Faculty of Informatics
Masaryk University

Formal Methods for Analysis of Biological Systems under Parameter Uncertainty

David Šafránek

Habilitation Thesis

Brno, November 12, 2018

Abstract

In this habilitation thesis, we discuss a set of novel formal methods for computer-aided analysis of mathematical and computational models of biological systems. The presented techniques target two problems that have recently appeared in the field of computational systems biology — namely, the problem of *parameter synthesis* and the problem of *robustness analysis*. The techniques described in the thesis target both of the problems. Besides systems biology, the techniques presented in the thesis are applicable in the emerging domains of cyber-physical and cyber-biological systems.

The shared attribute of all techniques presented in the thesis is the fact they are based on rigorous formal methods, namely, on model checking algorithms developed for formal verification of computer systems. This attribute signifies the uniqueness of the techniques and positions them primarily in the state-of-the-art *in silico* toolchains of synthetic biology workflows where engineering and design of synthetically re-programmed living cells are performed.

Both discussed problems address the challenging need for efficient global analysis of computational models frequently used in systems and synthetic biology. Model classes targeted by the techniques described in the thesis represent the majority of models that can be encoded in SBML level 3, the community standard for model-based systems biology. In particular, we address methods for deterministic continuous-time models based on differential equations that for historical reasons still represent the most frequently used formalism. Next, we discuss methods working with continuous-time stochastic models that are used to describe biological systems and molecular mechanisms observed at the detailed level of individual discrete events of molecular interactions. Finally, we discuss methods working with abstract discrete models (Boolean or Thomas' Networks) representing the logic of positive and negative influences among molecules. These formalisms are used to describe biological systems at a high level of abstraction and they make an important tool for inference of new hypotheses and design of targeted wet-lab experiments. Most of the discussed methods are demonstrated in several case studies.

The thesis is based on extended versions of multiple conference and journal papers joint into a unified framework and accompanied with a significantly extended overview of other existing approaches.

Acknowledgement

First of all, I would like to thank master's and doctoral students in close cooperation with whom I obtained the results presented in this thesis, namely, Martin Demko, Petr Dluhoš, Sven Dražan, Jana Dražanová, Juraj Kolčák, Matej Hajnal, Samuel Pastva, Adam Streck, Matej Troják, Martin Vejnár, and Tomáš Vejpustek.

I would like to thank very much prof. Luboš Brim, the head of the Systems Biology Laboratory which I am a member of at Faculty of Informatics, Masaryk University, for his continuous moral and professional support.

I would like to thank my colleagues in close cooperation with whom I obtained the results presented in this thesis. Namely, the colleagues from Masaryk University – Jiří Barnat, Nikola Beneš, Milan Češka, Ivana Černá, Jiří Damborský, Pavel Krejčí, the colleagues from Free University Berlin – Hannes Klarner and Heike Siebert, and the colleagues from Cachan and Université Paris-Saclay – Loïc Paulevé and Stefan Haar.

Additionally, I appreciate very much the discussions and support coming from prof. Marta Kwiatkowska (University of Oxford), prof. Jan Van Schuppen (University of Delft), prof. Jean-Marie Jacquet (University of Namur), Alessandro Abate (University of Oxford), Ezio Bartocci (TU Vienna), Luca Bortolussi (University of Trieste), Luca Cardelli (Microsoft Research), Jan Červený (CzechGlobe), and Ralf Steuer (Humboldt University Berlin). I would like to thank Oded Maler (who unfortunately passed away in September 2018) for great discussions forming my motivation and approach to interdisciplinary research.

Finally, I am deeply indebted to my family for their great moral support and patience without which I would have never finished this thesis.

The work presented in this thesis has been supported by a number of grant projects — in particular, European 6th Framework STREP NEST project EC-MOAN Reg. No. 043235, the Czech Grant Agency projects GA15-11089S and GA18-00178S, the EU OP project CyanoTeam (Reg. No. CZ.1.07/2.3.00 /20.0256), and the MEYS project of the National infrastructure C4SYS - Centre for Systems Biology (Reg. No. LM2015055).

Contents

Chapter 1

Introduction

All biological systems, from single pathways to multicellular organisms, can be seen as complex systems of interacting components. Biological systems can also be seen as reactive systems, as they continuously interact with their environment. Systems biology thus studies complex *interactions* in biological systems, with the aim to understand better the processes that happen in such a system, as well as to grasp the emergent properties of such a system as a whole [241].

The paradigm of systems biology is based on the model-based approach to study the complex phenomena. The model approximately represents a certain phenomenon under given biological and physical assumptions. While the biological knowledge is always uncertain and incomplete, models represent rigorous structures that can be analysed by computational methods. The emphasis on computational methods and models refines the general paradigm of systems biology to a more concrete scheme of *computational systems biology* [240].

Computational systems biology can, by drawing upon mathematical approaches developed in the context of computer science and engineering [173, 309], contribute to the creation of powerful simulation, analysis and reasoning tools for working biologists. These tools can be used in designing and devising new experiments and ultimately, for understanding functional properties of a genome, proteome, cells, and organisms.

In last years we are continuously experiencing growing collaboration between biologists and computer scientists in many areas of systems and synthetic biology. This is because it has turned out that formal mathematical approaches to modelling and analysis that have been developed for parallel and distributed computer systems and are referred to as *formal methods* are applicable to biological systems as well as both kinds of systems have a lot in common. In particular, techniques developed in computer science for automated formal verification have the potential to be exploited in computational systems biology. Of special interest is model checking that gives a feasible methodology to verify/refute interesting biological hypotheses [84].

1.1 Motivation

The computational analysis of the precise dynamics of biochemical systems involves the construction of appropriate *computational models*. Building suitable sound dynamic models can be seen as a key step toward the development of predictive models for cells or whole organisms. While the structure of such models is usually available, some of the quantitative features of models cannot be easily determined. These quantitative attributes, which significantly affect the system dynamics, are usually reflected in the model as *parameters*.

In order to obtain reliable models, parameters, such as reaction rates or concentration values, need to be specified exactly. For a typical model, some of the parameters values can be determined from the literature or experimental data, many parameters values are uncertain or unknown. The reason is that measurement of many parameters *in vitro* or even *in vivo* is hardly possible. In particular, the level of detail at which parameter values can be measured is insufficient in comparison to the level of detail needed for a precise model. An example is measuring protein production rates, which vary with different regulation modes arising during protein transcription. Another problem is that the models have to be specified at a certain level of abstraction approximately representing the modelled phenomenon. This means that a parameter in a model can abstractly represent a value that is composed in a non-trivial way from several physical parameters describing detailed physical or biophysical characteristics of a modelled phenomenon.

The algorithmic analysis of dynamics generated by models with unknown parameter values is one of the main challenges in computational systems biology [284, 293]. The problem is addressed from several perspectives. On the one end, there exist data-oriented approaches of *parameter estimation* based on finding parameter values that optimally fit the given time-series data [288] typically employing randomised algorithms and techniques of artificial intelligence and machine learning. In systems control theory, the problem is known as *parameter identification* that includes the rigorous problem of systems identification from data [189] or the problem augmenting parameter estimation with the inference of statistical guarantees [314]. On the other end, there are behaviour-oriented and property-oriented approaches that target the global problem of finding the parameter values exhibiting a specified behaviour. This problem is targeted by methods of *parameter synthesis* or *parameter scanning* [147, 38] based on finding parameter values reproducing a given temporal phenomenon, or methods of *bifurcation analysis* [104, 58, 230], focused on finding parameter values that significantly change the behaviour of the system.

A related problem to parameter synthesis has been raised by Kitano in [242]. It addresses the so-called *robustness analysis* taking into account the fact that various biological systems can display a certain phenomenon for parameter values sets of significantly different cardinality. Intuitively, the larger is the set of parameter values exhibiting a given feature the more robust the feature is. The metrics provided to measure robustness thus give an instrument that allows us to compare models with respect to a given property and a given range of parameter uncertainty. Since the notion is relatively novel, there are not yet efficient techniques providing a fully automatised algorithmic support for robustness analysis.

1.2 Lessons Learned in Systems Biology Projects

In the application-oriented part of our research, we have found a strong evidence for the significance of the problems mentioned above. In the following we describe our insights taken from the research projects in which we have been involved.

The project EC-MOAN[1] targeted several gaps in understanding interactions between the metabolic and genetic layer of *Escherichia Coli*. The formal methods were utilised to help biologists to find rigorous explanation of several considered hypotheses. Model checking techniques have been pioneered to be used for analysis of biological models by employing methods known from control theory. Gaps between biologists and mathematicians/computer scientists were narrowed but not yet sufficiently. The need for methods working under parameter uncertainty has been formulated and pioneered by two groups involved in the project [51, 38].

In the project CyanoTeam/CyanoNetwork[2] we targeted new challenges of systems biology of cyanobacteria and photosynthesis. Owing to the fact cyanobacteria is not that well studied as *E. coli*, several new challenges have appeared. In particular, we have formulated a general framework for modelling and analysis of biological processes making the logical back-end for the comprehensive modelling web tools targeting photosynthesis [361] and cyanobacteria [245]. The platforms utilise well-established analysis techniques based on simulation. An important challenge is to bring techniques working with uncertain parameters to these platforms. A necessary step is to develop techniques that can be in future deployed to the platforms in the form of a push-button technology.

Lessons that are continuously being learned from ongoing national collaborations[3] with local laboratories show the crucial importance of model-based workflow in systems and synthetic biology. Especially, development of methods targeting parameter synthesis with formal guarantees can sig-

[1] http://www.ec-moan.org, 6th Framework Programme FP6-2005-NEST-PATH-COM

[2] http://www.cyanoteam.org, EC OP project Reg. No. CZ.1.07/2.3.00/20.0256

[3] http://c4sys.cz, The national infrastructure C4SYS - Centre for Systems Biology, MEYS project No. LM2015055

nificantly reduce the efforts needed to design synthetically modified cells (e.g., the research reported in [139] and described in Section 7.2).

1.3 Focus of the Thesis

The research goal targeted in this thesis is the development of efficient computer-scientific techniques that allow automatised analysis of models of biochemical systems under parameter uncertainty. Our main contribution is in narrowing the existing gaps in the current state-of-the-art technology. In particular, this is realised by introducing new original methods and techniques that utilise model checking as the central method allowing to analyse and explore models with unknown parameters with respect to a given global property (hypothesis). In particular, we target the problems of *parameter synthesis* and *robustness analysis* from the perspective of formal methods based on model checking. The results computed by these methods are parameter values that evaluate the global property with some formal guarantees.

We formulate a concrete framework supporting a specific set of model classes frequently used in systems biology. Our study is then focused to these classes. Existing state-of-the-art techniques targeting the research goal are briefly described. Under this setting, we identify existing critical gaps. Next, we formulate the problems to be investigated and solved formally and we describe technical solutions to these problems. The solutions described in the thesis are based on approaches that represent our own contribution to the field. The impact of the contributed methods is demonstrated on several biological problems with various levels of complexity.

Owing to the fact that the individual approaches have been published in separate research articles, we give their comprehensive overview unified in an abstract framework. The description of our methods is accompanied with a significantly extended overview of related approaches that are discussed and compared with our methods.

It is worth noting that this thesis focuses primarily on the problems of parameter synthesis and robustness analysis. Methods practically used in a wider context of inverse problems [160] for fitting the models to experimental data are not discussed here. A comprehensive review of those methods is provided, e.g., in [314]. A critical aspect of those methods is that the models are typically over-parameterised resulting in situations when parameters are poorly constrained by experimental data [207] thus making a good estimation of a single reliable parameter valuation impossible. Our approach avoids such situations by reformulating the inverse problem in an abstract way. In particular, parameter synthesis is a method that *a priori* returns a set of parameter valuations satisfying a general property instead of searching for a single optimal value fitting a given measurement.

1.4 Thesis Structure

The structure of the thesis is the following. Chapter 2 fixes the modelling framework and discusses state-of-the-art techniques available for this framework. Additionally, a family of temporal properties used to capture biological phenomena is discussed with a special emphasis on logics used in our methods. Chapter 3 states formally the problems considered in the remaining part of the thesis. Most important relations to related work directly applicable to the stated problems are discussed here. Additionally, our contribution to the stated problems is briefly summarised. Chapters 4-6 are devoted to a detailed description of the techniques used to solve the problems given in Chapter 3. Finally, case studies performed on several biological problems are described in detail in Chapter 7.

Chapter 2

Background

2.1 Computational Systems Biology

Systems biology can be characterised as an approach to the understanding of life through the study of how the properties of biological systems arise through interactions between the system components [283]. In 2002, the specialised field of computational systems biology has been introduced by Kitano [240]. In general, it is a scientific paradigm that addresses questions fundamental to our understanding of life by means of rigorous methods. Progress in this field has direct impact to practical innovations in medicine, drug discovery and engineering. An important attribute is that processes in living matter are studied as dynamical systems and systems engineering methodology is applied. The methodology is based on mathematical models used to rigorously represent biological phenomena in abstract forms.

2.1.1 Biological Systems

Biological systems are complex systems of interacting biological components, which may be molecules, cells, organisms or entire species, that change their properties with time in response to external and internal stimuli. Studying the *dynamic behaviour* of these systems is the basis for understanding of cellular functions or disease mechanisms.

The formalism used to model a biological system is essential, since it not only dictates the possible behaviour that may or may not be captured, but also determines the computational means of detecting and analysing that. Fisher and Henzinger [173] distinguish two kinds of models relevant for computational systems biology: *executable* versus denotational (or, as they phrase it, *computational* versus mathematical).

2.1.2 Mathematical Approach

Mathematical models in systems biology are typically represented by means of ordinary differential equations (ODEs) that have a long tradition in chemistry since 1864 [206, 285]. They are used to approximate the behaviour of well-mixed (bio)chemical masses deterministically by continuous functions of time. They are frequently used by biophysicists and therefore make the main formal language of computational systems biology. To share this type of models, the format SBML [228] has been developed. It is derived from XML and provides a way to prepare the models for computational processing. Commonly known repositories support this approach, probably the best known of these is Biomodels.org [265].

The most common method for analysis of mathematical models of dynamical systems is the numerical simulation. There are several tools developed especially for systems biology, e.g., Copasi [224], CellDesigner [182], CellML [271], etc. They support simulation and other numerical tasks. These tasks require detailed knowledge of the biological system, i.e., *quantitative parameters* identifying the physical aspects of system interactions (e.g., kinetic coefficients of chemical reactions). Therefore, of special interest is *parameter estimation* [288], a method that combines simulation with optimisation to find values of unknown parameters with respect to experimental data. It is a key issue in systems biology, as it represents the crucial step to obtaining better models of biological systems that give more precise predictions. In biological models, the control parameters used to define the behaviour of models are kinetic or rate parameters. Some of these parameters usually cannot be experimentally determined which leads to the need to estimate these parameters by computational methods. Due to the non-linear character of mathematical models in biology, these methods are criticised by mathematicians because of inexactness and ill-formed settings lacking precise discussions of identifiability. Some of these drawbacks can be avoided by using advanced statistical methods [314].

2.1.3 Computational Approach

Computational models form the new promising basis for studying various biological phenomena by drawing upon formal approaches developed in the context of computer science and engineering [173, 309, 229, 172, 362]. This is because it has turned out that formal approaches to modelling and analysis, referred to as *formal methods*, are applicable to biological systems as well, as both kinds of systems have a lot in common, especially, discrete event dynamics, concurrency and reactiveness. Moreover, in both kinds of systems the interaction of components is a source of various *emergent system properties* that are not explicitly encoded in individual system parts.

There are two main properties of computational models that cannot be, in general, easily obtained in mathematical models. First, operational concurrent discrete-event semantics of computational models gives an adequate description of reactive events occurring among molecules and other particles. Moreover, operational view allows to study *how* the complex behaviour emerges from individual interactions. Second, computational models can represent the system at *arbitrary level of abstraction* (from detailed reactive event view to higher-level events abstractly representing complex processes). Moreover, models can be written in modular and compositional form that allows possibility to *integrate* modules at different level of abstraction. Integration of partial models is one of the crucial tasks of modern systems biology that aims to explain the behaviour in the global context. This is difficult to be achieved with mathematical models.

In last years, many techniques and tools employing computational models and formal methods have been developed resulting with strong case studies. Examples of widely used formalisms are Boolean networks [349, 106, 244], Petri Nets [215, 70, 105, 307], timed automata [53, 333, 199], compact process algebraic representations such as BioPEPA [114], Kappa [130], BNGL [161] or suitable adaptations of π-calculus [304, 316]. Overview of methods is given, e.g., in [45, 174].

2.2 Biological Networks

The theory and practise of systems biology are inherently based on the concept of studying biological systems as networks of interacting species. Biological knowledge is reconstructed and organised in the form of *biological networks*. Biological networks are built from biological knowledge databases, experimental data and biophysical principles employing many simplifying assumptions. There are two fundamental types of biological networks – *reaction networks* and *influence networks*. Recent complex network reconstructions typically mix the two, we call such kinds of biological networks as *reaction-influence networks*.

Example 2.1 *An example of a general biological network is given in Figure 2.1. The network is described in the standard visual notation SBGN (Systems Biology Graphical Notation) [264] in terms of a so-called process diagram. The network describes a simplified case of transcription of a gene (*geneX*) controlled by a particular protein* TF *(a so-called transcription factor). The outcome of the transcription process is a molecule of messenger mRNA (*rnaX*).*

The auxiliary species denoted $\varnothing$ represents the general matter out of the border of the modelled system.

In [99] the two kinds of networks (reaction and influence networks) are studied and compared by using the utility of graph morphisms. It is shown that under given fixed assumptions the reaction networks make a more

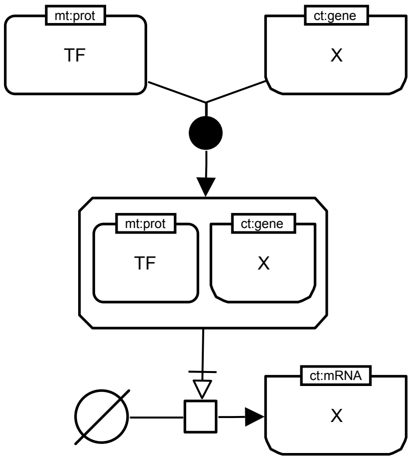


Figure 2.1: An example of a biological network having the form of a reaction-influence network. The example is shown in the standard graphical format SBGN.

concrete forms (implementations) of influence networks. In particular, influence networks can be, from the graph-theoretic perspective, understood as abstractions of reaction networks. The abstraction relation is formalised by means of morphisms between networks. These morphisms reflect (and preserve) the kinetic interpretation of biological systems described by the networks.

The information encoded in biological networks can be used to study relations between individual networks topologies existing in nature. E.g., in [191, 190] a fully structural approach was proposed to measure the distance of biological networks obtained by changing the graph structure. The natural development of biological networks structure is hypothesised to follow several optimisation goals set with respect to the environment in which the living cell resides in long-time horizon [235].

2.2.1 Chemical Reaction Networks

Chemical reaction networks (reaction networks for short) provide a detailed mechanistic view of biochemical systems – nodes are (bio)chemical species and stoichiometry-labelled (multi-)edges represent elementary (bio)chemical reactions. Chemical reaction networks are used as the basic formalism in chemistry. On the theoretical side, they were studied mainly from algebraic or graph-theoretic perspectives [169, 354, 3, 233, 121, 321] allowing to infer properties of network dynamics from network structure.

For the purpose of using reaction networks in this thesis, we modify and extend the notation and definition introduced in [100]. In general, chemical reaction networks make the basic formalism for describing biological systems at the elementary mechanistic level. They also form the basis of SBGN process diagrams.

Notation 2.2 *Denote $\mathfrak{L}$ the universe of all* chemical species. *We assume $\mathfrak{L}$ to be lexicographically ordered.*

Definition 2.3 Chemical reaction network (CRN) *(or* reaction network *for short) is a pair $(\Lambda, \mathcal{R})$ where $\Lambda \subset \mathfrak{L}$ is a finite set of* species *and $\mathcal{R}$ is a finite set of* reactions *such that $\varrho \in \mathcal{R}$ is a pair $\varrho = (r_\varrho, p_\varrho)$ where $r_\varrho \in \mathbb{N}_0^{|\Lambda|}$ is the* reactant complex *and $p_\varrho \in \mathbb{N}_0^{|\Lambda|}$ is the* product complex *representing the stoichiometry of reactants and products, respectively. In both cases, we assume the components of a complex follow the lexicographical order.*

For any $l \in \Lambda$, notation $r_\varrho(l)$ is used to denote the stoichiometry coefficient corresponding to l in the reactant complex component r_ϱ.

