aspect that weak lumpability handles, thus BDE seems to be closer related to weak lumpability. To state it in Buchholz’s terms [4]:

For exactly lumpable partitions exact results for individual probabilities of the original Markov chain can be computed for a special class of initial vectors, namely for all vectors for which the partition is weakly lumpable.

The procedure enjoys linearithmic average runtime complexity, resulting from using the partition refinement schema by Valmari et al [17]. This approach does not guarantee minimality, which is already shown for BDE by Boreale [1].

2 Problem Statement and Objectives of Study

The two methods above use different approaches to induce bisimulation relations: The first method uses linear algebra to find a lower dimensional basis, while the second method uses a stoichiometry-based quantity to find the coarsest partition with respect to the quantity.

This brings up the question whether the two are equivalent, thus yield the same representation with the same degree of reduction when the same dynamic system is input in the corresponding representation in the other framework. Do both compute the same kind of bisimulations and what are the guarantees that the two provide in terms of reduction.

In order to address this questions, one needs to convert a weighted automaton into a CRN and vice versa. To do this it seems most appropriate to approach the common ground of both methods: Probabilistic automata, whereof Markov chains are a subset with special semantics. The method of Tribastone et al. [5] is derived from Markov chains using the notions of lumping, the method of Kiefer addresses a superset of probabilistic automata, that is weighted automata. Essentially this boils the task down to find a conversion from CRNs to probabilistic automata and from weighted automata to probabilistic automata. Is this possible? Does this preserve important semantic properties like the language accepted by the automata, the execution semantics of both models and is it possible to assign a parametrized language to a CRN when adding a parametrized initial and boundary condition the other way round — when leaving the initial and final vector unbound as a parameter, is it possible to derive some rewriting system that induces a grammar when bound with concrete values? Is there some trick that allows to circumvent this parametrization e.g. by using the one vector as an initial vector or an auxiliary state and letter in the alphabet that with a parametrized transition matrix?

In the end, questions on which framework is more flexible, expressive, succinct, has better computational properties and covers certain language theoretic aspects arise. Similar ones arise for the reduction methods that are applicable and certain variants of them. If we find a method in one domain, is it straight forward to port it into the other?

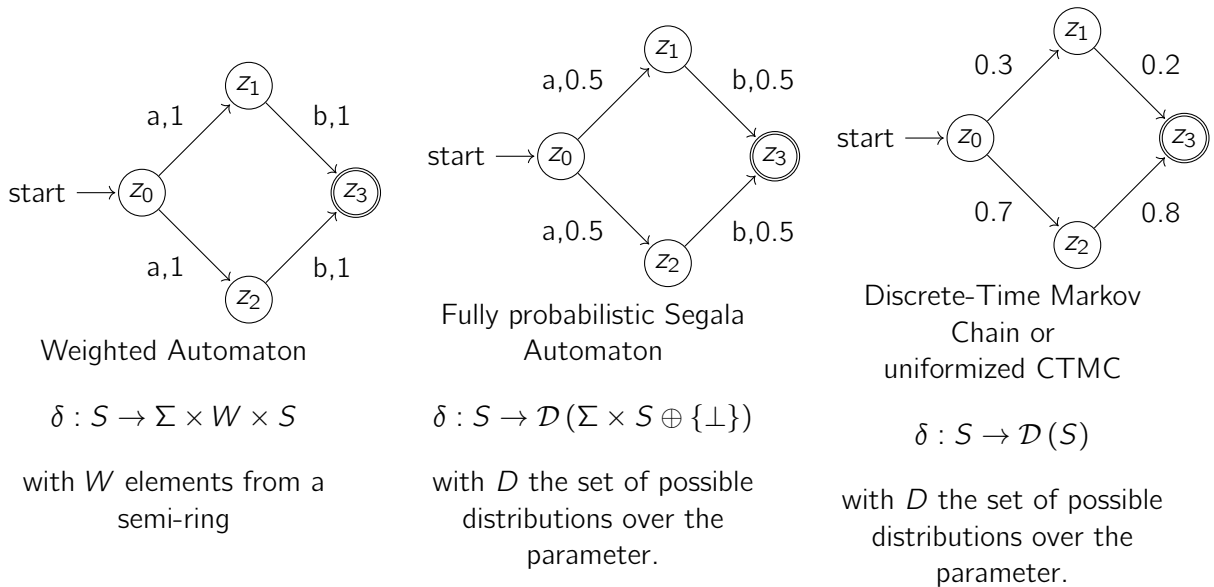
3 Methods and Literature

The two methods under consideration are described by the two concrete publications by Kiefer et al. [9] and Tribastone et al. [5] respectively. The methods used by the before mentioned authors were originally introduced by Schützenberger [15] and Paige & Tarjan [12].

There are plenty of other derived methods using these tools: Kiefer et al. provided a similar method that enables the basis to preserve the Kolmogorov axioms [10] of probability theory [8].