Example 2.4 *An example of a simple reaction network with interesting non-linear behaviour is a so-called Schloegel's reaction system representing a set of ordinary reactions acting on a single chemical species [326]:*

$$\varrho_1 : \ 2A \to 3A \qquad \varrho_2 : \ 3A \to 2A \qquad \varrho_3 : \ \varnothing \to A \qquad \varrho_4 : \ A \to \varnothing$$

Encoding in terms of Definition 2.3 is the following (since the complexes include at most a single component the respective brackets are omitted):

$$(\{A\}, \{\underbrace{(2,3)}_{\varrho_1}, \underbrace{(3,2)}_{\varrho_2}, \underbrace{(0,1)}_{\varrho_3}, \underbrace{(1,0)}_{\varrho_4}\})$$

Example 2.5 *The following network is a simple example of a reversible catalytic reaction system where E_f represents an enzyme catalysing the forward reaction, E_b is an enzyme catalysing the backward reaction, S is a substrate, and P a final product.*

$$\begin{array}{lll} \varrho_1 : \ S + E_f \to SE_f & \varrho_2 : \ SE_f \to S + E_f & \varrho_3 : \ SE_f \to P + E_f \\ \varrho_4 : \ P + E_b \to PE_b & \varrho_5 : \ PE_b \to P + E_b & \varrho_6 : \ PE_b \to S + E_b \end{array}$$

Encoding reflecting Definition 2.3 is the following (the set of species is considered to follow the displayed lexicographic order):

$$(\{E_b, E_f, S, SE_f, P, PE_b\}, \{\overbrace{((0,1,1,0,0,0),(0,0,0,1,0,0))}^{\varrho_1}, \overbrace{((0,0,0,1,0,0),(0,1,1,0,0,0))}^{\varrho_2},$$
$$\underbrace{((0,0,0,1,0,0),(0,1,0,0,1,0))}_{\varrho_3}, \underbrace{((1,0,0,0,1,0),(0,0,0,0,0,1))}_{\varrho_4}, \underbrace{((0,0,0,0,0,1),(1,0,0,0,1,0))}_{\varrho_5},$$
$$\underbrace{((0,0,0,0,0,1),(1,0,1,0,0,0))}_{\varrho_6}\})$$

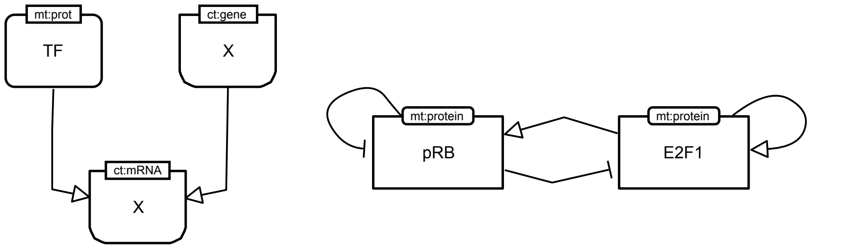


Figure 2.2: (left) The influence graph abstractly representing the biological network from Figure 2.1. (right) A two-node influence network representing a variant of a genetic switch.

2.2.2 Influence Networks

Influence networks describe systems at a higher level of abstraction than CRNs by focusing on influences among individual system components – nodes are biochemical species or abstract biological objects and edges represent positive or negative influences. Gene regulatory networks [134] or signalling pathways [239] make typical examples used in systems biology. Influence networks are frequently used in systems biology to describe processes without giving mechanistic details on the individual (bio)chemical interactions. They make also a part of the graphical format SBGN in the form of so-called activity flow diagrams. We present the following compact definition of influence networks that we have introduced in [248].

Definition 2.6 Influence network (IN) *is a pair* $(\Lambda, \Gamma_\Lambda)$ *where* $\Lambda \subset \mathfrak{L}$ *is a finite set of* species *and* Γ_Λ *is a finite set of* species influences *such that* $\Gamma_\Lambda \subseteq \Lambda \times \Lambda$. *For each* $l \in \Lambda$ *we denote the set of its* influencers *as* $n^{-}(l) = \{x \in \Lambda \mid (x, l) \in \Gamma_\Lambda\}$.

Example 2.7 *A simple example of an influence network is depicted in Figure 2.2 (left). It represents an influence abstraction of the biological network shown in Figure 2.1. It is presented in the SBGN format. The presence of protein* TF *and gene* X *has a positive influence on the presence of the respective mRNA molecule encoding a protein* X. *Note that in our framework of INs we do not capture the information about the type of the influence, only the topology is represented.*

It is worth noting that we have decided to omit specification of the influence type at the level of INs because there are many different types of individual influences that have various semantics which is typically not clearly fixed (i.e., the specification of SBGN gives a pallet of arrows representing an influence at a very abstract level without specifying the respective mechanistic function underlying the influence). Detailed specification of influence functions is moved to the level of semantics.

Example 2.8 *Another influence network is depicted in Figure 2.2 (right). The network consists of two protein species found in human cells. These proteins are mutually influencing each other in a negative and positive way indirectly through controlling the transcription of the influenced (regulated) protein. Each of the proteins additionally self-regulates transcription of its own. This is caused by a negative (pRB) and a positive (E2F1) feedback. This genetic circuit (called a genetic switch) is important in controlling the irreversible decision of a cell to proceed with initiating the doubling procedure [345]. In Section 7.3 we provide a detailed study of this circuit.*

2.2.3 Reaction-Influence Networks

CRNs and INs introduced in previous sections make a basis for formal specification of biological systems. CRNs provide a framework for detailed mechanistic description of biochemistry underlying biological processes whereas INs allow describing complex processes at an abstract level avoiding mechanistic details. As it has been discussed in Example 2.1, a mixture of both kinds of networks is typically used in practice. In typical modelling scenarios, not all modelled processes can be decomposed down to the mechanistic level. The reasons are the enormous complexity and interconnection of biological processes and the large extent of uncertainty in biological knowledge. A feasible solution is to use influences in cases where simplifying descriptions are needed (or sufficient) and to combine them with reactions to describe well-known parts of modelled processes. To that end, reaction-influence networks (RINs) are formally established in the following definition.

Definition 2.9 *Let $(\Lambda, \mathcal{R})$ be a reaction network. We define a* reaction-influence network *as a triple $(\Lambda, \mathcal{R}, \Gamma_{\mathcal{R}})$ where $\Gamma_{\mathcal{R}} \subseteq \Lambda \times \mathcal{R}$ is a set of* reaction influences. *For each $\varrho \in \mathcal{R}$ we define a set of its* influencers *as $n^-(\varrho) = \{x \in \Lambda \mid (x, \varrho) \in \Gamma_{\mathcal{R}}\}$. Additionally, we denote the* set of influenced reactions *$\mathcal{R}^- = \{\varrho \in \mathcal{R} \mid n^-(\varrho) \neq \emptyset\}$ and the* set of uninfluenced reactions *$\mathcal{R}^{\emptyset} = \mathcal{R} \setminus \mathcal{R}^-$.*

Remark 2.10 *Definition 2.9 is compatible with the notion of biological model structure as understood in the standard format Systems Biology Markup Language (SBML) level 2 [228]. In SBML, the set of species is represented as a* list of species *and the set of reactions as a* list of reactions. *Reactions influencers are called* modifiers.

Example 2.11 *An example of a biological network that can be represented in terms of a RIN is given in Figure 2.1. The network has four species, $\Lambda = \{geneX, geneX\text{-}TF, rnaX, TF\}$. The basic CRN has the following two ordinary chemical reactions displayed with a respective encoding reflecting the lexicographic order of Λ:*

$$\begin{array}{lll} \varrho_1: & geneX + TF \to geneX\text{-}TF & ((1,0,0,1),(0,1,0,0)) \in \mathcal{R} \\ \varrho_2: & \varnothing \to rnaX & ((0,0,0,0),(0,0,1,0)) \in \mathcal{R} \end{array}$$

First, there is a second-order reaction ϱ_1 that describes a complexation process (association). A molecule of the transcription factor (TF) binds to the particular segment of DNA (the gene X). Second reaction ϱ_2, describes formation of an RNA molecule. This first-order reaction is described in a very abstract form without chemical details (i.e., small molecules such as ATP or transcription enhancers involved in the process are not specified). All elements making the matter from which RNA is built are set outside of the system.

There is a single influence in the network, $\Gamma_{\mathcal{R}} = \{(geneX\text{-}TF, \varrho_2)\}$, representing a positive effect of the complex $geneX\text{-}TF$ on the occurrence of the second reaction. The influence abstracts from many elementary reactions describing the details of chemical interactions among all involved molecules.

2.3 Models of Biological Networks

One of the main aims of computational systems biology is to study the dynamics of biological networks, in particular, how a population of components affected by network interactions evolves in time. To that end, a biological network is extended to a *biological model* that associates the network structure with a suitable semantics reflecting the dynamics of network components understood at a particular level of abstraction. The semantics fulfils the following tasks:

- Network components (species) are given a mathematical interpretation as *variables* (numbers of molecules, molar concentrations, or a suitable abstract interpretation of these quantities),

- Network interactions (reactions or influences) are given a mathematical interpretation as *rules* describing the dynamics – changes in affected variables occurring in discrete or continuous time.

Discrete-value semantics can capture either quantitative (microscopic or mesoscopic [317]) interpretation of chemical/biological entities (e.g., number of particles, atoms, molecules) or abstract qualitative interpretations of selected qualities of modelled components (e.g., absence/presence of a molecule, several levels/intervals abstracting quantitative values). *Continuous-value* semantics represents a so-called macroscopic view [231] where the modelled components are expected to appear in large quantities provided that it is inconvenient to distinguish small differences (e.g., the molar concentration of a species in a cell or in a certain organelle).

Rules are distinguished at several levels. The most basic form of a rule is a chemical reaction. It makes the central organisation-building element in reaction networks (Definition 2.3). Elementary reactions representing mass transfer ($A \rightarrow B$), dissociation ($AB \rightarrow A + B$), and compound formation ($A + B \rightarrow AB$) represent the atomic rules of chemistry/biology.

However, modelling of biological processes at that level requires very detailed knowledge that is often not completely known (e.g. detailed schemes of enzymatic reactions or reactions in gene regulation). Moreover, such processes suffer from a large extent of combinatorial explosion due to the fact the molecules and compounds can exist in many different concurrent states (e.g., phosporylation, methylation, etc.). To that end, rules in models are used with suitable levels of abstraction and compactness. One of such abstractions, in the form of general chemical reactions, has been already considered in Definition 2.3 (non-trivial stoichiometry coefficients allowed in reactant and product complexes). Another level of abstraction that avoids combinatorial explosion by context-free partial specification of reactants and products forms, is provided in rule-based [161, 130, 223, 302] and to some extent also in process-algebraic languages [310, 64].

Quantities that can be associated with rules are time and probability. Since each interaction occurs in time with a specific *rate*, the respective rule is executed with this rate implying the inherent time aspect of the system dynamics. Naturally, time is considered as continuous, dense quantity. When the information on rate is unknown or abstracted out due to simplifications, *discrete-time* semantics is employed. It deals with the shortest (discrete) time step which can represent an arbitrary finite time horizon. Discrete-time abstraction allows to treat computational models as *untimed*, i.e., the exact duration of a single time step is left unspecified. It is worth noting that in the most of cases the occurrence of any rule is modelled as instantaneous and it occurs immediately after the conditions for occurrence are satisfied. There exist models that refine these aspects of semantics (e.g., delayed interactions [97] or non-instantaneous interactions [29, 108]).

With respect to the execution of interactions, rules can be either *deterministic* or *stochastic*. Deterministic rules represent interaction events that occur each time all preconditions are satisfied (i.e., if the given system state contains at least the required amount of all reactants, the reaction is enabled and occurs). There is no noise affecting the interaction. Interaction events can occur concurrently – concurrency is an inherent feature of biological and biochemical systems. There are several ways of modelling concurrent events in computer science and most of them are applicable to biological models. In the case of instantaneous events, the so-called synchronous or asynchronous updating scheme is frequently used [106]. The former scheme results in purely deterministic behaviour in which in every step all events occurring concurrently are synchronised as occurring in the same time instant. It is apparent that some behaviour can be lost in such settings. On the contrary, the asynchronous updating scheme employs the so-called interleaving of concurrent events. In particular, the model includes all possible sequences of concurrent events. They can be selected from a given state by means of a non-deterministic choice.

Deterministic rules absolutely neglect the presence of noise that causes the occurrence of events to be a random phenomenon. A well-known physical model of molecule interactions reflecting stochasticity has been addressed by Gillespie in [195, 196]. To that end, stochastic rules are considered in modelling languages based on this physical model. In the stochastic settings, the exact time between individual events reflecting occurrence of a given reaction is a random variable depending on a particular state of the system and on physical aspects of the reactant molecules.

In general, the choice of time representation, event atomicity and the way of dealing with concurrency significantly affect the expressive power of the model. For example, in the very recent work [108] it is shown that in case of discrete-time discrete-value models of influence networks a more expressive semantics not previously addressed in computer science is required to capture the biological influences correctly. It is also worth noting that a large variety of semantics addressing the issues of concurrency in the ways acceptable for biological networks is comprehensively captured in the formalism of Petri Nets [70, 105, 321, 307].

Summary of individual model types with respect to the semantics of species and rules is given in Figure 2.3. The left half-space shows models that work with discrete (or qualitative) semantics of species. The right half-space shows models employing continuous semantics of species. The vertical categorisation shows models with discrete (or untimed) semantics of rule dynamics in the top half-space. In the bottom half-space, models describing continuous-time dynamics are displayed. In the following subsections, we describe classes of models that are supported by methods developed in our research and over-viewed in the next chapters. First, we consider the most commonly used continuous-time continuous-value models for all kinds of considered biological networks – cCRNMs (continuous models of CRNs), cINMs (continuous models of INs) and cRINMs (continuous models of RINs). Second, in the stochastic case, we limit ourselves to the most commonly used stochastic models well-defined for chemical reaction networks – sCRNMs (stochastic models of CRNs). Although there exist scenarios under which Gillespie's theory can be applied to Hill kinetics of influence networks [159] and Michaelis-Menten kinetics of metabolic networks [322] making a subclass of RINs, they cannot be employed universally but only in cases when several additional criteria are satisfied. Finally, we consider discrete-time discrete-value models of influence networks – dINMs (discrete models of INs). These models (known as Boolean networks or Thomas' networks [348, 347]) are becoming popular for qualitative modelling of gene regulatory networks where the detailed knowledge is very complex to be modelled mechanistically. Moreover, genetic regulatory mechanisms are not yet fully explored. It is worth noting that discrete-time discrete-value semantics has been also used in case of CRNs [93]. However, we do not consider these models in our frame-

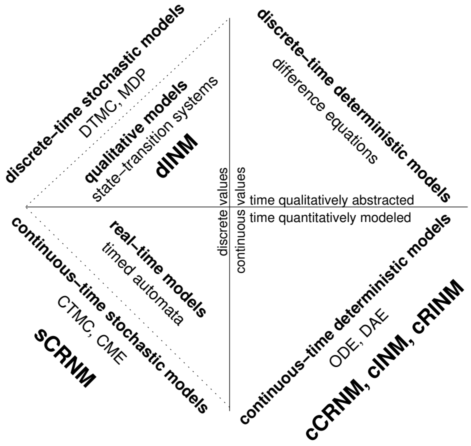

discrete–time stochastic models
DTMC, MDP
qualitative models
state–transition systems
dINM
discrete–time deterministic models
difference equations
discrete values
continuous values
time qualitatively abstracted
time quantitatively modeled
real–time models
timed automata
continuous–time stochastic models
CTMC, CME
sCRNM
continuous–time deterministic models
ODE, DAE
cCRNM, cINM, cRINM

Figure 2.3: Model types categorised with respect to different level of details captured in their semantics. The model types covered in this thesis are typeset in bold and placed to the appropriate quadrants of the scheme.

work due to the fact it is not clear what kind of (non-structural) parameters would make good sense for parameterisation of qualitative models of CRNs.

For the purpose of this thesis, we do not consider discrete-time mathematical models that make the top right quadrant of the scheme in Figure 2.3. It is worth noting that those models are relevant for mathematical biology [123] as they are used for modelling of some abstract processes at the population level, e.g., logistic growth.

2.3.1 Continuous Models

The following definition declares the most frequently used type of mathematical models that is used with CRNs. The model considers variables to evolve deterministically in continuous time in large concentrations (macroscopic view). Therefore every model variable captures a particular species concentration as a function of time. In every continuous time instant, reactions are quantified in terms of an instant flux describing the actual amount of transferred mass. To this end, the fundamental law of mass-action [206] is employed to represent *kinetic rate function* that characterises quantitatively the speed of mass transfer from reactants to products. In our case, we compile several notions into a single definition and modify the notation to avoid overlaps with other notions used throughout the thesis.

Definition 2.12 (Continuous CRN Model – cCRNM)
Let $\mathcal{N} = (\Lambda, \mathcal{R})$ *be a CRN.* Continuous chemical reaction network model of $\mathcal{N}$*, denoted cCRNM(*$\mathcal{N}$*), is defined as a tuple* $(\mathcal{N}, \nu_{cRN}, \mathbb{X}_{\mathfrak{c}}, x_0)$ *where*

- $\nu_{cRN} : \mathcal{R} \to \mathbb{R}_{\geq 0}$ *is a* (deterministic) kinetic rate coefficients map,
- $\mathbb{X}_\mathfrak{c} = \mathbb{R}_{\geq 0}^{|\Lambda|}$ *is a* (continuous) model state space *with* $x(t) \in \mathbb{X}_\mathfrak{c}$ *denoting a* (continuous) state *in time* $t \in \mathbb{R}_{\geq 0}$,
- $x_0 \in \mathbb{X}_\mathfrak{c}$ *is an* initial state.

Instead of $x(t)$ *we write* x *when a concrete value of time* t *is irrelevant in the respective context. Further, we assume the components of* x *follow the lexicographical order of* Λ. *For any* $l \in \Lambda$, *the notation* x_l *is used to denote the state component relative to* l, *the so-called* model variable.

Additionally, every $\varrho \in \mathcal{R}$ *is assigned a* mass-action kinetic function $\kappa_\varrho : \mathbb{X}_\mathfrak{c} \to \mathbb{R}_{\geq 0}$ *in the following form:*

$$\kappa_\varrho(x) = \nu_{cRN}(\varrho) \cdot \prod_{l \in \Lambda} x_l^{r_\varrho(l)}$$

Dynamics of a cCRNM is defined as a continuous function $x : \mathbb{R}_{\geq 0} \to \mathbb{X}_\mathfrak{c}$ *satisfying* $x(0) = x_0$ *and the following system of differential equations (given in vector form):*

$$\frac{dx}{dt} = f(x) = \sum_{\varrho \in \mathcal{R}} (p_\varrho - r_\varrho)\kappa_\varrho(x)$$

Example 2.13 *The CRN from Example 2.4 can be assigned a continuous model defined on the state space* $\mathbb{X}_\mathfrak{c} = \mathbb{R}_{\geq 0}$ *by the following differential equation:*

$$\frac{dx}{dt} = \nu_{cRN}(\varrho_1)x^2 - \nu_{cRN}(\varrho_2)x^3 + \nu_{cRN}(\varrho_3) - \nu_{cRN}(\varrho_4)x$$

where x *represents the one-dimensional model variable describing the development of species A. The kinetic rate coefficients* $\nu_{cRN}(\varrho_1), ..., \nu_{cRN}(\varrho_4)$ *are supposed to be fixed to some concrete positive real values.*

To describe the continuous semantics of influence networks, we follow the ideas presented in [135, 54]. In particular, every component is expected to have associated some degradation dynamics and every component with a non-empty set of influencers is assigned a production dynamics that mathematically reflects all the affecting influencers in terms of a set of so-called *regulation functions* that are expected to be monotonous.

Definition 2.14 (Continuous IN Model – cINM)
Let $\mathcal{N} = (\Lambda, \Gamma_\Lambda)$ *be an influence network.* Continuous influence network model of $\mathcal{N}$, *denoted cINM(*$\mathcal{N}$*), is defined as a tuple* $(\mathcal{N}, \langle \nu_{cIN_+}, \nu_{cIN_-} \rangle, \mathbb{X}_\mathfrak{c}, x_0)$ *where*

- $\nu_{cIN_+} : \Lambda \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ *is a* production rate coefficients map *and* $\nu_{cIN_-} : \Lambda \to \mathbb{R}_{\geq 0}$ *is a* degradation rate coefficients map,

- $\mathbb{X}_{\mathfrak{c}} = \mathbb{R}_{\geq 0}^{|\Lambda|}$ *is a* (continuous) model state space *with* $x(t) \in \mathbb{X}_{\mathfrak{c}}$ *denoting a* (continuous) state *in time* $t \in \mathbb{R}_{\geq 0}$*,*

- $x_0 \in \mathbb{X}_{\mathfrak{c}}$ *is an* initial state.

Additionally, every $l \in \Lambda$ *is assigned a (possibly empty) set of* regulation functions $\{\sigma_l^1, ..., \sigma_l^{k_l}\}$ *such that* $k_l \geq 0$ *and each* $\sigma_l^i : \mathbb{X}_{\mathfrak{c}} \to \mathbb{R}_{\geq 0}$ *is a continuous and (non-strictly) monotonous function of* x_l *satisfying* $l \in \Lambda_i$ *where* $\Lambda_i \subseteq n^-(l)$*,* $\Lambda_i \neq \emptyset$*, is a subset of* influencers of l *such that for all* $i, j \in \{1, ..., k_l\}, i \neq j. \Lambda_i \cap \Lambda_j = \emptyset$ *and* $\bigcup_{i \in \{1,...,k_l\}} \Lambda_i = n^-(l)$*. Moreover,* ν_{cIN_+} *is required to be defined for all* (l, i) *such that* $l \in \Lambda_i$*.*

Dynamics of a cINM is defined as a continuous function $x : \mathbb{R}_{\geq 0} \to \mathbb{X}_{\mathfrak{c}}$ *satisfying* $x(0) = x_0$ *and the following system of differential equations (containing an equation for every* $l \in \Lambda$*):*

$$\frac{dx_l}{dt} = f_l(x) = \sum_{i=1}^{k_l} \nu_{cIN_+}(l, i)\sigma_l^i(x) - \nu_{cIN_-}(l)x_l$$

Example 2.15 *The influence network from Example 2.7 can be associated with a continuous model under the condition that the semantics of both influences of* mRNA_X *molecule is given including their mutual (joint) effect. Assuming that* TF *regulates transcription of* mRNA_X *positively, it is apparent that it has the key role in controlling the transcription process. Since the molecule* geneX *has only the qualitative effect (transcription cannot be performed without a particular gene), we focus only on the subnetwork of the components* TF *and* mRNA_X*. A model of this subnetwork may have the following form:*

$$\frac{dx_{\mathrm{mRNA\_X}}}{dt} = \nu_{cIN_+}(\mathrm{mRNA\_X}, 1)\sigma^1_{\mathrm{mRNA\_X}}(x) - \nu_{cIN_-}(\mathrm{mRNA\_X})x_{\mathrm{mRNA\_X}}$$
$$\frac{dx_{\mathrm{TF}}}{dt} = -\nu_{cIN_-}(\mathrm{TF})x_{\mathrm{TF}}$$

where the state space has two dimensions corresponding to variables mRNA_X *and* x_{TF}*, the coefficient* $\nu_{cIN_+}(\mathrm{mRNA\_X}, 1)$ *is set to a concrete value reflecting (theoretically) maximal rate of mRNA production,* $\nu_{cIN_-}(\mathrm{RNA\_X})$ *and* $\nu_{cIN_-}(\mathrm{TF})$ *are replaced with some concrete values representing degradation coefficients, and* $\sigma^1_{\mathrm{mRNA\_X}}(x)$ *is a regulation function of* x_{TF} *defined by employing Hill kinetics in the following way:*

$$\sigma^1_{\mathrm{mRNA\_X}}(x) = \frac{x_{\mathrm{TF}}^4}{x_{\mathrm{TF}}^4 + 3^4}$$

This regulation function is a monotonous (increasing) S-shape function ranging in the domain $[0, 1)$ *and satisfying* $\sigma^1_{\mathrm{mRNA\_X}}(x) = 0.5$ *for* x *such that* $x_{\mathrm{TF}} = 3$*.*

The requirement on the monotonicity of regulation functions is presented due to practical reasons to simplify the semantics of regulations to either increasing or decreasing functions of the (combined) influencers.

Finally, we close the collection of continuous models by describing the most general class considered in this thesis – the class of reaction-influence models. The concept is the following: non-regulated reactions follow the law of mass action whereas regulated reactions have the kinetics defined by a custom monotonous function projecting the system state (including the state of influencers) into the regulated reaction rate.

Definition 2.16 (Continuous RIN Model – cRINM)
Let $\mathcal{N} = (\Lambda, \mathcal{R}, \Gamma_{\mathcal{R}})$ be a RIN. Continuous reaction-influence network model of $\mathcal{N}$*, denoted cRINM($\mathcal{N}$), is defined as a tuple $(\mathcal{N}, \nu_{cRIN}, \mathbb{X}_{\mathfrak{c}}, x_0)$ where*

- $\nu_{cRIN} : \mathcal{R} \to \mathbb{R}_{\geq 0}$ *is a* (deterministic) kinetic function coefficients map,
- $\mathbb{X}_{\mathfrak{c}} = \mathbb{R}_{\geq 0}^{|\Lambda|}$ *is a* (continuous) model state space *with $x(t) \in \mathbb{X}_{\mathfrak{c}}$ denoting a* (continuous) state *in time $t \in \mathbb{R}_{\geq 0}$,*
- $x_0 \in \mathbb{X}_{\mathfrak{c}}$ *is an* initial state.

Additionally, every $\varrho \in \mathcal{R}^{\emptyset}$ is assigned a mass-action kinetic function $\kappa_\varrho : \mathbb{X}_{\mathfrak{c}} \to \mathbb{R}_{\geq 0}$ *in the following form:*

$$\kappa_\varrho(x) = \nu_{cRIN}(\varrho) \cdot \prod_{l \in \Lambda} x_l^{r_\varrho(l)}$$

Every $\varrho \in \mathcal{R}^{-}$ is assigned a regulated kinetic function $\kappa'_\varrho : \mathbb{X}_{\mathfrak{c}} \to \mathbb{R}_{\geq 0}$ *in the following form:*

$$\kappa'_\varrho(x) = \sigma_\varrho(x, \nu_{cRIN}(\varrho))$$

where σ_ϱ is an arbitrary continuous monotonous function of all variables x_l such that $l \in n^{-}(\varrho)$ which is parameterised by $\nu_{cRIN}(\varrho)$.

Dynamics of a cRINM is defined as a continuous function $x : \mathbb{R}_{\geq 0} \to \mathbb{X}_{\mathfrak{c}}$ satisfying $x(0) = x_0$ and the following system of differential equations (given in vector form):

$$\frac{dx}{dt} = f(x) = \sum_{\varrho \in \mathcal{R}^{\emptyset}} (p_\varrho - r_\varrho)\kappa_\varrho(x) + \sum_{\varrho \in \mathcal{R}^{-}} (p_\varrho - r_\varrho)\kappa'_\varrho(x)$$

Example 2.17 *The network from Example 2.11 can be assigned the following continuous model:*

$$\begin{aligned}
\tfrac{dx_{\text{geneX}}}{dt} &= -\nu_{RIN}(\varrho_1)x_{\text{geneX}}x_{\text{TF}} \\
\tfrac{dx_{\text{geneX-TF}}}{dt} &= \nu_{RIN}(\varrho_1)x_{\text{geneX}}x_{\text{TF}} \\
\tfrac{dx_{\text{rnaX}}}{dt} &= \sigma_{\text{rnaX}}(x, \nu_{RIN}(\varrho_2)) \\
\tfrac{dx_{\text{TF}}}{dt} &= -\nu_{RIN}(\varrho_1)x_{\text{geneX}}x_{\text{TF}}
\end{aligned}$$

where the state space is determined by the four-dimensional domain of non-negative reals capturing the concentration of the four components, the semantics of the uninfluenced reaction ϱ_1 is described in terms of mass action kinetics and the *influenced reaction ϱ_2 is assigned a monotonous regulation function of the component $x_{\text{geneX-TF}}$ that may be specified by employing Hill kinetics in the following form:*

$$\sigma_{\text{rnaX}}(x, \nu_{RIN}(\varrho_2)) = \frac{\nu_{RIN}(\varrho_2) \cdot x^4_{\text{geneX-TF}}}{x^4_{\text{geneX-TF}} + 3^4}$$

where $\nu_{RIN}(\varrho_2)$ represents some concrete value that sets the maximal possible transcription rate of the RNA molecule that is controlled (by scalling) employing the same S-shaped regulation function that has been used in the previous example.

The remaining value of $\nu_{RIN}(\varrho_1)$ has to be set to reflect the physical properties of the TF-DNA complexation reaction.

An important aspect that can be seen in cRINMs is the possibility of a very general use of a kinetic rate coefficient anywhere in the regulation function (in the case of assigning the dynamics to influenced reactions). From the mathematical perspective, the class of non-linear models represented by cRINMs is not restricted at the place or regulation functions. The regular form is only dictated in case of non-influenced reactions where mass-action kinetics is employed.

Notation 2.18 *We will use the notation cBNM to refer to any kind of continuous model defined above (cCRNM, cINM or cRINM).*

The vector function $f : \mathbb{X}_{\mathfrak{c}} \to \mathbb{X}_{\mathfrak{c}}$ defining the dynamics of a cBNM is called a vector field.

Note that our framework provides the most frequently used expressions describing biological networks in terms of ODEs. There exist different forms taking into account various physical aspects of the reaction (e.g., cooperative binding [337], general mass action [225], S-systems [208], etc.)

All types of continuous-time models mentioned in this section describe the dynamics by means of a continuous function of time. Owing to the fact that these systems are non-linear, it is not in general possible to obtain the analytical solution. In particular, numerical simulation methods are used to obtain approximations of the dynamics in a finite time horizon.

The general notion that formally captures the (exact or approximated) dynamics of a continuous-time dynamical system is called a *continuous signal* (a signal for short), defined in the following way.

Definition 2.19 (Signal) *Let $n \in \mathbb{N}$ and $T = [0, \tau]$ where $\tau \in \mathbb{R}_{\geq 0}$. Then $\mathfrak{x} : T \to \mathbb{R}^n_{\geq 0}$ is a* bounded-time continuous signal *and T its* time domain. *If $T = [0, \infty)$ we speak about a* unbounded-time continuous signal.

For a given cBNM $\mathcal{M}$ of a biological network $\mathcal{N}$, we refer the signal representing a behaviour of $\mathcal{M}$ as a *signal generated by* $\mathcal{M}$. More precisely, it is either an *exact signal* representing an exact (analytical) solution, or an *approximate signal* constructed with a selected numerical method with some error.

The semantics of any cBNM is defined either as the exact signal that represents the unbounded-time dynamics from the initial state or an approximate signal that gives a bounded-time numerical approximated prefix of the exact signal.

Definition 2.20 (cCBNMs semantics) *Let $\mathcal{M}$ be a cBNM with initial state x_0.* Exact semantics *of $\mathcal{M}$, denoted $[\![\mathcal{M}]\!]_{\mathrm{cBNM}}$, is defined as a unbounded-time continuous signal $\mathfrak{x}$ given by the continuous function $x(t)$ representing the solution of the respective system of differential equations satisfying $\forall t \in \mathbb{R}_{\geq 0}.\mathfrak{x}(t) = x(t)$ and $\mathfrak{x}(0) = x_0$.*

Approximate semantics *of $\mathcal{M}$, denoted $[\![\mathcal{M}]\!]^{\epsilon}_{\mathrm{cBNM}}$, is defined as a piece-wise constant signal $\mathfrak{x}^{\epsilon} : [0, \tau] \to \mathbb{R}^{n}_{\geq 0}$ (an* approximate signal*) constructed with a selected numerical method with some error ϵ and a given time bound τ satisfying $\forall t \in [0, \tau].x(t) = \mathfrak{x}^{\epsilon}(t) + \epsilon(t)$ and $\mathfrak{x}^{\epsilon}(0) = x_0$.*

The semantics defined above employs the "local" view (fixed by a given initial state) typically used by systems biologists. This design decision is considered to emphasise the contrast between the common usage of cBNMs (simulation) and other model classes for which a "global" view of the state space and global analysis are usually employed.

2.3.2 Stochastic Models

Stochastic computational models allow to incorporate noise which causes fluctuations in microscopic or mesoscopic component quantities and that way affects the biological system dynamics [159, 239]. In physics, chemistry and related fields, the probabilistic time-evolution of a system with discrete component quantities is described by so-called master equations. In the case of biological phenomena, the chemical master equation (CME) provides an exact mathematical model for stochastic dynamics [196]. It is formalised as a set of differential equations, providing a denotational representation of component quantities distribution in continuous-time. Gillespie [195, 197] has made an important breakthrough in stochastic modelling by introducing techniques for exact simulation of CME. From the computer scientific viewpoint, the CME can be equivalently represented by continuous-time Markov chains (CTMC) [299] which provide operational semantics and allow us to consider continuous-time stochastic models as computational (executable) models [142].

The following definition fixes the syntax and semantics of the stochastic model as used in the thesis. The central notion is the hazard function describing the expected rate of a state transition. It is defined by stochastic interpretation of the law of mass action. It differs from the deterministic version by taking into account the combinatorics of selecting the set of particular molecules making the reactant complex from the current solution (state).

Definition 2.21 (Stochastic CRN Model – sCRNM)
Let $\mathcal{N} = (\Lambda, \mathcal{R})$ be a CRN. Stochastic chemical reaction model of $\mathcal{N}$*, denoted sCRNM($\mathcal{N}$), is defined as a tuple* $(\mathcal{N}, \nu_{sRN}, \mathbb{X}, X_0)$ *where*

- $\mathbb{X} \subset \mathbb{N}_0^{|\Lambda|}$ *is a finite set of admissible states of the model, the so-called* model state space,
- $X_0 \in \mathbb{X}$ *is an* initial state,
- *and* $\nu_{sRN} : \mathcal{R} \to \mathbb{R}_{\geq 0}$ *is a* stochastic rate coefficients map.

Additionally, every $\varrho \in \mathcal{R}$ *is assigned a* stochastic rate function *(also called a hazard function)* $h_\varrho : \mathbb{X} \to \mathbb{R}_{\geq 0}$ *in the following form:*

$$h_\varrho(X) = \nu_{sRN}(\varrho) \cdot \prod_{l \in \Lambda} \binom{X(l)^{\iota(l)}}{r_\varrho(l)}$$

where $\iota(l) = 1$*, if* $r_\varrho(l) > 0$*, or* $\iota(l) = 0$*, otherwise.*

For a reaction $\varrho = (r_\varrho, p_\varrho)$ *and a state* $X \in \mathbb{X}$ *we define the* state change from X to $X' \in \mathbb{X}$ induced by ϱ*, denoted* $X \xrightarrow{\varrho} X'$*, and satisfying* $X' = X - r_\varrho + p_\varrho$*.*

Semantics of a sCRNM $\mathcal{M}$*, denoted* $[\![\mathcal{M}]\!]_{\text{sCRNM}}$*, is defined as a CTMC* $[\![\mathcal{M}]\!]_{\text{sCRNM}} = (\mathbb{X}, X_0, \mathbf{Q})$ *representing a continuous-time discrete-state Markov process* $\{X(t) | t \in \mathbb{R}_{\geq 0}\}$ *where* $\mathbf{Q} \in \mathbb{R}^{|\mathbb{X}| \times |\mathbb{X}|}$ *is the transition matrix defined as:*

$$\forall X_i, X_j \in \mathbb{X}, X_i \neq X_j . \mathbf{Q}(X_i, X_j) = \sum_{\varrho \in \mathbf{Q}(X_i, X_j)} h_\varrho(X_i)$$

where $\mathbf{Q}(X_i . X_j) = \{\varrho \in \mathcal{R} | X_i \xrightarrow{\varrho} X_j\}$*, and additionally,*

$$\forall X_i \in \mathbb{X} . \mathbf{Q}(X_i, X_i) = - \sum_{X_j \neq X_i} \mathbf{Q}(X_i, X_j).$$

Individual state components X_l *are called (random)* model variables*.*

For a given state $X \in \mathbb{X}$ *the set of all paths in the state transition system underlying* $[\![\mathcal{M}]\!]_{\text{sCRNM}}$ *and starting at* X *is denoted* $Path^{[\![\mathcal{M}]\!]_{\text{sCRNM}}}(X)$*.*

In contrast to cCRNMs, the semantics of a sCRNM is defined as an entire structure representing all possible behaviours of a system – the so-called *global view* of systems dynamics. The selection of an initial state X_0 can be understood as an additional information used by specific analysis tasks employing the local view on systems dynamics (e.g., simulation or local model checking).

The probability of a transition from state X_i to X_j occurring within t time units is $1 - e^{-\mathbf{Q}(X_i, X_j) \cdot t}$, if such a transition cannot occur then $\mathbf{Q}(X_i, X_j) = 0$. The time before any transition from X_i occurs is exponentially distributed with an overall *exit rate* $E(X_i)$ defined as $E(X_i) = \sum_{X_j \in \mathbb{X}} \mathbf{Q}(X_i, X_j)$.

A path π of a CTMC is a non-empty sequence $\pi = X_0 t_0 X_1 t_1 \dots$ where $\forall i, j \geq 0$. $\mathbf{Q}(X_i, X_j) > 0$ and $t_i \in \mathbb{R}_{\geq 0}$ is the amount of time spent in the state X_i for all $i \geq 0$. There exists the unique probability measure on $Path^{[\![\mathcal{M}]\!]_{\mathrm{sCRNM}}}(X)$ defined, e.g., in [260]. Intuitively, any subset of $Path^{[\![\mathcal{M}]\!]_{\mathrm{sCRNM}}}(X)$ has a unique probability that can be effectively computed. For the CTMC $\mathcal{C}$ the transient state distribution is denoted $\pi^{\mathcal{C},X,t}$. It gives for all states $X' \in \mathbb{X}$ the transient probability $\pi^{\mathcal{C},X,t}(X')$ defined as the probability, of being in state X' at the finite time t, having started in the state X.

From the modelling perspective specific to stochastic models, there is yet another notion of a quantity that can enhance the model semantics. In particular, interactions and even variable values can be assigned quantitative costs and *rewards*, e.g., time spent in particular concentration levels, energy consumed by particular reactions, etc. By adding this kind of information (if available), models can be adjusted to provide interesting and detailed quantitative predictions resulting from complex dynamics. The notion of rewards is formally introduced in Section 2.5.5.

2.3.3 Discrete Models

Computational models are inherently discrete provided that the dynamics (execution) occurs in terms of a series of discrete events. In the untimed setting, non-determinism allows capturing all possible "timings" (orderings) of concurrent events.

For the purpose of this thesis, we consider a class of discrete-time discrete-value models frequently used for modelling of influence networks. These models are known as Boolean networks [348] or multi-valued Thomas' networks [347, 62]. We use a general definition that has been stated in our work [248].

Definition 2.22 (Discrete Influence Network Model – dINM)
Let $\mathcal{N} = (\Lambda, \Gamma_\Lambda)$ be an influence network. Discrete influence network model of $\mathcal{N}$, *denoted dINM($\mathcal{N}$), is defined as a tuple $(\mathcal{N}, \nu_{dIN}, \mathbb{X}, X_0, \mathbf{m})$ where*

- $\mathbf{m} \in \mathbb{N}^{|\Lambda|}$ *is a* vector of maximal values,
- $\mathbb{X} \overset{df}{=} \prod_{l \in \Lambda} \{0, ..., \mathbf{m}_l\}$ *is the* model state space,
- $X_0 \subseteq \mathbb{X}$ *is a* set of initial states,
- *and $\nu_{dIN} : \Omega \to \{0, \dots, \mathbf{m}_l\}$ is an* influencing configurations mapping *where $\Omega \overset{df}{=} \bigcup_{l \in \Lambda} (\{l\} \times \Omega_l)$ is the set of all* influencing configurations *with $\Omega_l \overset{df}{=} \prod_{u \in n^-(l)} \{0, \dots, \mathbf{m}_u\}$ denoting the set of* contexts *affecting the state component X_l. A context is assumed to reflect the lexicographic order of Λ. The meaning of $\nu_{dIN}(l, \omega) = k$ (for some $k \in \{0, ..., \mathbf{m}_l\}$) is the*

intention of setting the state component X_l to the target value *k if X meets the context ω.*

Semantics of a dINM $\mathcal{M}$*, denoted $[\![\mathcal{M}]\!]_{\mathrm{dINM}}$, is defined as a state-transition system $[\![M]\!]_{\mathrm{dINM}} = (\mathbb{X}, T, X_0)$ where $T \subseteq \mathbb{X} \times \mathbb{X}$ is a transition relation generated by the dINM in the following way:*

$$X \to X' \in T \textit{ iff either } \nu_{dIN}(l, \omega_l(X)) > X_l \wedge (X_l < \mathbf{m}_l \wedge X' = X_{[l \mapsto X_l + 1]}) \\ \textit{or } \nu_{dIN}(l, \omega_l(X)) < X_l \wedge (X_l > 0 \wedge X' = X_{[l \mapsto X_l - 1]})$$

where $\omega_l : \mathbb{X} \to \Omega_l$ with $\omega_l(X) \overset{df}{=} \prod_{u \in n^-(l)} \{X_u\}$ is the projection of the state X to the context affecting the component X_l reflecting the lexicographic order of Λ and $X_{[l \mapsto k]}$ denotes X with an updated component X_l such that $X_l = k$.

Similarly to the case of sRNMs, the semantics of dINMs is represented globally by means of a state-transition system. Moreover, the notion of initial states is generalised to an arbitrary subset of the model state space. This is a model design decision reflecting the typical usage of dINMs for global analysis of systems dynamics, e.g., finding states from which certain attractors (strongly connected components of the state-transition graph) can be reached.

Example 2.23 *The influence network from Example 2.7 can be associated with a dINM under the condition that the semantics of both influences of* mRNA_X *molecule is given including their mutual (joint) effect. The gene* geneX *has a direct positive effect on the production of* mRNA_X*. Assuming that* TF *regulates transcription of* mRNA_X *positively, we have both influences affecting* mRNA_X *positive. The logic of the mutual effect of both influences has the form of the logical AND. The reason is that the presence of both the gene and the transcription factor in the cell is required to perform the transcription of* mRNA_X*. Assuming the lexicographically ordered set of species $\Lambda = \{\mathrm{geneX}, \mathrm{mRNA\_X}, \mathrm{TF}\}$, a model that considers boolean variables ($\mathbf{m}_l = 1$ for all species $l \in \Lambda$) can have the following form:*

$$\begin{array}{ll} & \nu_{dIN}(\mathrm{mRNA\_X}, (0,0)) = 0 \\ \mathbb{X} = \{(x_1, x_2, x_3) \mid x_1, x_2, x_3 \in \{0,1\}\} & \nu_{dIN}(\mathrm{mRNA\_X}, (0,1)) = 0 \\ X_0 = \mathbb{X} & \nu_{dIN}(\mathrm{mRNA\_X}, (1,0)) = 0 \\ & \nu_{dIN}(\mathrm{mRNA\_X}, (1,1)) = 1 \end{array}$$

2.3.4 Parameterised Models

We are interested in models with uncertain or unknown parameters. To that end, we extend models to become functions over parameter values that are assigned to model parameters. Parameters may take values from a given *parameter domain* $\mathbb{P}$. In the case of cBNMs and sCRNMs, such domain is the set of positive reals, $\mathbb{P} = \mathbb{R}_{\geq 0}$. In the case of dINMs, the parameter domain is a set of natural numbers, $\mathbb{P} = \mathbb{N}_0$. An element $p \in \mathbb{P}$ is called a *parameter value*.