Buchholz describes lumpability, bisimulation and the connection to partitions [3, 4]. Feret et al. [6] generalize the notion of lumpability to weighted labeled transition systems. Larsen and Skou transfer the notion of bisimulation to probabilistic transition systems [11]. Partition refinement has also been used earlier by Hopcroft [7] for the minimization of finite automata. Partition-based approaches that act on markov chains acn also be found. Petrov [13] describes an algorithm to aggregate species for which the deterministic semantics of the thermodynamic limit is preserved in chemical reaction networks. Valmari and Franceschinis elaborate on an algorithm using partition refinement on markov chains. Rubino and Secola introduce an algorithm that computes weak bisimulations on markov chains [14]. Even though Brzozowski's method [2] seems to be related to backwards reduction introduced by Schützenberger, it uses rather a minimal partition construction approach, that is it constructs the automaton, such that there is no coarser partition.

When trying to connect both bisimulation methods, it is reasonable to restrict oneself to a concrete model that both other models — that is CRNs and WAs — shall be reduced to such that one is able to execute both algorithms on the representation used by the other each. We set this model to be the fully probabilistic Segala automaton [16]. An overview of the models can be found in the graphic below.



$A_{p,u} \xrightarrow{k_1} A_{u,u}$
 $A_{u,p} \xrightarrow{k_1} A_{u,u}$
 $A_{u,u} \xrightarrow{k_2} A_{p,u}$
 $A_{u,u} \xrightarrow{k_2} A_{u,p}$
 $A_{p,u} + B \xrightarrow{k_3} A_{p,u} B$
 Stochastic Rewrite System
 with rates k_i

$$\frac{dx_0}{dt} = x_0^3 + 2x_0 + x_0' - 3x_0'' + x_0x_1^2$$

$$\frac{dx_1}{dt} = x_1^5 + x_2 + x_1' - 3x_0''$$

$$\frac{dx_2}{dt} = 2x_0x_1x_2^2x_3^{-1}$$

$$\frac{dx_4}{dt} = x_3''' + x_0' + x_1'' + x_2^{(4)}$$

System of Polynomial Ordinary Differential Equations, in general:

$$\frac{dx_i}{dt} = f(x_0, \dots, x_n, x_0', \dots, x_n', \dots)$$

with f a ploynomial.

References

- [1] Michele Boreale. “Algebra, coalgebra, and minimization in polynomial differential equations”. In: *International Conference on Foundations of Software Science and Computation Structures*. Springer. 2017, pp. 71–87.
- [2] Janusz A Brzozowski. “Canonical regular expressions and minimal state graphs for definite events”. In: *Mathematical theory of Automata* 12.6 (1962), pp. 529–561.
- [3] Peter Buchholz. “Bisimulation relations for weighted automata”. In: *Theoretical Computer Science* 393.1-3 (2008), pp. 109–123.
- [4] Peter Buchholz. “Exact and ordinary lumpability in finite Markov chains”. In: 1994.
- [5] 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.
- [6] Jerome Feret et al. “Lumpability abstractions of rule-based systems”. In: *Theoretical Computer Science* 431 (2012), pp. 137–164.
- [7] 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.
- [8] Stefan Kiefer and Björn Wachter. “Stability and complexity of minimising probabilistic automata”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2014, pp. 268–279.
- [9] Stefan Kiefer et al. “On the Complexity of Equivalence and Minimisation for Q-weighted Automata”. In: *Logical Methods in Computer Science* 9 (2013).
- [10] A Kolomogoroff. *Grundbegriffe der wahrscheinlichkeitsrechnung*. Springer-Verlag, 1933.
- [11] Kim G Larsen and Arne Skou. “Bisimulation through probabilistic testing”. In: *Information and computation* 94.1 (1991), pp. 1–28.
- [12] Robert Paige and Robert E. Tarjan. “Three Partition Refinement Algorithms”. In: *SIAM Journal on Computing* 16.6 (1987), pp. 973–989. DOI: 10.1137/0216062. eprint: <https://doi.org/10.1137/0216062>. URL: <https://doi.org/10.1137/0216062>.
- [13] Tatjana Petrov. “Markov chain aggregation and its application to rule-based modelling”. In: *Modeling Biomolecular Site Dynamics*. Springer, 2019, pp. 297–313.
- [14] Gerardo Rubino and Bruno Sericola. “On weak lumpability in Markov chains”. In: *Journal of Applied Probability* (1989), pp. 446–457.
- [15] Marcel Paul Schützenberger. “On the definition of a family of automata”. In: *Information and control* 4.2-3 (1961), pp. 245–270.
- [16] Roberto Segala and Nancy Lynch. “Probabilistic simulations for probabilistic processes”. In: *Nordic Journal of Computing* 2.2 (1995), pp. 250–273.
- [17] 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.