In this section we provide a general concept of model parameterisation that has been, to the best of our knowledge, not done yet. The intuition behind our concept is the following. In the context of quantitative models, the parameterisation reflects the typical scenario in systems biology where quantities represented by kinetic rate coefficients are considered as parameters. In the case of discrete models, the parameters are qualitative and reflect the settings of the logic driving the model dynamics. More specifically, the parameterisations we consider here target the kinetic coefficients mappings $\nu_{cRN}, \nu_{sRN}, \nu_{cIN_+}, \nu_{cIN_-}, \nu_{cRIN}$ (in case of cBNMs and sRNMs) and ν_{dIN} (in case of dINMs). The domain of these mappings is extended with a new symbol $\perp$ representing an undefined parameter value. In particular, $\nu_\bullet : \mathcal{R} \to \mathbb{R}_{\geq 0} \cup \{\perp\}$ for $\bullet \in \{\text{cRN}, \text{cRIN}, \text{sRN}\}$, $\nu_{cIN_+} : \Lambda \times \mathbb{N} \to \mathbb{R}_{\geq 0} \cup \{\perp\}$, $\nu_{cIN_-} : \Lambda \to \mathbb{R}_{\geq 0} \cup \{\perp\}$, and $\nu_{dIN} : \Omega \to \mathbb{N}_0 \cup \{\perp\}$.

A concrete parameterisation of a particular model is given by the set of objects for which the respective mapping is set to $\perp$. In case of cRNMs, sRNMs and cRINMs the parameterised objects are reactions. In the case of cINMs and dINMs the situation is more complicated since the parameters we consider are set more deeply in the structures representing the model semantics. In dINMs, the parameterised object are the selected influencing configurations. In cINMs, the parameterisation comprises species with unknown degradation coefficients and production rate coefficients assigned to individual regulation functions. The formal definition of parameterised variants of the considered model classes is given in Definition 2.24.

Definition 2.24 (Parameterised Models) *Let $\mathbb{P}$ be a parameter domain. Let $\mathcal{N}$ be a reaction-influence network $\mathcal{N} = (\Lambda, \mathcal{R}, \Gamma_{\mathcal{R}})$, a reaction network $\mathcal{N} = (\Lambda, \mathcal{R})$, or an influence network $\mathcal{N} = (\Lambda, \Gamma_\Lambda)$ associated with a corresponding model (cRINM, cCRNM/sCRNM, or cINM/dINM, respectively).*

For an arbitrary model $\mathcal{M}$ we define its parameterisation *$\chi_{\mathcal{M}}$ (the set of unknown objects that are parameterised – represented by appropriate* parameters*), the respective* parameter valuations *assigning each model parameter a particular* parameter value *from $\mathbb{P}$, producing the* parameter space *$\mathcal{P}_{\chi_{\mathcal{M}}}$ (the set of all parameter valuations):*

- *If $\mathcal{M}$ is a cCRNM, $\mathcal{M} = (\mathcal{N}, \nu_{cRN}, \mathbb{X}_{\mathfrak{c}}, x_0)$, then $\chi_{\mathcal{M}}$ is defined as a set of reactions for which the kinetic rate coefficient is considerex undefined, $\chi_{\mathcal{M}} \overset{df}{=} \{\varrho \in \mathcal{R} \mid \nu_{cRN}(\varrho) = \perp\}$. The undefined values $\nu_{cRN}(\varrho)$ such that $\varrho \in \chi_{\mathcal{M}}$ make the* model parameters.

- *If $\mathcal{M}$ is a cRINM or sCRNM then all the notions are defined analogously to the previous case (the only formal difference is in considering the kinetic* *function coefficients map ν_{cRIN} or the stochastic kinetic coefficients map ν_{sRN} instead of ν_{cRN}, respectively).*

- *If $\mathcal{M}$ is a cINM, $\mathcal{M} = (\mathcal{N}, \langle \nu_{cIN_+}, \nu_{cIN_-} \rangle, \mathbb{X}_\mathfrak{c}, x_0)$, then $\chi_\mathcal{M}$ is defined as $\chi_\mathcal{M} \overset{df}{=} \chi^+_\mathcal{M} \cup \chi^-_\mathcal{M}$ where $\chi^+_\mathcal{M} \overset{df}{=} \{(l, i) \in \Lambda \times \mathbb{N} \mid \nu_{cIN_+}(l, i) = \perp \wedge i \leq k_l\}$ and $\chi^-_\mathcal{M} \overset{df}{=} \{l \in \Lambda \mid \nu_{I-}(l) = \perp\}$ are arbitrary sets of respective coefficients considered undefined. The undefined values $\nu_{cIN_+}(l, i)$ such that $(l, i) \in \chi^+_\mathcal{M}$ and $\nu_{cIN_-}(l)$ such that $l \in \chi^-_\mathcal{M}$ make the* model parameters.

- *If $\mathcal{M}$ is a dINM, $\mathcal{M} = (\mathcal{N}, \nu_{dIN}, \mathbb{X}, X_0, \mathbf{m})$, then the parameterisation $\chi_\mathcal{M}$ is defined as a set of influencing configurations with undefined target value, $\chi_\mathcal{M} \overset{df}{=} \{(l, \omega) \in \Omega \mid \nu_{dIN}(l, \omega) = \perp\}$. The undefined values $\nu_{dIN}(l, \omega)$ such that $(l, \omega) \in \chi_\mathcal{M}$ make the* model parameters.

The parameter space *of $\mathcal{M}$ is defined as $\mathcal{P}_{\chi_\mathcal{M}} \overset{df}{=} \mathbb{P}^{\chi_\mathcal{M}}$. A mapping $\mu \in \mathcal{P}_{\chi_\mathcal{M}}$, $\mu : \chi_\mathcal{M} \to \mathbb{P}$, is called a* parameter valuation.

Intuitively, the parameter space $\mathcal{P}_{\chi_\mathcal{M}}$ is given by a set of valuations of model parameters. The parameterisation $\chi_\mathcal{M}$ targets the interactions (or reactions) for which the corresponding quantities assigned in the respective coefficients map are considered unknown.

The following notation sets the notion of concretisation of a parameterised model by replacing a given coefficients map with the selected parameter valuation.

Notation 2.25 *For a model $\mathcal{M}$ of arbitrary type and a parameter space $\mathcal{P}_{\chi_\mathcal{M}}$ considered for a given parameterisation $\chi_\mathcal{M}$ we denote $\mathcal{M}_{\mathcal{P}_\chi}$ a corresponding* parameterised model.

For a given parameter valuation $\mu \in \mathcal{P}_{\chi_\mathcal{M}}$ we denote $\mathcal{M}_{\mathcal{P}_\chi}(\mu)$ the concrete model in which the parameters are set with respect to μ.

In particular, if $\mathcal{M}_{\mathcal{P}_\chi}$ is a cCRNM then $\mathcal{M}_{\mathcal{P}_\chi}(\mu)$ is a variant of $\mathcal{M}$ where $\nu_{cRN}(\varrho) = \mu(\varrho)$ for every model parameter $\nu_{cRN}(\varrho)$, $\varrho \in \chi_\mathcal{M}$. Parameter valuations of cRINMs and sCRNMs are treated analogously.

If $\mathcal{M}_{\mathcal{P}_\chi}$ is a cINM then $\mathcal{M}_{\mathcal{P}_\chi}(\mu)$ is a variant of $\mathcal{M}$ where $\nu_{cIN_+}(l, i) = \mu(l, i)$ for every model parameter $\nu_{cIN_+}(l, i)$ such that $(l, i) \in \chi^+_\mathcal{M}$ and $\nu_{cIN_-}(l) = \mu(l)$ for every model parameter $\nu_{cIN_-}(l)$ such that $l \in \chi^-_\mathcal{M}$.

If $\mathcal{M}_{\mathcal{P}_\chi}$ is a dINM then $\mathcal{M}_{\mathcal{P}_\chi}(\mu)$ is a variant of $\mathcal{M}$ where $\nu_{dIN}(l, \omega) = \mu(l, \omega)$ for every model parameter $\nu_{dIN}(l, \omega)$ such that $(l, \omega) \in \chi_\mathcal{M}$.

Finally, we lift the notion of model semantics to the semantics of parameterised models. In the case of infinite sets of parameter values (this is inherent for continuous parameters considered in cases of cBNMs and sRNMs), the lifted notion of semantics can result in infinite sets of structures.

Notation 2.26 *The semantics of a parameterised model $\mathcal{M}_{\mathcal{P}_\mathcal{X}}$ is denoted $[\![\mathcal{M}_{\mathcal{P}_\mathcal{X}}]\!]_\bullet$ where $\bullet \in \{$cRNM, cRINM, cINM, sCRNM, dINM$\}$. It represents the set of structures representing semantics of individual unparameterised models: $[\![\mathcal{M}_{\mathcal{P}_\mathcal{X}}]\!]_\bullet \overset{df}{=} \{[\![\mathcal{M}_{\mathcal{P}_\mathcal{X}}(\mu)]\!]_\bullet \mid \mu \in \mathcal{P}_{\mathcal{X}_\mathcal{M}}\}$.*

The parameterisation framework presented in this section is universal in terms of the possibility to extend the basic parameterisation to more complex parameterisations of mathematical functions representing the model semantics. A way how this can be done is demonstrated in the case of cINMs. A similar scenario may be relevant in the case of cRINMs when more parameters can be necessary to be considered per a single regulated kinetic function. In our simplified case, we have fixed a single parameter per a regulated kinetic function.

In some cases, the concept of model parameterisation can be extended to include the choice of initial conditions as a parameter. This is often employed in cBNMs or sRNMs in order to globalise the view on systems dynamics. In this thesis, we do not consider this kind of parameterisation although it has been targeted in our work on cBNMs [86] and sRNMs [359]. In particular, in this thesis we clearly separate the global analysis wrt generalisation of initial conditions from the globalisation performed by parameterising the models. Mixing of both forms of abstraction in a single notion would be counter-intuitive.

Bounded Parameters

It is worth noting that the parameter domains considered in Definition 2.24 are considered to be unbounded in general. However, in the case of dINMs it follows from the semantics (Definition 2.22) that the domains of individual influences settings mappings are always bounded by the maximal value that closes the range of the affected variable. In the case of cBNMs and sCRNMs the domain of parameters is, in general, an unbounded set of positive real numbers. In practical scenarios, the parameters are always considered bounded by (bio)physical limits (e.g., maximal physically possible rate of complexation given by the structural properties of a related enzyme, maximal physically possible rate of degradation of a species in a given physical environment, etc.) These limits can be found in biological databases (e.g., Brenda [327] or BioNumbers [287]).

2.3.5 Relations Among Models

From the computational point of view, there exist many well developed and efficient techniques for exhaustive analysis of models appearing in the top left quadrant of the scheme in Figure 2.3. In the case when boundedness assumptions are applied, the models have a finite number of states. However, models in other quadrants incorporate continuous or dense quantities which significantly complicate or even disallow the exact analysis. Model reduction techniques are often employed to translate models into a simplified form in which analysis can be made more efficient, decidable or semi-decidable depending on the character of the model and the respective analysis problem.

In the non-stochastic part of the bottom left quadrant, the reduction of timed automata into untimed finite automata [7] enables exhaustive analysis for a certain class of continuous-time discrete-value models. In the stochastic part, there is a reduction procedure allowing to reduce a continuous-time stochastic model by a continuous-time Markov chain to a discrete-time Markov chain and a Poisson process (or a birth process) by uniformization techniques [339, 142]. All reductions at this level are exact, provided that no information is lost.

Concerning the bottom right quadrant, there are techniques to abstract (or approximate) continuous-value models by discrete-value models. Formal abstractions allow specific properties to be preserved by means of over-approximation (resp. under-approximation) of model behaviour. Over-approximating abstractions are conservative in the sense that each behaviour of the original model is also present in the abstract model but there can appear a new behaviour, not present in the original model. Under-approximating abstractions ensure that no execution is added to the abstract model but some can be lost. For certain classes of ordinary differential equations, there exist formal abstraction techniques providing a discrete-time discrete-value over-approximation in terms of non-deterministic finite automata [135, 55, 209, 120] or a continuous-time discrete-value over-approximation in terms of timed automata [276]. In the former case, the extent of falsely added executions is usually large whereas the latter case prevents the addition of any executions with non-realistic timing and therefore the number of false executions can be reduced.

Besides formal abstraction, there are approximations that distort the original behaviour. Such approximations do not guarantee the preservation of dynamics but the deviation of behaviour is ensured not to exceed a certain (specified) approximation error. In the case of continuous-time deterministic models, typically non-linear, the most widely used approximation is provided by numerical simulation (integration) methods.

Some classes of continuous-time stochastic models can be approximated by deterministic continuous-time models (ODEs) by means of fluid-approximation techniques [131]. Based on these techniques, more sophisticated analysis methods for stochastic models, combining fluid-approximation with CTMC analysis have recently been proposed [74]. The advantage of these techniques is that they avoid the state-explosion problem. Another approximation technique for stochastic models is called linear noise approximation [71]. It is based on approximating the discrete-value stochastic semantics of a sCRNM in terms of a continuous-value Gaussian process.

2.3.6 Other Types of Models

Rule-Based Models

Models defined in previous sections have dealt with fixed (and finite) network structures. Such a setting is adequate for high-level modelling, but it can be limiting when a more detailed level of modelling is required. For example, let us consider a phosphorylation reaction that affects a structure of a certain protein P by attaching a phosphate group to one of its amino acid sites. In terms of reaction networks, we have to consider a new species P' that represents the protein under that particular modification. However, there might be many other sites that can be phoshorylated in a similar way. In particular, protein P can exist in several states that combine the presence of phosphorylated sites arbitrarily. When using reaction or influence networks, we have to model all those states explicitly as individual forms of P. The central feature of rule-based models is to avoid the combinatorial explosion by lifting a concrete species to a species type that assigns the species with a set of allowed modifications (states). Reactions are lifted to rules that act over species types instead of concrete species. Such an approach allows powerful compaction of model descriptions.

Rule-based languages that form a fundamental basis of the rule-based paradigm are κ-calculus [130, 78] and BNGL [161, 63]. In addition to expressing species states the language has constructs to describe bonds among the species structures. In full κ-calculus, rates are assigned to rules allowing to turn the model state space into a continuous-time stochastic model driven by a CTMC in a way similar to Definition 2.21 under the assumption that the set of solutions generated by the rules is finite. There are also ways to deal with unbounded models in terms of simulation [336]. Owing to the complex forms of rules targeting structural changes of biological objects, achieving a correct stochastic semantics is a non-trivial goal and there are several recent papers targeting those issues in [96, 170].

The principle of rule-based modelling has been employed also in the tool BioCham addressing a comprehensive set of analysis methods [167, 94, 166]. In our research, we have developed an abstract rule-based language BCS allowing compact human-readable description of chemical reaction networks [136]. In [88] we have demonstrated on existing models of photosynthesis that rule-based description can significantly reduce the size of the model by compactly capturing its unambiguous core part.

Based on the idea of rule-based languages, there has been recently developed a new typed multi-scale language Chromar allowing to express very general attributes of biological objects [223] while being based on rigorous structural operational semantics. The language makes an important contribution for application of rule-based methods in modelling of biological structures and processes at the multi-cellular level.

Process Algebraic Models

Process algebraic models provide an expressive framework that allows formal specification and modular (compositional) analysis of concurrent processes [310]. The processes are described in a compact form without any ambiguity about the interactions, communications, and synchronisations between a collection of concurrent processes (also called agents). Under this framework, biological species can be modelled as processes interacting with each other. The compositionality is an important aspect of process algebraic approaches, it offers the possibility of defining the whole (possibly multi-scale) system hierarchically, starting from the specification of its subcomponents. Examples of process algebraic approaches in computational systems biology are Beta-Binders/BlenX [138], SPiM [304], BioSPI [316], Bio-PEPA [114], sCCP [76], and BioShape [42]. Enhancing the process algebraic description with typed structures allows description of high-level processes such as translocation or movement between cellular parts [138]. Several approaches specifically focus on inter-cellular interaction [315, 98].

The relation of process algebraic and rule-based description and the framework of biological models used in the thesis is the following. Specifications encoded in process algebraic or rule-based languages are usually employed as intermediate models that are then translated in basic computational models using different semantics: deterministic (ODEs), stochastic (Continuous Time Markov Chains), or qualitative (transitions systems). That way, the rule-based and process-algebraic framework can be considered as a pre-processor for biological models defined in Section 2.3. Practically, the translation can be done under the assumption that the resulting models have a finite number of state components.

Petri Nets

The modelling framework of Petri nets was introduced by Carl Adam Petri in 1962 with the purpose of describing chemical processes [303]. Petri nets have been exploited by computational systems biologists as a formalism suitable for the description of biochemical reaction systems, where the tokens are interpreted as single molecules of the species involved. The Petri net formalism provides a natural framework in which both qualitative (given by the static structural topology of the Petri nets) and quantitative (given by the time evolution of the token distribution) analysis are tightly integrated [215]. Important tools for Petri nets used in computational biology are Snoopy [216], MARCIE [328], GreatSPN [21], and MIRACH [246].

Advantages of Petri nets related to their application to biological models are two-fold. First, high-level extensions of Petri nets such as coloured Petri nets [194], Time or Hybrid Petri nets [307, 281, 144] allow describing processes at an abstract and even multi-scale level. Second, Petri nets provide a basis for specific analysis methods that are performed at the level of the formalism, e.g., [214, 321, 69, 107, 202].

Discrete Time Stochastic Models

Discrete computational models employ non-determinism to model concurrency. To quantitatively differentiate among all possible executions in a particular state, rules can be assigned probabilities, resulting in discrete stochastic models [77, 367] most typically represented by discrete-time Markov chains (DTMC).

Hybrid Models

In addition to the basic model paradigms mentioned in previous sections, it is worth mentioning that there is a class of models that allows to combine the discrete and continuous semantics. Such models are called *hybrid models*. They are most typically represented by means of hybrid automata [217] or process algebraic techniques [185, 75]. Hybrid models allow to mix discrete-value components with continuous-value components and discrete-time dynamics with continuous-time dynamics. Such a complicated semantics limits the model analysis [219, 95]. Hybrid models can be satisfactorily used for modelling and simulation [144] and, when simplifying assumptions are employed (e.g., considering linear dynamics of continuous components), also for a more advanced analysis of biological processes [128, 181]. A hybrid model represented in terms of a hybrid automaton with uncertain parameters is employed in [28] targeting design of robust cardiac pacemakers. The work exploits the design problem of estimating optimal parameters of pacemakers.

To incorporate noise, stochastic hybrid models [335] (i.e., stochastic hybrid automata) allow both discrete-time and continuous-time dynamics to evolve randomly. Coupling of both kinds of dynamics while keeping their stochasticity complicates analysis even more. To this end, simulation-based (statistical) [133] or fluid-flow approximation techniques [75, 122] have been developed.

2.4 Model Encoding and Abstraction

In Section 2.3 we have presented several kinds of mathematical and computational models used in systems biology. The main focus in this thesis is to cBNMs, sCRNMs, and dINMs that cover the commonly used types of models. In this section, we address the problem of translating these models into low-level formalisms that enable us to perform computer-aided analysis of the models.

In particular, we start with discussing the issue of bounded model variable and parameter domains that make an important precursor of decidability of most of the analysis methods. Next, we present extended variants of several formal methods structures. Finally, methods that allow translating the models into these formalisms.

2.4.1 Model Variables

Firstly, it is important to discuss the number of variables that might appear in a model. Since this number gives the dimension of the underlying dynamical systems, for many methods it is crucial to have a finite number of dimensions. In the case of all network models, we consider in this thesis, the number of species is always assumed to be finite. However, in biological reality (e.g., from the perspective of evolutionary biology) the number of distinct species can be undetermined. The only computational models that are capable of dealing with this issue (e.g., expressing polymerisation) are provided by rule-based or process-algebraic formalisms. However, in the case of unbounded network models, the analysis tasks are limited to static analysis and simulation [336].

Secondly, it is important to note that component quantities in biological models most typically do not evolve unlimitedly under normal conditions. In particular, concentration (or a number of molecules) is always limited by physical degradation processes. However, it might not be easy to identify the exact bounds without a deeper analysis of the model. In any case, some physical assumptions on maximal (extreme) bounds on component quantities can be always considered. In our encoding of models, we always consider bounded variable domains.

Considering dINMs, bounds on variables are inherently included in the model. In particular, every species has a finite domain of discrete "concentration levels" provided that the maximal level can be understood as an "arbitrary value above any number from the lower level". The problem of mapping experimentally measured data into such a finite domain has been extensively studied and algorithms are available [332].

In the case of the continuous and stochastic models, no bounds are given and the dynamics of the model can theoretically evolve unboundedly. Introducing any bounds without the risk of losing some important information on systems dynamics requires detailed analysis of the systems dynamics that can be very hard or even intractable in general (e.g., equilibria and their asymptotic properties). In practical cases, reasonable physical bounds are used together with assumption-based principles applied to reason about the dynamics at the system's boundary. In any case, the variable domains remain uncountable and must be treated accordingly in a computer.

2.4.2 Model Parameters

The parameter domains we consider are $\mathbb{R}_{\geq 0}$ and $\mathbb{N}_0$, for quantitative models and discrete models, respectively. Apparently, these domains reflect all theoretically possible settings (valuations) of parameter values and are thus unbounded in general. In the case of dINMs, the set of parameter values is typically finite since the parameters representing unknown influences take values from some finite domain (combinations of values from the finite domain of variable values). In the case of quantitative models, the range of admissible parameter values is typically bounded by physical limitations of the particular reactions or processes (corresponding maximal numbers are usually publicly available). Nevertheless, the range remains infinite since the parameters domain in these models is uncountable.

2.4.3 Parameter Perturbations

In many practical cases it is useful to study the system dynamics in a smaller range of parameter values around a certain reference value. In that case, a lower bound and an upper bound are considered for model parameters. In other words, the meaning of $\perp$ covering all admissible values is changed to cover only values from a given interval. The (multi-dimensional) range of parameter valuations is called a *perturbation space*.

Let $\mathcal{M}$ be a cCRNM, a cRINM or a sCRNM with a parameterisation $\chi_{\mathcal{M}}$. We assign each $\varrho \in \chi_{\mathcal{M}}$ a so-called *perturbation interval* $[min_{\varrho}, max_{\varrho}] \subseteq \mathbb{P}$ where min_{ϱ}, $max_{\varrho} \in \mathbb{P}$ are the minimal and maximal allowed rate coefficients of the reaction ϱ. The parameter space $\mathcal{P}_{\chi_{\mathcal{M}}}$ can be set to exactly cover a given perturbation space, in particular, every parameter valuation $\mu \in \mathcal{P}_{\chi_{\mathcal{M}}}$ then satisfies $\mu(\varrho) \in [min_{\varrho}, max_{\varrho}]$ for every $\varrho \in \chi_{\mathcal{M}}$. In general, the set of parameter valuations restricted to the product of perturbation intervals is called a *perturbation space*, denoted $\mathbf{P}$, $\mathbf{P} \subseteq \mathcal{P}_{\chi_{\mathcal{M}}}$, defined in the following way:

$$\mathbf{P} \overset{df}{=} \{\mu \mid \mu(\varrho) \in [min_{\varrho}, max_{\varrho}], \varrho \in \chi_{\mathcal{M}}, \mu \in \mathcal{P}_{\chi_{\mathcal{M}}}\}.$$

Every parameter valuation having such a restricted form is called a *perturbation*. The concept of perturbation space is analogously extended to remaining types of models (dINMs and cINMs). In general, every parameterised object is assigned a min-max interval instead of $\perp$.

2.4.4 Parameterised Kripke Structure

For non-stochastic models, we use the notion of Kripke structure as the underlying low-level formalism to encode the model semantics. The reason is that most of our methodology is based on model checking methods which require to associate state-transition semantics with temporal logic. Kripke structure [115] is a state-transition system extended with atomic propositions associated to states and thus allowing to reason over its states and paths with temporal logics. In particular, we extend the standard notion of Kripke structure to allow encoding of parameterised models. To that end, the notion of *parameterised Kripke structure* is defined in the following definition.

Definition 2.27 *Let* AP *be a set of atomic propositions. A* parameterised Kripke structure *(PKS) over* AP *is a tuple* $\mathcal{K} = (\mathcal{P}, S, S_0, \rightarrow, L)$ *where* $\mathcal{P}$ *is a set of parameter values (a so-called* parameter space of $\mathcal{K}$*),* S *is a finite set of states,* $S_0 \subseteq S$ *is the set of initial states,* $L : S \rightarrow 2^{\mathrm{AP}}$ *is a labelling of the states and* $\rightarrow \subseteq S \times P \times S$ *is a transition relation labelled with the parameter valuations. We write* $s \xrightarrow{p} t$ *instead of* $(s, p, t) \in \rightarrow$. *We assume that the PKS is total, i.e., for all* s, p *there exists at least one* t *such that* $s \xrightarrow{p} t$.

Notation 2.28 *We use the notation* $\mathcal{P}(s,t) = \{p \in \mathcal{P} \mid s \xrightarrow{p} t\}$ *to denote the* transition-enabling set *of all parameter values under which the transition from* s *to* t *is allowed.*

Remark 2.29 *For the specific analysis task based on LTL model checking we also use a slightly extended definition of (fair) PKS described by a tuple* $\mathcal{K} = (\mathcal{P}, S, S_0, F, \rightarrow, L)$ *where the set of* fair states $F \subseteq S$ *is added.*

Fixing a concrete parameter value $p \in \mathcal{P}$ reduces a parameterised Kripke structure $\mathcal{K}$ to a standard Kripke structure $\mathcal{K}_p = (S, S_0, \xrightarrow{p}, L)$. A parameterised Kripke structure can be seen as a Kripke structure with labelled transitions, where the transition labels are the sets $\mathcal{P}(s,t)$.

Definition 2.30 *Let* $\mathcal{K} = (\mathcal{P}, S, S_0, \rightarrow, L)$ *be a PKS. Define a* p-path *in the PKS* $\mathcal{K}$*, denoted* π_p*, as a (possibly infinite) sequence of states* $\pi_p = s_0 s_1 \cdots$ *such that* $s_0 \in S_0$*,* $s_i \xrightarrow{p} s_{i+1}$ *for all* $i \geq 0$*. To denote the* i*th state of* π_p *we use the notation* $\pi_p(i)$*,* $\pi_p(i) = s_i$.

For a PKS $\mathcal{K} = (\mathcal{P}, S, S_0, F, \rightarrow, L)$ *with fair states* F*, a* p-path π_p in $\mathcal{K}$ *is defined as a fair infinite-length* p*-path, i.e., a* p*-path such that there is a state* $\gamma \in F$ *which appears in the sequence infinitely often.*

It is worth noting that a notion similar to parameterised Kripke structure we employ here has been also mentioned in the context of hardware verification, in particular, to capture delay parameters in logical circuits [301, 211]. Our notion is conceptually different since we define parameterised Kripke structure as a general structure that encapsulates the family of Kripke structures for each possible valuation of the parameters. In our case, the parameterised object is the entire transition relation.

As depicted in Figure 2.4 and described in Section 2.4.6, any cBNM can be approximated and abstracted by means of a PKS. The relevant procedures are described in Section 2.4.6.

It is also worth noting that the notion of parameter space of a PKS does not necessarily coincide with the notion of parameter space $\mathcal{P}_{\chi_\mathcal{M}}$ of a given model $\mathcal{M}$. In particular, this applies in cases of cINMs and cRINMs. Again, more on that is discussed in Section 2.4.6.

2.4.5 Parameterised CTMC

In the case of sCRNMs, we set the parameterised models semantics in terms of parameterised CTMCs. In particular, we assume an sCRNM $\mathcal{M} = (\mathcal{N}, \nu_{sRN}, \mathbb{X}, X_0)$ with a parameterisation $\chi_\mathcal{M}$ where every $\varrho \in \chi_\mathcal{M}$ is assigned a perturbation interval of parameter valuation $\nu_{sRN}(\varrho) \in [min_\varrho, max_\varrho] \subseteq \mathbb{R}_{\geq 0}$ forming the perturbation space $\mathbf{P} \subseteq \mathcal{P}_{\chi_\mathcal{M}}$ as defined in Section 2.4.3. For every $\mu \in \mathbf{P}$, the dynamics of $\mathcal{M}_{\mathcal{P}_\chi}(\mu)$ is represented by a CTMC $\mathcal{C}_\mu = (\mathbb{X}, X_0, \mathbf{Q}_\mu)$ according to Definition 2.21. The parameterised sCRNM $\mathcal{M}$ is therefore represented by a family of CTMCs induced by the parameter perturbation space $\mathbf{P}$, $\mathbf{C} = \{\mathcal{C}_\mu \mid \mu \in \mathbf{P}\}$. The individual CTMCs in $\mathbf{C}$ differ only in the transition matrix that is affected by changing the parameter values.

2.4.6 Discrete Abstraction of Continuous Models

In order to enable the application of automatised global analysis methods to cBNMs, a model is approximated and abstracted in terms of a state-transition system that represents a finite quotient of the continuous model. The workflow of model approximation and abstraction is shown schematically in Figure 2.4 (left).

A cBNM is approximated by an intermediate model also represented in the semantics domain of ODEs but having a more regular structure that is suitable for rigorous abstraction [10]. In the general case, a piecewise multi-affine ODE (PWMA) is employed for the intermediate model. PWMA reflects a finite partition of the vector field provided that on every region of the partition the vector field is defined by a multi-affine function. For such kind of systems, rigorous abstraction methods have been developed, e.g., [135, 55, 49, 276, 120], allowing to formally over-approximate

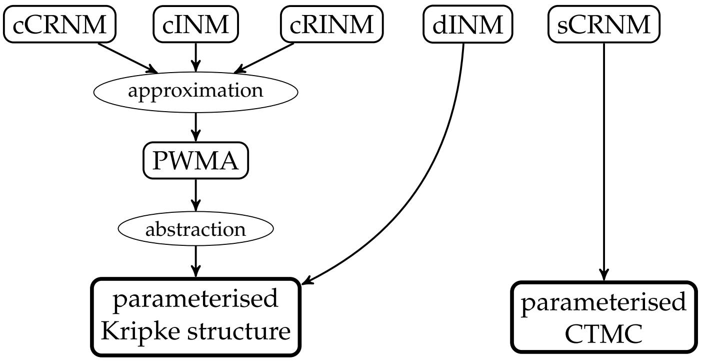


Figure 2.4: General workflow of encoding a parameterised models employed in the thesis.

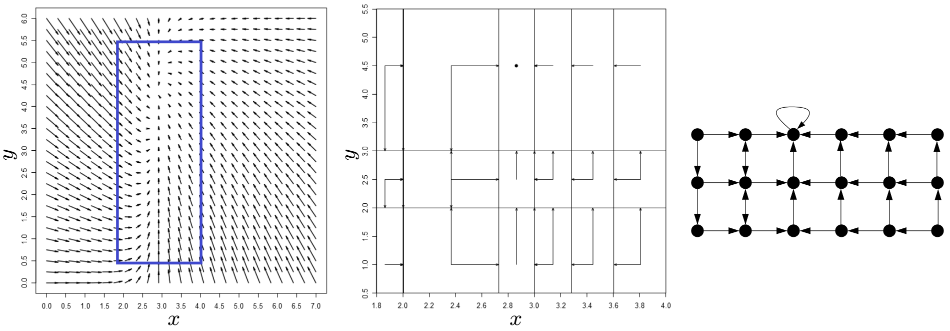


Figure 2.5: A vector field of an approximated ODE system (left) and discretisation of the emphasised region (middle). Thresholds determining the rectangles were obtained by the algorithm in [205]. States and transitions of the Kripke structure corresponding to the discretisation (right).

the continuous system by a finite state-transition system. In this thesis we employ rectangular abstraction methods introduced in [55, 49, 120], previously implemented in software tools RoVerGeNe [49] and BioDiVinE [35], and applied e.g. in [205, 46]. The abstraction is presented by using an encoding of the resulting parameterised state-transition relation in terms of SMT formulae. This allows us a compact description of the abstract parameterised model that copes with continuous parameter space. This encoding has been first proposed in our work [56].

Model Approximation

Semantics of a cBNM $\mathcal{M}$ of any type as defined in Section 2.3 has the general form of a continuous dynamical system $\dot{x} = f(x, p)$ where $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n_{\geq 0}$ where $n \in \mathbb{N}$ is the dimension of the system (determined by the cardi-

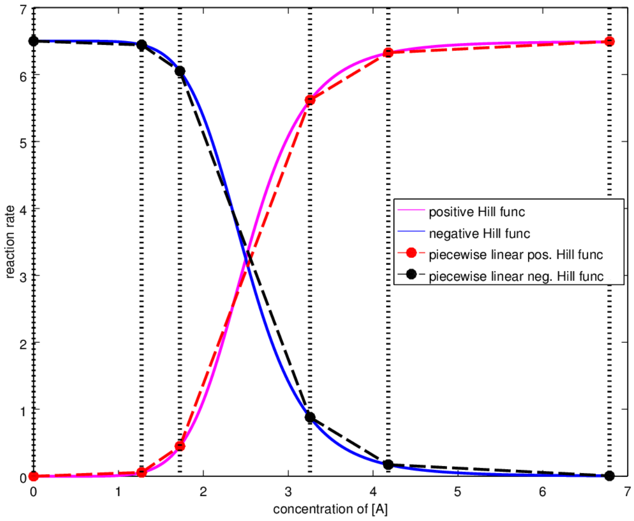


Figure 2.6: Optimal approximation of sigmoid functions by piece-wise affine functions according to [205].

nality of the set of species of the respective biological network) and $p \in \mathbb{R}^m$ is a parameter vector where $m \in \mathbb{N}$ is the number of parameters (determined by the cardinality of the parameterisation $\chi_{\mathcal{M}}$). The shape of the vector field f depends on the particular model type:

- If $\mathcal{M}$ is a cCRNM representing a network where all stoichiometry coefficients are 1 then for all $i \leq n$, f_i is multi-affine in x and affine in p. In case there is a stoichiometry coefficient higher than 1, the affected f_i is a polynomial of degree equal to the maximal stoichiometry coefficient. Affinity in p is preserved.

- If $\mathcal{M}$ is a cINM or cRINM then for all $i \leq n$, f_i is in general a non-linear function typically having the form of a rational function of x which is affine in p. Nevertheless, there can be components of x in which some f_i might be polynomial or even affine (e.g., a species in a reaction-influence network which does not take part in a complex affected by any influenced reaction).

The approximation is needed if the criterion that every f_i is multi-affine in x is violated. Apparently, this applies in cases of cCRNMs with a stoichiometry coefficient higher than 1 and in case of any cINM or cRINM that contains a non-linear regulation function (or a reaction with a reactant with stoichiometry greater than 1 in case of cRINMs). To achieve that we employ the approach defined in [205]. In particular, each regulation function occurring in f_i is approximated with an optimal piece-wise multi-affine function as shown in Figure 2.6 (in case of a one-dimensional regulation function the approximate function is piece-wise affine). In this procedure, a finite number of segments is introduced for every component of x. Segments partition the vector field orthogonally into a finite set of (hypher)rectangular regions. Analogous procedure is applied to polynomial functions representing kinetics of reactions with the stoichiometry of reactants greater than 1.

Model Abstraction

We assume that we are given a set of thresholds $\{\theta^i_1, \ldots, \theta^i_{n_i}\} \subset \mathbb{R}_{\geq 0}$ for each variable x_i such that $\theta^i_1 < \theta^i_2 < \cdots < \theta^i_{n_i}$. In case of a cINM or cRINM, the thresholds are given by exactly the points that separate individual segments of an approximated regulation function. However, in case of a cCRNM with stoichiometry coefficient at most 1 the selection of thresholds can be arbitrary. Further, we assume that the dynamical system satisfies for all $i \leq n$, f_i is multi-affine on every n-dimensional rectangle $[\theta^1_{j_1}, \theta^1_{j_1+1}] \times \cdots \times [\theta^n_{j_n}, \theta^n_{j_n+1}]$. Each rectangle is uniquely identified via an n-tuple of numbers: $R(j_1, \ldots, j_n) = [\theta^1_{j_1}, \theta^1_{j_1+1}] \times \cdots \times [\theta^n_{j_n}, \theta^n_{j_n+1}]$, where the range of each j_i is $\{1, \ldots, n_i - 1\}$. We also define $VR(j_1, \ldots, j_n)$ to be the set of all vertices of $R(j_1, \ldots, j_n)$.

Boundary Condition In order to establish a finite rectangular abstraction of the intermediate PWMA model, special care has to be given to boundary rectangles. A boundary rectangle is any rectangle $R(j_1, ..., j_n)$ where for some i either $j_i = 1$ or $j_i = n_i - 1$. Any dimension i satisfying that condition is called a boundary dimension of $R(j_1, ..., j_n)$. We restrict ourselves to models where the dynamics is bounded in the range specified by lower and upper thresholds – trajectories cannot exit that range (note that this could occur only in boundary rectangles). Formally, all trajectories determined by the PWMA model are required to keep $x_i \in [\theta^i_1, \theta^i_{n_i}]$. We restrict ourselves to parameter spaces where this requirement is satisfied for all parameter values. More precisely, we assume the so-called *boundary condition* to be satisfied: for every boundary rectangle $R(j_1, ..., j_n)$ we assume that for all $p \in \mathbb{P}, i \in \{1, ..., n\}, x \in R(j_1, ..., j_n)$ it holds that $(j_i = 1 \wedge x_i = \theta^i_1) \Rightarrow f_i(x, p) > 0$ and $(j_i = n_i - 1 \wedge x_i = \theta^i_{n_i}) \Rightarrow f_i(x, p) < 0$.

Abstraction-Encoding PKS There are two basic questions solved by the abstraction procedure. First, for a given rectangle we need to identify to which neighbouring rectangle the dynamics flows. Second, we need to check if there is a possibility to remain in the given rectangle for infinite time.

Technically, the rectangular abstraction results in an *(abstraction-encoding) parameterised Kripke structure* $\mathcal{K}^{abst} = (\mathcal{P}_{\chi_\mathcal{M}}, S, S_0, \rightarrow, L)$ with $S = \{(j_1, \dots, j_n) \mid \forall i : 1 \leq j_i \leq n_i\}$ where each $\alpha \in S$ represents the rectangle $R(\alpha)$ and $S_0 = \{\alpha_0\}$ such that $x_0 \in R(\alpha_0)$ (a rectangle corresponding to the initial condition x_0). The atomic propositions representing concentration inequalities are assigned to adequate states by means of the labelling L. In particular, $L : S \rightarrow 2^{AP}$ where $AP = \{x_i \odot \theta^i_j \mid 1 \leq i \leq n, 1 \leq j \leq n_i\}, \odot \in \{\leq, \geq\}\}$. The parameter space of $\mathcal{K}^{abst}$ is directly given by the parameter space of the abstracted model.

The parameterised transition relation $\rightarrow$ is defined between any two states representing neighbouring rectangles. Each transition is associated with a subset $\mathcal{P}(\alpha, \alpha')$ of parameter values under which it is enabled. In particular, the set $\mathcal{P}(\alpha, \alpha')$ is encoded symbolically by a formula $\Phi_{\alpha,\alpha'}$, $\mathcal{P}(\alpha, \alpha') = \{p \in \mathcal{P} \mid p \models \Phi_{\alpha,\alpha'}\}$, that expresses the condition that conservatively characterises continuous flows from $R(\alpha)$ to $R(\alpha')$ by investigating the vertices of the facet $R(\alpha) \cap R(\alpha')$.

Formally, Let $\alpha = (j_1, \dots, j_n) \in S$, $1 \leq i \leq n$ and $d \in \{-1, +1\}$. We denote $\alpha^{i,d} = (j_1, \dots, j_i + d, \dots, j_n)$ (if $j_i + d$ is in the valid range). Thus $\alpha^{i,d}$ describes all the neighbouring rectangles of α. We further denote $v^{i,+1}(\alpha) = VR(\alpha) \cap \{(..., j_i + 1, ...)\}$ and $v^{i,-1}(\alpha) = VR(\alpha) \cap \{(..., j_i, ...)\}$. For every pair of states $\alpha, \alpha^{i,d} \in S$, $1 \leq i \leq n$, $d \in \{-1, 1\}$ the formula $\Phi_{\alpha,\alpha^{i,d}}$ encoding the set of parameter values $\mathcal{P}(\alpha, \alpha^{i,d})$ is defined in the following way:

$$\Phi_{\alpha,\alpha^{i,d}} := \bigvee_{v \in v^{i,d}(\alpha)} d \cdot f_i(v, p) > 0$$

Additionally, the rectangular abstraction approximates the potential existence of a fixed point in any rectangle $\alpha \in S$. This is achieved by means of introducing a self-transition $\alpha \rightarrow \alpha$. In particular, a self-transition is enabled in a state $\alpha \in S$ for all parameter valuations $p \in \mathbb{P}$ satisfying $\mathbf{0} \in \mathit{hull}\{f(v, p) \mid v \in VR(\alpha)\}$ (the zero vector included in the convex hull of the rectangle vertices). This is symbolically encoded by the formula $\Phi_{\alpha,\alpha}$ defined in the following way:

$$\Phi_{\alpha,\alpha} := \exists c_1, \dots, c_k : \left(\bigwedge_{i=1}^{k} c_i \geq 0\right) \wedge \left(\sum_{i=1}^{k} c_i = 1\right) \wedge \left(\sum_{i=1}^{k} c_i \cdot f(v_i, p) = 0\right) \tag{2.1}$$

where $k = |VR(\alpha)|$ is the number of vertices of the rectangle α.

The formula $\Phi_{\alpha,\alpha}$ can be simplified with a coarser characterisation given by the following formula:

$$\begin{aligned}\Phi'_{\alpha,\alpha} := \neg \textstyle\bigvee_{1 \leq i \leq n} \big(& (\Phi_{\alpha^{i,-1},\alpha} \wedge \Phi_{\alpha,\alpha^{i,+1}} \wedge \neg\Phi_{\alpha,\alpha^{i,-1}} \wedge \neg\Phi_{\alpha^{i,+1},\alpha}) \\ & \vee (\neg\Phi_{\alpha^{i,-1},\alpha} \wedge \neg\Phi_{\alpha,\alpha^{i,+1}} \wedge \Phi_{\alpha,\alpha^{i,-1}} \wedge \Phi_{\alpha^{i,+1},\alpha})\big)\end{aligned}$$

The formula $\Phi'_{\alpha,\alpha'}$ is true just if there is either a pair of transitions $\alpha^{i,-1} \rightarrow \alpha \rightarrow \alpha^{i,+1}$ or a pair of transitions $\alpha^{i,+1} \rightarrow \alpha \rightarrow \alpha^{i,-1}$ provided that the respective two transitions are the only transitions allowed in ith dimension through the rectangle α. According to [49], this situation implies that the zero vector is not included in the convex hull of the vectors in rectangle vertices. That makes a necessary condition for non-existence of a fixed point inside the rectangle. In particular, $\Phi'_{\alpha,\alpha} \Rightarrow \neg\Phi_{\alpha,\alpha}$.

Interval-Based Encoding of Parameter Value Sets

In case every $f_i(x, p)$ depends on at most a single component in p (for every $i \leq 0$ there is *at most one unknown parameter occurring in* f_i, we call the parameters to be *mutually independent*), encoding of parameter values can be significantly simplified. In such case the set $\mathcal{P}(\alpha, \alpha')$ can be represented as a Cartesian product of closed intervals describing ranges of individual parameters values that enable the transition $\alpha \rightarrow \alpha'$. In a bounded parameter space these sets form (hypher)rectangular regions. In consequence, the representation of parameter sets, as well as the overall parameter synthesis procedure, can be significantly simplified.

Relation between a cBNM and its Discrete Abstraction

In [49, 55] it is shown that for a given multi-affine or even piece-wise multi-affine continuous system, the rectangular abstraction mentioned above conservatively abstracts almost all signals of the continuous system. A comprehensive overview of the conservativeness results for piece-wise affine and multi-affine systems is given in [120].

Formally, we consider a piece-wise multi-affine system (PWMA) $\dot{x} = f(x, p)$ defined on the bounded state space $\mathbb{X} = \prod_{i=1}^{n}[\theta_1^i, \theta_{n_i}^i] \subset \mathbb{R}_{\geq 0}$ where n is the dimension of the system, θ_1^i and $\theta_{n_i}^n$ is the lower and upper bound of variable x_i, and the parameter p ranges in a parameter space $\mathbb{P} \subset \mathbb{R}^m_{\geq 0}$ where m is the dimension of the parameter space. We consider we have obtained a PKS $\mathcal{K}^{abst}$ representing a rectangular abstraction of the PWMA system. We assume $\mathcal{K}^{abst}$ satisfies the boundary condition and is constructed using the steps mentioned above. Before we characterise the relation between $\mathcal{K}^{abst}$ and the abstracted PWMA system, we state the definition of signal-path correspondence.

Definition 2.31 *Let $\mathcal{K}^{abst} = (\mathcal{P}_{\mathcal{X}\mathcal{M}}, S, S_0, \rightarrow, L)$ be a parameterised Kripke structure encoding a rectangular abstraction of a PWMA system $\dot{x} = f(x, p)$ satisfying the boundary condition. We say a continuous* signal $\mathfrak{x} : T \rightarrow \mathbb{R}^n_{\geq 0}$ corresponds to a p-path $\pi_p = \alpha_0, \alpha_1, ...$ *of $\mathcal{K}^{abst}$ iff all the following conditions are satisfied:*

1. *$\mathfrak{x}$ satisfies $\mathfrak{x}(0) \in R(\alpha_0)$,*

2. *for any $t \in T$ there is $i \in \mathbb{N}_0$ such that $\mathfrak{x}(t) \in R(\alpha_i)$,*

3. *$\mathfrak{x}$ passes subsequently through the states of π_p.*

The relation between the semantics of the PWMA system and $\mathcal{K}^{abst}$ is characterised by the condition of *global sufficiency* [120] stated formally in the following claim.

Claim 2.32 *For any initial condition $x_0 \in \mathbb{X}$ and the respective continuous (exact) signal $\mathfrak{x}$ generated by a PWMA system $\dot{x} = f(x, p)$, $\mathfrak{x}(0) = x_0$, there exists a p-path π_p in $\mathcal{K}^{abst}$ such that $\mathfrak{x}$ corresponds to π_p.*

Remark 2.33 *The dual condition called global necessity has been also discussed in [55, 120]. It requires for any path π in $\mathcal{K}^{abst}$ there exist an initial condition x_0 and a corresponding signal $\mathfrak{x}$ in the PWMA system such that $\mathfrak{x}(0) = x_0$ and $\mathfrak{x}$ corresponds to π. However, it is shown that even for piece-wise affine systems the condition is not satisfied by the rectangular abstraction.*

The conservativeness of the abstraction guarantees the resulting Kripke structure provides an over-approximation of the original PWMA system. In particular, no behaviour is lost by the abstraction but false behaviour can be introduced.

Next, the relation between the original cBNM and the abstraction has to be discussed. Due to the fact the approximation step affects the shape of the signals, we cannot directly transfer the global sufficiency condition to the cBNM. Precise characterisation of the error produced by the approximation step is not yet available. However, it has been shown in [205] and also in our own work the vector field of the original cBNM and the approximated PWMA system can agree well if sufficient precision is used for local piece-wise linearisation of the regulation functions. The more significant problem is the extent of over-approximation – the number of false (so-called spurious) paths introduced in the abstraction is usually high. To that end, we have provided a variant of the procedure that combines the abstraction procedure with local numerical simulation [87]. However, the method is computationally demanding. Recently we have developed a method employing δ-decision SMT over formulae with reals [79] that embeds the δ-decision-based solver of ODEs [187] within the concept of rectangular abstraction.

Finally, we can summarise the overall procedure – a given parameterised cBNM $\mathcal{M}_{\mathcal{P}_\chi}$ with an initial condition x_0 and a parameterisation $\chi_\mathcal{M}$ is first approximated by a PWMA model and then abstracted in terms of a parameterised Kripke structure constructed as mentioned above and denoted $\mathcal{K}^{abst^\epsilon}(\mathcal{M}_{\mathcal{P}_\chi})$. The first step (approximation) means that for any given $\mu \in \mathcal{P}_{\chi_\mathcal{M}}$ the semantics $[\![\mathcal{M}_{\mathcal{P}_\chi}(\mu)]\!]_{\mathrm{cBNM}}$ represented by an exact signal $\mathfrak{x}_\mu$ is approximated with some (unspecified) numerical error ϵ by a signal $\mathfrak{x}_\mu^\epsilon$

generated by a PWMA model. In the second step (abstraction), the signal $\mathfrak{x}^{\epsilon}_{\mu}$ is over-approximated by a set of all paths π in $\mathcal{K}^{abst^{\epsilon}}(\mathcal{M}_{\mathcal{P}_{\chi}})_{\mu}$ such that $\pi(0) = \alpha_0$ and $x_0 \in R(\alpha_0)$. The resulting semantics of $\mathcal{M}_{\mathcal{P}_{\chi}}$ obtained after the approximation and abstraction steps is denoted $[\![\mathcal{M}_{\mathcal{P}_{\chi}}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}}$ and defined:

$$[\![\mathcal{M}_{\mathcal{P}_{\chi}}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}} \overset{df}{=} \mathcal{K}^{abst^{\epsilon}}(\mathcal{M}_{\mathcal{P}_{\chi}}).$$

Other Approaches

The main characteristics of the abstraction presented above are represented by two properties: (i) the abstract structure represents the behaviour of the dynamical system *globally* (it covers any initial condition of the original continuous system), (ii) the abstract structure captures parameter uncertainty by encoding the parameter values in terms of first-order formulae (it covers any parameter value in the considered domain of uncertainty). An alternative way to represent global behaviour together with parameter uncertainty is using polytopes as implemented in RoVerGeNe [49] and further refined in SpaceRover [67] (the tools are performing on cINMs but can be easily extended to work with general cRINMs). The approach introduced in [51] targets qualitative abstractions that are specific for cINMs. It relies on representing the regulatory function by means of discontinuous step functions resulting in piece-wise affine dynamics defined on a partition that is given symbolically by qualitative (numerically unspecified) thresholds. The method is implemented in the GNA tool [135].

A universal global approach to the finite representation of dynamical (and even hybrid) systems under parameter uncertainty is to encode the models entirely in terms of first-order formulae over real numbers and solve them using a δ-decision SMT solver [156, 187, 252]. This provides an implicit (relational) abstraction of the continuous dynamics flow [323]. Such a relation gives a unique symbolic representation of the system dynamics in the given bounded domain. However, in the case of global analysis (e.g., large and complex initial sets and the need to explore the entire bounded domain of systems dynamics), these techniques quickly become computationally demanding. We believe that techniques that will combine discrete abstractions with symbolic approaches have high potential.

Abstractions of nonlinear systems based on interval analysis have been introduced in [344, 313]. Although they are guaranteed to give over-approximations to the discrete dynamics of the continuous system (sufficiency), the quality of the approximation can be (and typically is) rather poor.

A technique close to the approach employed here is presented in [65]. Rather than using a fixed polyhedral subdivision, a simplicial subdivision is computed based on the system itself, and the resulting flow induced multi-valued map provides a good abstraction of the continuous system.

However, a good polyhedral subdivision itself is extremely difficult to compute, and the method is likely to be useful only in low dimensions.

There is a group of approaches based on qualitative reasoning (theorem proving) with predicates over the reals [305, 350]. In that case, the construction is not fully automatised and it is challenging to find suitable abstraction predicates for non-linear systems.

Several techniques allowing hybridisation of the continuous dynamics have been introduced, e.g., [18, 129]. These techniques follow the dynamics with a significantly less extent of false behaviour but they are restricted to bounded-time reachability problems. In [66] the authors employ conic abstraction restricted to affine systems that allows unbounded reachability analysis. For multi-affine systems, conic abstractions have been combined with rectangular partitioning to reduce the extent of overapproximation [60].

The techniques presented in [101, 276, 364] employ timed automata in various forms for the abstraction. An abstraction method allowing to capture unbounded-time behaviour is addressed in [101].

For certain problems, polynomial boundaries (barrier certificates [308] or multivariate polynomial partitioning [250]) can be employed. Such methods give definitely more precise results than polyhedral partitioning but are computationally demanding and not yet applicable to models with uncertain parameters.

2.5 Temporal Logics for Biological Systems

In this subsection we give an overview of temporal logics that are considered relevant for expressing properties of biological systems. The mentioned set of properties is not complete but it rather focuses on the types of logics used in our research. Logics used in following chapters are described formally.

Qualitative Logics

With respect to the level of abstraction employed, there are two classes of temporal properties. Qualitative properties abstract away from any quantitative information like time aspects or energy costs of systems dynamics. Qualitative properties are in general interpretable on all types of models, especially on untimed discrete-value models (dINMs and discrete abstractions of cBNMs). Two basic logical formalisms allowing to express qualitative properties of systems dynamics are the linear-time temporal logic (LTL) [306], interpreted on individual model executions (paths), and computation tree logic (CTL) [117], interpreted on trees of (non-deterministically) branching model executions.

Quantitative properties are usually expressed in formalisms based on the aforementioned temporal logics. They can be roughly divided into deterministic (applicable to cRINMs, cCRNMs and cINMs) and stochastic logics (applicable to sCRNMs). Deterministic logics are mostly focusing on a quantitative notion of time. The time extension of CTL called Timed Computational Tree Logic (TCTL) has been introduced in [7].

Quantitative Logics

Metric Temporal Logic (MTL) [254] is general quantitative temporal logic that allows to quantify modalities with the time frame represented by a closed time interval. MTL possesses both discrete and continuous semantics, as it can be interpreted on infinite timed state sequences as well as continuous signals. The possibility of singular values occurring as time-quantifiers of the modalities causes the problem of formula satisfiability to be undecidable for MTL. To that end, Metric Interval Logic (MITL) has been introduced in [8] as a practical restriction of MTL that allows only non-singular intervals to quantify the modalities. Another time extension of LTL called Timed Propositional Temporal Logic (TPTL) [9] is based on freeze-quantification where extra clocks are used to specify temporal constraints. Motivated by the application of verification and monitoring techniques to continuous-value and hybrid systems, Signal Temporal Logic (STL) has been introduced [277]. It combines the dense time modalities of MITL with the numerical predicates over real numbers. Technically, the logic is interpreted over piece-wise linear continuous-time signals on which a qualitative and a quantitative semantics [151] is defined including an efficient algorithm [149].

A version of LTL with constraints over the reals, called LTL($\mathbb{R}$), has been proposed in [15] to express temporal properties of molecular concentrations and their derivatives. A quantifier-free fragment of the first-order extension of LTL($\mathbb{R}$), called QFLTL($\mathbb{R}$), has been considered in [163]. It allows to use free variables in the atomic propositions and, thus, it enables to analyse robustness of numerical data time series with temporal logic and to automatically compute LTL($\mathbb{R}$) specifications from experimental traces.

Stochastic logics express the probability and performance measures on Markov chains. To formalise properties of CTMCs, Continuous Stochastic Logic (CSL) [20] has been introduced. It is a probabilistic extension of CTL with continuous-time semantics. To further broaden the scope of possibly expressible behaviour, CSL have been extended to allow the specification over reward-based stochastic models, i.e., Markov chains with real-valued rewards/costs attached to states and transitions [260]. The extension enables to express properties such as the expected time a system spends in a specified set of states over a time interval or the expected number of times that a particular reaction occurs.

Extended Logics

Expressing biological phenomena can require extensions of existing logics. Biologically relevant temporal logic extensions target precise quantitative description of oscillations [143, 43] or qualitative properties combining linear-time properties with branching-time [280]. In our work we have introduced two extensions of temporal logics. In the domain of linear-time logics we have addressed an extension of STL by means of introducing a value-freezing operator that allows to store a signal value at the time point where the operator is evaluated. The stored value can be then referred in predicates placed in the scope of the value-freezing operator. This allows STL* to specify and distinguish various dynamic aspects which occur in biological systems, in addition to the phenomena mentioned above, these can be, e.g., damped oscillations or local extremes in species concentration.

In the domain of branching-time logics, we have combined two known extensions of CTL – an extension HCTL adding hybrid operators including past operators and allowing to use of state variables that can be fixed in certain parts of the formula as well as quantified [16], and an extension UCTL adding event predicates over single-step system evolutions [346]. The resulting logic called HUCTL [58] allows to efficiently express global and local properties of phase spaces of dynamical systems that cannot be expressed in LTL/CTL, e.g., the presence of a given number of mutually exclusive stable attractors. The need for hybrid branching-time logics in the domain of biological systems has been also addressed in [17].

In the remaining part of this section, we describe in more detail the logics that are selected to be used in following chapters of the thesis.

2.5.1 Linear Time Temporal Logic – LTL

LTL captures the temporal properties of paths in discrete state-transition systems. In particular, LTL formulae are interpreted on infinite paths generated by a Kripke structure.

LTL formulae are defined by the following abstract syntax:

$$\varphi ::= Q \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \mathbf{U} \varphi_2$$

where Q ranges over atomic propositions taken from a set AP. We use the standard abbreviations like $\mathbf{F}\varphi$ which stands for $true\mathbf{U}\varphi$ or $\mathbf{G}\varphi$ which stands for $\neg\mathbf{F}\neg\varphi$.

The semantics of an LTL formula φ is interpreted on infinite paths of a Kripke structure $\mathcal{K} = (S, S_0, \rightarrow, L)$. For an infinite path $\pi = s_0 s_1 ...$ and some $i \in \mathbb{N}_0$ we use the notation π^i to denote the infinite path $\pi^i = s_i s_{i+1} ...$ and the notation $\pi(i)$ to denote the state s_i.

$\pi \models_{\mathcal{K}} Q$ iff $Q \in L(\pi(0))$

$\pi \models_{\mathcal{K}} \neg\varphi$ iff $\pi \not\models_{\mathcal{K}} \varphi$
$\pi \models_{\mathcal{K}} \varphi_1 \wedge \varphi_2$ iff $\pi \models_{\mathcal{K}} \varphi_1$ and $\pi \models_{\mathcal{K}} \varphi_2$
$\pi \models_{\mathcal{K}} \mathbf{X}\,\varphi$ iff $\pi^1 \models_{\mathcal{K}} \varphi$
$\pi \models_{\mathcal{K}} \varphi_1 \mathbf{U} \varphi_2$ iff $\exists i \in \mathbb{N}_0$ such that $\pi^i \models \varphi_2$ and $\forall j < i.\pi^j \models \varphi_1$

We say a Kripke structure $\mathcal{K}$ satisfies φ, written $\mathcal{K} \models \varphi$, iff for all $s \in S_0$ it holds that every path π in $\mathcal{K}$ such that $\pi(0) = s$ satisfies $\pi \models_{\mathcal{K}} \varphi$. This is called a universal interpretation of LTL.

Examples of some typical LTL formulae used for biological systems are [165]:

- $\mathbf{F}[\varphi]$ expresses a reachability of a state where the condition φ holds,
- $\mathbf{G}[\varphi]$ expresses a stabilisation with φ being continually true,
- $[\varphi_1 \Rightarrow \mathbf{F}\varphi_2] \wedge [\varphi_2 \Rightarrow \mathbf{F}\varphi_1]$ expresses permanent oscillation between φ_1 and φ_2.

2.5.2 Computation Tree Temporal Logic – CTL

The key characteristics of CTL is it captures the branching behaviour of discrete state-transition systems. More precisely, CTL formulae are interpreted on states of a Kripke structure.

CTL formulae are defined by the following abstract syntax:

$$\varphi ::= Q \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{AX}\varphi \mid \mathbf{EX}\varphi \mid \mathbf{AU} \mid \mathbf{EU}$$

where Q ranges over atomic propositions taken from a set AP. We denote by $cl(\varphi)$ the set of all subformulae of φ. We use the standard abbreviations like $\mathbf{EF}\varphi$ which stands for $\mathbf{E}(true\,\mathbf{U}\,\varphi)$ or $\mathbf{AG}\varphi$ which stands for $\neg\mathbf{EF}\,\neg\varphi$.

The semantics of a CTL formula φ is interpreted on states of a Kripke structure $\mathcal{K} = (S, S_0, \rightarrow, L)$. For every state $s \in S$ and a particular form of the formula the semantics is given in the following way:

$s \models_{\mathcal{K}} Q$ iff $Q \in L(s)$
$s \models_{\mathcal{K}} \neg\varphi$ iff $s \not\models_{\mathcal{K}} \varphi$
$s \models_{\mathcal{K}} \varphi_1 \wedge \varphi_2$ iff $s \models_{\mathcal{K}} \varphi_1$ and $s \models_{\mathcal{K}} \varphi_2$
$s \models_{\mathcal{K}} \mathbf{AX}\,\varphi$ iff for all π in $\mathcal{K}$ such that $\pi(0) = s$ it holds that $\pi(1) \models_{\mathcal{K}} \varphi$
$s \models_{\mathcal{K}} \mathbf{EX}\,\varphi$ iff there exists π in $\mathcal{K}$ such that $\pi(0) = s$ and $\pi(1) \models_{\mathcal{K}} \varphi$
$s \models_{\mathcal{K}} \mathbf{A}(\varphi_1\,\mathbf{U}\,\varphi_2)$ iff for every π in $\mathcal{K}$ such that $\pi(0) = s$ there exists
$i \in \mathbb{N}_0$ such that $\pi(i) \models \varphi_2$ and $\forall j < i.\pi(j) \models \varphi_1$
$s \models_{\mathcal{K}} \mathbf{E}(\varphi_1\,\mathbf{U}\,\varphi_2)$ iff there exists π in $\mathcal{K}$ such that $\pi(0) = s$ and
$\exists i \in \mathbb{N}_0$ such that $\pi(i) \models \varphi_2$ and $\forall j < i.\pi(j) \models \varphi_1$

We say a Kripke structure $\mathcal{K}$ satisfies φ, written $\mathcal{K} \models \varphi$, iff for all $s \in S_0$, $s \models_{\mathcal{K}} \varphi$.

Examples of some typical CTL formulae used for biological systems are [165]:

- $\mathbf{EF}[\varphi]$ expresses a reachability of a state where the condition φ holds,
- $\mathbf{AG}[\varphi]$ expresses a stabilisation with φ being continually true,
- $\mathbf{EF}[\mathbf{AG}[\varphi_1]] \wedge \mathbf{EF}[\mathbf{AG}[\varphi_2]]$ expresses a bistable switch (two different stable situations φ_1, φ_2 can be reached).

Most frequent types of temporal properties investigated for biochemical models have been collected in [291]. There are two important fragments of CTL relevant for biological models. A formula is said to be *positive* if it does not contain any negations. We say that a formula is *existential* (or in ECTL) if it is positive and only contains existential temporal operators. We say that a formula is *universal* (or in ACTL) if it is positive and only contains universal temporal operators.

2.5.3 Signal Temporal Logic – STL

STL is interpreted on continuous signals. It has been introduced in [277] to address the needs to specify the temporal properties of real-valued signals. Due to the deterministic nature of signals, STL is entirely based in the linear time framework. It is worth noting that LTL cannot be directly employed for continuous structures as it inherently targets discrete paths (e.g., there is no direct meaning of the operator **X** (next) in the continuous settings). To that end, STL employs principles of linear temporal logics quantitatively addressing time (MTL [254] and MITL [8]) and refines them with predicates evaluated over real values. Technically, STL is based on MITL where the time quantification is restricted to non-singular closed intervals due to decidability reasons. The notion specific for STL is a *predicate* defined in the following way.

Definition 2.34 *Let $\mathfrak{x} \in (\mathbb{R}^n_{\geq 0})^T$ be a signal for some $n \in \mathbb{N}$ and $\tau \in T$ a given time point. A* predicate $\mathfrak{p}$ *is defined as a generic constraint applied to a signal value in the time point τ:*

$$\mathfrak{p}(\mathfrak{x}, \tau) > 0$$

It can be interpreted as a subset of the signal value domain $\mathbb{R}^n_{\geq 0}$ satisfying the given constraint.

The syntax of STL is the following:

Definition 2.35 *Syntax of STL is defined by the following grammar:*

$$\varphi ::= \mathfrak{p} \mid true \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathbf{U}_I \varphi_2$$

where $\mathfrak{p}$ is a predicate as of Definition 2.34 and $I \subseteq \mathbb{R}_{\geq 0}$ a closed non-singular interval.

The time-quantified versions of operators $\mathbf{F}$ and $\mathbf{G}$ are defined in the standard way: $\mathbf{F}_I\varphi = true\mathbf{U}_I\varphi$, $\mathbf{G}_I\varphi = \neg\mathbf{F}_I\neg\varphi$.

Qualitative semantics is defined with respect to the structure of the formula as follows.

Definition 2.36 *Let $\mathfrak{x} \in (\mathbb{R}^n_{\geq 0})^T$ be a signal for some $n \in \mathbb{N}$ and $\tau \in T$ a time point.* Qualitative satisfaction of an *STL* formula *φ is defined inductively:*

$$\begin{array}{lll}
(\mathfrak{x}, t) \models true & & \\
(\mathfrak{x}, t) \models \mathfrak{p} & \iff & \mathfrak{x}(t) \in \mathfrak{p} \\
(\mathfrak{x}, t) \models \neg\varphi & \iff & (\mathfrak{x}, t) \not\models \varphi \\
(\mathfrak{x}, t) \models \varphi_1 \vee \varphi_2 & \iff & (\mathfrak{x}, t) \models \varphi_1 \vee (\mathfrak{x}, t) \models \varphi_2 \\
(\mathfrak{x}, t) \models \varphi_1\mathbf{U}_I\varphi_2 & \iff & \exists\, t' \in t \oplus I : (\mathfrak{x}, t') \models \varphi_2 \wedge \\
 & & \forall\, t'' \in [t, t'] : (\mathfrak{x}, t'') \models \varphi_1
\end{array}$$

where $t \oplus I$ stands for $\{t + u \mid u \in I\}$.

Further we say $\mathfrak{x}$ satisfies φ, written $\mathfrak{x} \models \varphi$, iff $(\mathfrak{x}, 0) \models \varphi$.

Boolean answer provided by the qualitative semantics is not fully compatible with quantitative entities in the continuous domain where real-valued distance functions play an important role. To that end, the quantitative semantics of STL has been introduced in [151]. It is based on measuring the distance between the signal value satisfying a formula and the closest value violating the formula. In the case of a predicate, such distance is given directly by the value of the predicate evaluated at a particular time point. For a general STL formula, the quantitative semantics is defined in the following way:

Definition 2.37 *Let $\mathfrak{x} \in (\mathbb{R}^n_{\geq 0})^T$ be a signal for some $n \in \mathbb{N}$ and $\tau \in T$ a time point.* Quantitative satisfaction of an STL formula *φ is defined inductively:*

$$\begin{array}{ll}
\rho(true, \mathfrak{x}, t) & = +\infty \\
\rho(\mathfrak{p}, \mathfrak{x}, t) & = \mathfrak{p}(\mathfrak{x}(t), t) \\
\rho(\neg\varphi, \mathfrak{x}, t) & = -\rho(\varphi, \mathfrak{x}, t) \\
\rho(\varphi_1 \vee \varphi_2, \mathfrak{x}, t) & = \max\left(\rho(\varphi_1, \mathfrak{x}, t), \rho(\varphi_2, \mathfrak{x}, t)\right) \\
\rho(\varphi_1\mathbf{U}_I\varphi_2, \mathfrak{x}, t) & = \max\limits_{t' \in t \oplus I} \min\left(\rho(\varphi_2, \mathfrak{x}, t'), \min\limits_{t'' \in [t, t']} \rho(\varphi_1, \mathfrak{x}, t'')\right)
\end{array}$$

where $t \oplus I$ stands for $\{t + u \mid u \in I\}$.

Further we define quantitative satisfaction of φ on a signal $\mathfrak{x}$, *written $\rho(\varphi, \mathfrak{x})$, as $\rho(\varphi, \mathfrak{x}) \overset{df}{=} \rho(\varphi, \mathfrak{x}, 0)$.*

The notion of quantitative semantics mentioned above addresses the so-called space robustness as the measure is defined on signal values (a change in quantitative semantics value is sensitive to shifting the signal in the value domain). In [151] the authors also consider the notion of time robustness where the measure is defined on the time points (a change in quantitative semantics value is sensitive to shifting the signal in the time domain). The results in this thesis address the space-based view of robustness as it seems to be a more relevant issue in biological systems in general.

Examples of STL formulae relevant for biological models are the following:

- $\mathbf{G}_{[0,300]}[(x_1 > 0.7) \Rightarrow F_{[3,5]}(x_2 > 0.7)]$ expresses that for each time point $t \in [0, 300]$ it holds that if the value of x_1 in t is greater than 0.7 then there exists time $t' \in [t+3, t+5]$ such that the value of x_2 in t' is also greater than 0.7,
- $\mathbf{G}_{[0,\tau]}[((x \geq max_x) \Rightarrow \mathbf{F}_{[p_1,p_2]}(x \leq min_x)) \wedge ((x \leq min_x) \Rightarrow \mathbf{F}_{[p_1,p_2]}(x \geq max_x))]$ expresses a periodical oscillation of x in the time horizon τ such that the minimal value min_x and the maximal value max_x are interchangeably being reached with a period fluctuating in the interval $[p_1, p_2]$.

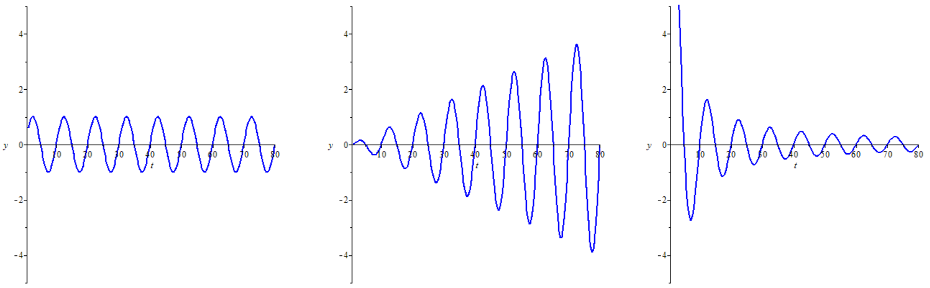
2.5.4 Value-Freezing Signal Temporal Logic – STL*

In STL it is not possible to express (and distinguish) the classes of signal oscillations such as damped oscillations or oscillations with increasing amplitude. The reason is the impossibility of globally referencing (and relatively comparing) concrete signal values occurring in time points in which some local property is satisfied. STL does not provide any constructs allowing to omit references to concrete values in predicates. E.g., the damped oscillation in Figure 2.7c can be expressed in STL as a sequence exactly reaching the 15 local extremes in the given order. In general, there can appear any number of local extremes in the observed time interval. Such a general property cannot be expressed in STL. To that end, we have introduced the logic STL* that allows referring signal values reached in the past.

As in the case of STL, a formula of STL* expresses a temporal property of finite-time continuous signals. The crucial phenomenon introduced in STL* is the concept of value-freezing. Signal value freezing is facilitated by the notion of *frozen time vector* defined in the following definition. The structure is used to store time values at various time points which then can be referred in predicates.

Definition 2.38 *Let $\mathcal{I}$ be a freezing operators index set.* Frozen time vector t^* *is a function:*

$$t^* : \mathcal{I} \rightarrow \mathbb{R}_{\geq 0}$$



(a) $x(t) = sin(\frac{2\pi t}{10})$

(b) $x(t) = \frac{t}{20} sin(\frac{2\pi t}{10})$

(c) $x(t) = \frac{20}{t} sin(\frac{2\pi t}{10})$

Figure 2.7: Various types of oscillations: (a) a permanent oscillation with constant amplitude, (b) oscillation with increasing amplitude, (c) damping oscillation.

The symbol $t_i^* = t^*(i)$ is referred to as i-th frozen time.

Predicates comprise Boolean expressions over values of a signal $\mathfrak{x}$ at time t and each frozen time t_i*, where x_j denotes the j-th component of the signal at time t, i.e. $\mathfrak{x}(t) = (x_1, \ldots, x_j, \ldots, x_n)$, and $x_j^{*_i}$ the j-th component at time t_i^*. When $|\mathcal{I}| = 1$, we usually write x_i^* instead of $x_i^{*_1}$.

In STL* only predicates given by linear inequalities are considered. This is an important assumption employed in Section 6.2 (analytical expressions of robustness can be efficiently computed for linear predicates). This restriction is sufficient in the majority of practical cases.

Definition 2.39 *Let* $n \in \mathbb{N}$*,* $b \in \mathbb{R}$ *and* $a_{ij} \in \mathbb{R}$ *where* $i \in \{0\} \cup \mathcal{I}$*,* $j \in \{1, \ldots, n\}$ *and not all* a_{ij} *are zero. A* predicate *is defined as a subset of* $\mathbb{R}^n \times (\mathbb{R}^n)^{\mathcal{I}}$ *such that:*

$$\sum_{j=1}^{n} a_{0j} x_j + \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{n} a_{ij} x_j^{*_i} + b \geq 0$$

Freeze operator is used to store the time point into frozen time vector, thus facilitating signal value freezing. The following definition introduces an auxiliary concept of storing the current time t as the ith component of the frozen time vector.

Definition 2.40 *Let* t^* *be a frozen time vector,* $i, j \in \mathcal{I}$ *and* $t \in \mathbb{R}_{\geq 0}$*.* Freezing ith component of t^* in t *is denoted as* $t^*[i \leftarrow t]$ *and defined:*

$$t^*[i \leftarrow t](j) = \begin{cases} t & i = j \\ t_j^* & i \neq j \end{cases}$$

Definition 2.41 *Let* $\mathcal{I}$ *denote a freezing-operators index set. Syntax of STL* is defined by the following grammar:*

$$\varphi ::= \mathfrak{p} \mid true \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathbf{U}_I \varphi_2 \mid *_i \varphi$$

where $i \in \mathcal{I}$, $true$ denotes the true constant, $\mathfrak{p}$ is a predicate as of Definition 2.39 and $I \subseteq \mathbb{R}_{\geq 0}$ a closed non-singular interval.

Note that all Boolean connectives and temporal operators $\mathbf{F}$ and $\mathbf{G}$ can be defined using the basic operators defined above in the same way as is done in STL. Similarly to predicates, when $|\mathcal{I}| = 1$, we usually omit the index of freeze operator, as in $* \, \mathbf{G}_I(x > x^*) = *_1 \, \mathbf{G}_I(x > x^{*_1})$.

Definition 2.42 *Let $\mathfrak{x} \in \left(\mathbb{R}^n_{\geq 0}\right)^T$ be a signal, $t \in T$ a time point and $t^* \in T^{\mathcal{I}}$ a frozen time vector. Formula satisfaction is defined inductively:*

$$\begin{array}{lcl}
(\mathfrak{x}, t, t^*) \models true & & \\
(\mathfrak{x}, t, t^*) \models \mathfrak{p} & \iff & (\mathfrak{x}(t), \mathfrak{x} \circ t^*) \in \mathfrak{p} \\
(\mathfrak{x}, t, t^*) \models \neg\varphi & \iff & (\mathfrak{x}, t, t^*) \not\models \varphi \\
(\mathfrak{x}, t, t^*) \models \varphi_1 \vee \varphi_2 & \iff & (\mathfrak{x}, t, t^*) \models \varphi_1 \vee (\mathfrak{x}, t, t^*) \models \varphi_2 \\
(\mathfrak{x}, t, t^*) \models \varphi_1 \mathbf{U}_I \varphi_2 & \iff & \exists\, t' \in t \oplus I : (\mathfrak{x}, t', t^*) \models \varphi_2 \wedge \\
 & & \forall\, t'' \in [t, t'] : (\mathfrak{x}, t'', t^*) \models \varphi_1 \\
(\mathfrak{x}, t, t^*) \models *_i \, \varphi & \iff & (\mathfrak{x}, t, t^*[i \leftarrow t]) \models \varphi
\end{array}$$

Operator $\circ$ is used to denote function composition, i.e. $(\mathfrak{x} \circ t^*) \in \left(\mathbb{R}^n_{\geq 0}\right)^{\mathcal{I}}$ and $(\mathfrak{x} \circ t^*)(i) = \mathfrak{x}(t^*_i)$ and $t \oplus I$ stands for $\{t + u \mid u \in I\}$.

We say a a signal $\mathfrak{x}$ satisfies a formula φ, *written* $\mathfrak{x} \models \varphi$, *iff* $(\mathfrak{x}, 0, \mathbf{0}) \models \varphi$ *where* $\mathbf{0}$ *denotes the* zero frozen time vector, *i.e.,* $\mathbf{0} \overset{df}{=} \mathcal{I} \times \{0\}$.

Intuitively, interpretation of $*_i \, \varphi$ is the following: freeze operator stores signal values at the time of $*_i \, \varphi$ evaluation, which can then be referred to using index i in predicates of φ. An example property, *"in the next five time units, x increases by* 8*"* can be specified as:

$$* \, \mathbf{F}_{[0,5]}(x \geq x^* + 8)$$

where x^* refers to value of x at time 0.

Quantitative semantics of STL* is described in detail in Chapter 6 as a part of a mechanism for robustness analysis of cBNMs.

Representative examples of STL* formulae are the following:

- $\mathbf{G}_{[0,80-(4+\delta)]} * [\mathbf{G}_{[4-\delta,4+\delta]}(y^* = x)]$, for some small $\delta > 0$, is an example of delayed correlation of two signals — the signal x copies the values of the signal y with a delay of 4 seconds,

- $\mathbf{G}_{[0,60]}[\mathbf{F}_{[0,10]} * [\mathbf{G}_{[1,10]}(x^* \geq x + c)] \wedge \mathbf{F}_{[0,10]} * [\mathbf{G}_{[1,10]}(x^* \leq x - c)]$ expresses a damping oscillation (such as depicted in Figure 2.7c) — *there is always a time instant in near future which is a local maximum for some future interval and there is also another time instant in near future which is a local minimum for some future interval* (the constant c sets the strength of damping).

2.5.5 Continuous Stochastic Logic – CSL

Let $\mathcal{C} = (\mathbb{X}, X_0, \mathbf{Q}, L)$ be a CTMC extended with a labelling function L which assigns to each state $X \in \mathbb{X}$ the set $L(X)$ of atomic propositions that are valid in state X. We consider a bounded time fragment of CSL with rewards.

The syntax of CSL is defined in the following way. A state formula φ is given as

$$\varphi ::= true \mid q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{P}_{\sim p}[\phi] \mid \mathsf{R}_{\sim r}[\mathsf{C}^{\leq t}] \mid \mathsf{R}_{\sim r}[\mathsf{I}^{=t}]$$

where ϕ is a path formula given as $\phi ::= \mathbf{X}\varphi \mid \varphi\, \mathbf{U}^I\, \varphi$, q is an atomic proposition, $\sim \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$ is a probability, $r \in \mathbb{R}_{\geq 0}$ is an expected reward and $I = [a, b]$ is a bounded time interval such that $a, b \in \mathbb{R}_{\geq 0} \wedge a \leq b$. Path operators $\mathbf{G}$ (always) and $\mathbf{F}$ (eventually) are derived in the standard way using the operator $\mathbf{U}$. In order to specify properties containing rewards ($\mathsf{R}_{\sim r}[\mathsf{C}^{\leq t}]$ is the *cumulative reward* acquired up to time t, $\mathsf{R}_{\sim r}[\mathsf{I}^{=t}]$ is the *instantaneous reward* in time t) the CTMC $\mathcal{C}$ is enhanced with reward (cost) structures. Two types of reward structures are used. *A state reward* $rew(X)$ defines the rate with which a reward is acquired in state $X \in \mathbb{X}$. A reward of $t \cdot rew(X)$ is acquired if a CTMC remains in state X for t time units. *A transition reward* $trew(X_i, X_j)$ defines the reward acquired each time the transition $X_i \to X_j$ occurs.

The formal semantics of the bounded fragment of CSL with rewards is defined similarly as the semantics of full CSL [20]. The key part of the semantics is given by the definition of the satisfaction relation $\models$. It specifies when a state X satisfies the state formula φ (denoted as $X \models \varphi$) and when a path π satisfies the path formula ϕ (denoted as $\pi \models \phi$). Let us recall the notation $Path^{\mathcal{C}}(X)$ denoting the set of all paths starting at X (Definition 2.21). The informal definition of $\models$ is as follows:

- $X \models \mathsf{P}_{\sim p}[\phi]$ iff the probability of all paths $\pi \in Path^{\mathcal{C}}(X)$ that satisfy the path formula ϕ (denoted as $Prob^{\mathcal{C}}(X, \phi)$) satisfies $\sim p$, where
  - π satisfies $\mathbf{X}\,\varphi$ iff the second state on π satisfies φ
  - π satisfies $\varphi\, \mathbf{U}^I\, \Psi$ iff there exists a time instant $t \in I$ such that the state on π occupied at t satisfies Ψ and all states on π occupied before $t' \in [0, t)$ satisfy φ
- $X \models \mathsf{R}_{\sim r}[\mathsf{C}^{\leq t}]$ iff the sum of expected rewards over $Path^{\mathcal{C}}(X)$ *cummulated* until t time units (denoted as $Exp^{\mathcal{C}}(X, \mathbf{X}_{\mathsf{C}^{\leq t}})$) satisfies $\sim r$
- $X \models \mathsf{R}_{\sim r}[\mathsf{I}^{=t}]$ iff the sum of expected rewards over all paths $\pi \in Path^{\mathcal{C}}(X)$ at time t (denoted as $Exp^{\mathcal{C}}(X, \mathbf{X}_{\mathsf{I}^{=t}})$) satisfies $\sim r$.

A set $Sat_{\mathcal{C}}(\varphi) = \{X \in \mathbb{X} \mid X \models \varphi\}$ denotes the set of states that satisfy φ.

The syntax and semantics can be easily extended with "quantitative" formulae in the form $\varphi ::= \mathsf{P}_{=?}[\phi] \mid \mathsf{R}_{=?}[\mathsf{C}^{\leq t}] \mid \mathsf{R}_{=?}[\mathsf{I}^{=t}]$, i.e., the topmost operator of the formula φ returns a quantitative result, as used, e.g., in PRISM [261]. In this case the result of a decision procedure is not in the form of a Boolean yes/no answer but is the actual numerical value of the probability $Prob^{\mathcal{C}}(X, \phi)$ or the expected reward $Exp^{\mathcal{C}}(X, \mathbf{X})$ for $\mathbf{X} \in \{\mathbf{X}_{\mathsf{I}^{=t}}, \mathbf{X}_{\mathsf{C}^{\leq t}}\}$. The computation of a numerical value is of the same complexity class as the computation of a result to be compared leading to a Boolean answer, although in some cases the comparison may be carried out on less precise or preliminary results.

Examples of CSL formulae of the considered fragment are the following:

- $\mathsf{P}_{\geq 0.9}[\mathbf{F}^{[1,2]}(A = 30)]$ expresses that the probability that the species A will be represented by 30 molecules between 1 and 2 time units is at least 0.9,
- $\mathsf{P}_{=?}[\mathbf{G}^{[500,1000]}(A < 30)]$ expresses the probability that the number of molecules of species A keeps below 30 within the time window between 500 and 1000 time units,
- $\mathsf{R}_{=?}[\mathsf{C}^{\leq 1000}](A > 20)$ expresses the fraction of time the system exhibits more than 20 molecules of A within the time horizon 1000 time units.

2.5.6 Interpretation on Biological Network Models

In this subsection, we relate semantics of temporal logics presented above with the model classes considered in Section 2.3. For each model class, we define the basic notion of formula satisfaction in a given model and we refine the notion to the settings of the parameterised variant of the particular model class.

dINM

Considering the logics described in detail in previous subsections, only basic temporal logics LTL and CTL can be directly interpreted on dINMs. Let us consider a dINM $\mathcal{M} = (\mathcal{N}, \nu_{dIN}, \mathbb{X}, X_0, \mathbf{m})$ and an LTL or CTL formula φ. The transition system $[\![\mathcal{M}]\!]_{\mathrm{dINM}} = (\mathbb{X}, T, X_0)$ is turned into a Kripke structure $\mathcal{K}(\mathcal{M}) = (S, S_0, \rightarrow, L)$ in the following way:

- $S := \mathbb{X}$,
- $S_0 := \{X_0\}$,

- $\rightarrow$ is obtained by totalising the transition relation T — every state $X \in \mathbb{X}$ with no outgoing transition in T is assigned a self-transition $X \rightarrow X$,
- L is defined in such a way that atomic propositions $AP = \{X_i < \theta \mid 0 \leq \theta \leq \mathbf{m}_i\} \cup \{X_i = \theta \mid 0 \leq \theta \leq \mathbf{m}_i\}$ are assigned to corresponding states.

We say $\mathcal{M}$ satisfies φ, written $\mathcal{M} \models \varphi$, iff $\mathcal{K}(\mathcal{M}) \models \varphi$.

Parameterised dINM

The representation of a dINM in terms of a KS can be directly extended to parameterised models. This is done by encoding the semantics of a parameterised model in terms of a PKS. In particular, the explicit representation by means of a set of transition systems is turned into a compact form of a single PKS.

Formally, let $\mathcal{M}_{\mathcal{P}_\chi} = (\mathcal{N}, \nu_{dIN}, \mathbb{X}, X_0, \mathbf{m})$ be a parameterised dINM with a parameterisation $\chi_\mathcal{M}$. The semantics of $\mathcal{M}_{\mathcal{P}_\chi}$, $[\![\mathcal{M}_{\mathcal{P}_\chi}]\!]_{\text{dINM}}$, is turned into a PKS $\mathcal{K}(\mathcal{M}_{\mathcal{P}_\chi}) = (\mathcal{P}_{\chi_\mathcal{M}}, S, S_0, \rightarrow, L)$ where

- S, S_0 and L are assigned in the same way as in the non-parameterised case described above,
- for any $\mu \in \mathcal{P}_{\chi_\mathcal{M}}$, $s \xrightarrow{\mu} s'$ is included in the transition relation of $\mathcal{K}(\mathcal{M}_{\mathcal{P}_\chi})$ iff $(s, s') \in T$ where T is the transition relation of $[\![\mathcal{M}_{\mathcal{P}_\chi}(\mu)]\!]_{\text{dINM}}$.

Note that the set of parameter valuations $\mathcal{P}_{\chi_\mathcal{M}}$ makes directly the parameter values set of $\mathcal{K}(\mathcal{M}_{\mathcal{P}_\chi})$.

Finally, we say that a parameterised model $\mathcal{M}_{\mathcal{P}_\chi}$ satisfies φ wrt a set of parameter valuations $\mathbf{P} \subseteq \mathcal{P}_{\chi_\mathcal{M}}$ iff $\forall \mu \in \mathbf{P}.\mathcal{K}(\mathcal{M}_{\mathcal{P}_\chi})_\mu \models \varphi$. The situation can be also rephrased in the following way: $\mathcal{M}_{\mathcal{P}_\chi}$ satisfies φ *robustly* wrt $\mathbf{P}$.

cBNM

For all kinds of cBNMs, there is natural interpretation of STL and STL*. In particular, we say that a cBNM $\mathcal{M}$ *satisfies an* STL *or* STL* *formula* φ, written $\mathcal{M} \models \varphi$, iff the signal $\mathfrak{x} = [\![\mathcal{M}]\!]_{\text{cBNM}}$ satisfies $\mathfrak{x} \models \varphi$ (we can write directly $[\![\mathcal{M}]\!]_{\text{cBNM}} \models \varphi$). The satisfaction can be understood also approximately, e.g., for signals obtained by numerical simulation: a cBNM $\mathcal{M}$ *approximately satisfies* an STL or STL* formula φ, written $\mathcal{M} \models^\epsilon \varphi$, iff an approximate signal $\mathfrak{x}^\epsilon = [\![\mathcal{M}]\!]^\epsilon_{\text{cBNM}}$ satisfies $\mathfrak{x}^\epsilon \models \varphi$ (we can write directly $[\![\mathcal{M}]\!]^\epsilon_{\text{cBNM}} \models \varphi$).

LTL and CTL cannot be interpreted on cBNMs directly but only on discrete abstractions of cBNMs. The relation can be specific for particular fragments of the logics and the particular form of differential equations underlying the abstracted cBNM. The rectangular abstraction defined in Section 2.4.6 is conservative resulting with the following characterisation.

First, we need to clarify the approximate interpretation of LTL over signals. We say that $\mathcal{M}$ *approximately satisfies* an LTL formula φ, written $[\![\mathcal{M}]\!]_{\mathrm{cBNM}} \models^{\epsilon} \varphi$, iff the approximate signal given by $[\![\mathcal{M}]\!]^{\epsilon}_{\mathrm{cBNM}}$ satisfies φ. Note that we have not defined the semantics of LTL over discrete bounded-time approximations of continuous signals. However, as long as approximate signals are finite timed sequences of real values, an extension of LTL employing constraints over reals (employed at the place of atomic propositions) can be used [162, 163]. An analogous solution has been introduced for CTL with constraints over reals [164]. These approaches allow to "practically" interpret LTL and CTL over such approximate signals. In the characterisation below we assume the relation $\mathfrak{x} \models^{\epsilon} \varphi$ is interpreted in the way "we can construct a sufficiently precise approximation of $\mathfrak{x}$ such that it correctly reflects the semantics of φ and thus it sufficiently characterises the satisfaction of φ".

Finally, we can proceed in characterising satisfaction of LTL and CTL by using approximate/abstract models.

- Let φ be a universally interpreted LTL formula and $\mathcal{M}$ a cBNM. By reflecting Claim 2.32, the clarification of existence of a sufficiently precise approximate signal mentioned above, and the universal interpretation of LTL, the following holds:

$$[\![\mathcal{M}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}} \models \varphi \Rightarrow [\![\mathcal{M}]\!]_{\mathrm{cBNM}} \models^{\epsilon} \varphi$$

- Let φ be an ACTL formula and $\mathcal{M}$ a cBNM with a parameterisation $\chi_{\mathcal{M}}$. By reflecting Claim 2.32 and the clarification of existence of a sufficiently precise approximate signal, the following holds:

$$[\![\mathcal{M}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}} \models \varphi \Rightarrow [\![\mathcal{M}]\!]_{\mathrm{cBNM}} \models^{\epsilon} \varphi$$

Remark 2.43 *If φ is of the ECTL fragment of CTL or it is an existentially interpreted LTL formula, we naturally get the following characterisation:*

$$[\![\mathcal{M}]\!]_{\mathrm{cBNM}} \models^{\epsilon} \varphi \Rightarrow [\![\mathcal{M}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}} \models \varphi$$

Parameterised cBNM

In the case of a parameterised cBNM $\mathcal{M}_{\mathcal{P}_{\chi}}$ the satisfaction of a formula φ can be considered at the level of the abstract/approximated semantics $[\![\mathcal{M}_{\mathcal{P}_{\chi}}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}}$. We declare that $[\![\mathcal{M}_{\mathcal{P}_{\chi}}]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}}$ satisfies φ wrt some set of parameter valuations $\mathbf{P} \subseteq \mathcal{P}_{\chi_{\mathcal{M}}}$, iff $[\![\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)]\!]^{abst^{\epsilon}}_{\mathrm{cBNM}} \models \varphi$ for all $\mu \in \mathbf{P}$. This can be equivalently written as $\mathcal{K}^{abst^{\epsilon}}(\mathcal{M}_{\mathcal{P}_{\chi}})_{\mu} \models \varphi$ for all $\mu \in \mathbf{P}$.

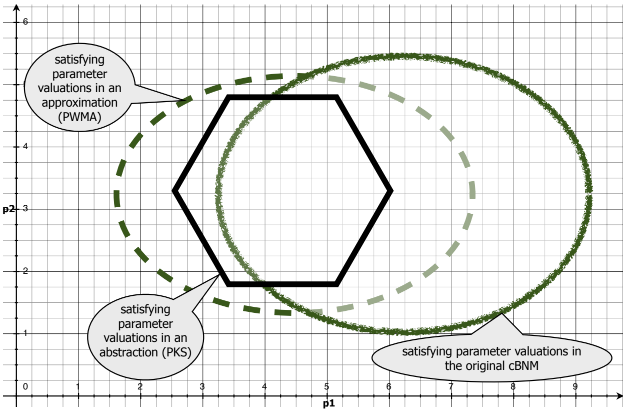

6
satisfying parameter valuations in an approximation (PWMA)
4
p2
3
2
satisfying parameter valuations in an abstraction (PKS)
1
satisfying parameter valuations in the original cBNM
0
1
2
3
4
5
6
7
8
9
p1

Figure 2.8: A two-dimensional set of parameter valuations (theoretically) satisfying a parameterised cBNM $\mathcal{M}_{\mathcal{P}_{\chi}}$ is marked by the solid oval. The set of parameter valuations (theoretically) satisfying φ on the corresponding PWMA model are depicted by the dashed oval. The set of parameter valuations satisfying a formula φ on $[\![\mathcal{M}_{\mathcal{P}_{\chi}}]\!]^{abst^{\epsilon}}$ is depicted by the polygonal region.

Owing to the fact $\mathcal{K}^{abst^{\epsilon}}$ makes an over-approximation of the intermediate PWMA model, every $\mu \in \mathcal{P}_{\chi_{\mathcal{M}}}$ such that $[\![\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)]\!]^{abst^{\epsilon}}_{\text{cBNM}} \models \varphi$ implies that an LTL or ACTL formula φ is (theoretically) satisfied also in the corresponding PWMA model. However, there might be $\mu \in \mathcal{P}_{\chi_{\mathcal{M}}}$ such that $[\![\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)]\!]^{abst^{\epsilon}}_{\text{cBNM}} \not\models \varphi$ but the formula is (theoretically) satisfied in the corresponding PWMA model.

Finally, since $\mathcal{M}$ comes through the approximation procedure, we cannot tell anything exact about satisfaction of φ in the original cBNM $\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)$. Thus we can speak about "approximate" satisfaction as discussed above: if $[\![\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)]\!]^{abst^{\epsilon}}_{\text{cBNM}} \models \varphi$ then $\mathcal{M}_{\mathcal{P}_{\chi}}(\mu) \models^{\epsilon} \varphi$ ($\models^{\epsilon}$ is understood up-to the distance between the signal $[\![\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)]\!]_{\text{cBNM}}$ and its approximated counterpart generated by the PWMA approximation of $\mathcal{M}_{\mathcal{P}_{\chi}}(\mu)$). The situation for a set of parameter valuations $\mathbf{P}$ wrt to an interpretation of an LTL or ACTL formula φ on the original cBNM, its approximate PWMA and the corresponding abstract Kripke structure is depicted schematically in Figure 2.8.

sCRNM

The semantics of a sCRNM model $\mathcal{M}$ is given as a CTMC $(\mathbb{X}, X_0, \mathbf{Q})$. To enable interpretation of temporal logic over CTMC, the structure is extended with a labelling function $L : \mathbb{X} \to 2^{AP}$ that assigns atomic propositions to states. Since the matrix $\mathbf{Q}$ represents a (transition) relation among states, the CTMC extended with L induces a Kripke structure (the transition relation has to be totalised in a similar way as mentioned above). Qualitative interpretation of LTL and CTL neglecting the stochasticity can be then realised in the same way as in the case of dINMs.

The logic naturally interpreted on sCRNMs is CSL. We say $\mathcal{M}$ satisfies a formula φ of the form $\mathsf{P}_{\sim p}[\phi]$, $\mathsf{R}_{\sim r}[\mathsf{C}^{\leq t}]$, or $\mathsf{R}_{\sim r}[\mathsf{I}^{=t}]$, written $\mathcal{M} \models \varphi$, iff $X_0 \in Sat_{[\![\mathcal{M}]\!]_{\mathrm{sCRNM}}}(\varphi)$.

Parameterised sCRNM

For a parameterised sCRNM $\mathcal{M}_{\mathcal{P}_\chi}$ the semantics $[\![\mathcal{M}_{\mathcal{P}_\chi}]\!]_{\mathrm{sCRNM}}$ is given by the (continuous) set of individual CTMCs representing the semantics of $\mathcal{M}_{\mathcal{P}_\chi}(\mu)$ for particular $\mu \in \mathcal{P}_{\chi_\mathcal{M}}$.

We say a CSL formula φ satisfies $\mathcal{M}_{\mathcal{P}_\chi}$ wrt a given set $\mathbf{P} \subseteq \mathcal{P}_{\chi_\mathcal{M}}$ iff for all $\mu \in \mathbf{P}$ it holds that $\mathcal{M}_{\mathcal{P}_\chi}(\mu) \models \varphi$. This is a theoretical notion that cannot be in general case decided exactly. In Chapter 5 we provide an approximate solution.

2.6 Model Analysis

In this section we describe a group of formal methods that makes a basis for advanced techniques employed for analysis of biological models. We start with elemental methods of general formal verification and finally describe methods relevant for biological models.

2.6.1 Restriction to Bounded Time

From the context of an observed phenomenon and the time-scale of relevant model behaviour, it can be possible to identify the time-horizon (or the number of steps in the untimed case) for which it is guaranteed that the phenomenon occurs. Even periodically repeating phenomena, e.g., circadian clock, can be approximately detected and analysed in finite time in the order of an appropriately selected time-scale. However, a non-trivial analysis has to be performed in some cases to estimate or over-approximate correctly the time horizon.

In the case of dINMs, the model dynamics is fully untimed. We consider the asynchronous semantics of concurrent updates that is conservative – no information on possible timing (ordering) of events is lost. Cycles in the dynamics correspond to possible equilibria (or attractors). However, it cannot be decided from the model what is the time horizon or periodicity of the real dynamics corresponding to a cycle.

In the case of continuous models, simulation-based method need to deal with correct reflecting of the time horizon. The techniques based on qualitative abstraction abstract the time-aspects conservatively in the lines similar to dINMs.

Stochastic models are based on CTMCs where the ergodicity allows to analyse the behaviour in infinite time horizon (steady-state analysis can be used to explore the long-time behaviour in ergodic models). However, when the transient analysis is performed, the considered time-horizon might be critical and needs to be treated with special care.

2.6.2 Reachability Analysis

One of the most basic problems of analysis of dynamical systems is the reachability analysis, a general problem well-studied in graph theory. The problem is to decide whether a given state (or a given set of states) can be reached in finite time from an initial state. The problem has been addressed in formal verification of concurrent systems. In the case of discrete models, many efficient algorithms that does not require explicit generation of the state space exist taking the advantage of a particular encoding of a specific class of models (see for example [69, 171, 356, 112] for untimed systems or [263, 125, 370] for timed systems).

dINM

In the case of dINMs, many of the general algorithms can be employed that perform either on the explicit or implicit representation of the respective Kripke structure. Especially, techniques employing symbolic representation of the state-transition relation in terms of BDDs have proved to be efficient in this case [62, 106]. However, reachability analysis of parameterised dINMs is still challenging. Several techniques exploring static analysis to characterise parameter valuations solving a given reachability problem have been recently addressed in [248].

cBNM

In the case of cBNMs the reachability analysis is undecidable and must be targeted approximately by employing symbolic δ-decision procedures [187, 252], numeric over-approximation methods [156], or by using finite abstractions (or hybridisations) of continuous dynamics. The global abstraction methods as discussed in Section 2.4.6 can be used for reachability analysis provided that the results have to be carefully interpreted with respect to the fact the abstraction is typically an over-approximation (in such case the property of not-reaching a certain state is guaranteed through the abstraction). Approaches that directly target reachability construct local abstractions of the flow of the dynamics by employing so-called *flow-pipes* [25, 109, 251] that over-approximate the signals or use various sophisticated forms of geometric or symbolic representation of system states sets [6, 178, 313]. In some work, zonotopal representations are employed [5, 198]. Local methods address the flow of the dynamics quite well. Several methods target unbounded-time analysis in some form [25, 119]. The work presented in [110] combines local and global approaches by targeting abstractions of linear thresholds-driven hybrid systems. The authors use local flow-pipe construction to construct a quotient transition system that models the reachability from one cell to another. The tool ProbReach [331] (based on δ-decision procedures) addresses statistical sampling of initial conditions and is suitable for the global analysis where the initial condition is uncertain.

sCRNM

sCRNMs are modelled by means of CTMCs. The transitions between the states in CTMC are governed by the transition rate matrix. It assigns a rate r to each pair of states in the CTMC, which are used as parameters of the exponential distribution, i.e., the probability of the transition being triggered within t time-units equals $1 - e^{-r \cdot t}$. The reachability analysis problem is solved by exact techniques based on so-called transient analysis of the Markov process [23, 260] or by approximate techniques based on statistical sampling [232, 27] of simulations [195], employing moment-closures [73, 22, 13], or by methods employing correct deterministic approximation of the transient distribution [71, 334].

Transient analysis is based on the computation of transient probability – having started in state X, the probability of being in state X' at time instant t. Formally, given an initial distribution $\pi^{\mathcal{C},X_0,0}$ (i.e. $\pi^{\mathcal{C},X_0,0}(X) = 1$ if $X_0 = X$, and 0, otherwise) at time 0 of a CTMC $\mathcal{C} = (\mathbb{X}, X_0, \mathbf{Q})$ what will the transient state distribution $\pi^{\mathcal{C},X_0,t}$ look like in some future yet finite time $t \in \mathbb{R}_{\geq 0}$.

A standard technique for computing transient probabilities is based on uniformization. The key idea is for a given CTMC to construct the uniformised DTMC where all exponential delays in the CTMC are normalised with respect to the fastest transition rate q. Then each step of the uniformised DTMC corresponds to a single exponentially distributed delay with the parameter q. The ith matrix power of the uniformised DTMC gives the probability of jumping between each pair of states in the DTMC in i steps. The transient probability in time t is computed as the sum of the matrix powers weighted by Poisson probabilities giving the probability of i such steps occurring in time t.

Formally, for the rate q satisfying $q \geq max\{E(X) \mid X \in \mathbb{X}\}$ ($E(X)$ is the exit rate of state X as described in Section 2.3.2) the uniformised DTMC

$\mathsf{unif}(\mathcal{C})$ is defined as $\mathsf{unif}(\mathcal{C}) = \big(\mathbb{X}, X_0, \mathbf{Q}^{\mathsf{unif}(\mathcal{C})}\big)$ where

$$\mathbf{Q}^{\mathsf{unif}(\mathcal{C})}(X, X') = \begin{cases} \frac{\mathbf{Q}(X,X')}{q} & \text{if } X \neq X' \\ 1 - \sum_{X'' \neq X} \frac{\mathbf{Q}(X,X'')}{q} & \text{otherwise.} \end{cases}$$

and the ith Poisson probability in time t is given as $\gamma_{i,q\cdot t} = e^{-q\cdot t} \cdot \frac{(q\cdot t)^i}{i!}$. The transient probability can be computed as follows:

$$\pi^{\mathcal{C},X_0,t} = \sum_{i=0}^{\infty} \gamma_{i,q\cdot t} \cdot \pi^{\mathcal{C},X_0,0} \cdot (\mathbf{Q}^{\mathsf{unif}(\mathcal{C})})^i \approx \sum_{i=lb_\epsilon}^{ub_\epsilon} \gamma_{i,q\cdot t} \cdot \pi^{\mathcal{C},X_0,0} \cdot (\mathbf{Q}^{\mathsf{unif}(\mathcal{C})})^i.$$

Although the sum is in general infinite, for a given precision ϵ the lower and upper bounds lb_ϵ, ub_ϵ can be estimated by using techniques such as of Fox and Glynn [176] which also allow for efficient solutions of the Poisson process. In order to make the computation of uniformization feasible the matrix-matrix multiplication is reduced to a vector-matrix multiplication as follows:

$$\pi^{\mathcal{C},X_0,0} \cdot (\mathbf{Q}^{\mathsf{unif}(\mathcal{C})})^i = (\pi^{\mathcal{C},X_0,0} \cdot (\mathbf{Q}^{\mathsf{unif}(\mathcal{C})})^{i-1}) \cdot \mathbf{Q}^{\mathsf{unif}(\mathcal{C})}.$$

Standard uniformisation can be intractable when the system under study is too complex, i.e., contains more than in order of 10^7 states and the upper estimate ub_ϵ, denoting the number of vector-matrix multiplications as iterations, is high (more than in order of 10^6). Therefore, many approximation techniques have been studied in order to reduce the state space and to lower the number of iterations ub_ϵ. State space reductions are based on the observation that in many cases (especially in biochemical systems) a significant amount of the probability mass in a given time is localised in a manageable set of states. Thus neglecting states with insignificant probability can dramatically reduce the state space while the resulting approximation of the transient probability is still sufficient. Methods allowing efficiently state-space reduction are based on finite projection techniques [292, 220] and dynamic state space truncation [142].

Since the number of iterations ub_ϵ inherently depends on the uniformisation rate q that has to be greater than the maximal exit rate of all the states of the system, a variant of standard uniformisation, so-called *adaptive* uniformisation [357], has been proposed. It uses a uniformisation rate that adapts depending on the set of states the system can occupy at a given time, i.e, after a particular number of reactions. In many cases, a significantly smaller rate q can be used and thus the number of iterations ub_ϵ can be significantly reduced during some parts of the computation. Moreover, adaptive uniformisation can be successfully combined with reduction techniques mentioned above [142]. The downside of adaptive uniformisation is that the Poisson process has to be replaced with a general *birth* process which is more expensive to solve. See, e.g [357], for more details.

2.6.3 Model Checking

Model checking is an exhaustive technique of checking whether all signals/paths generated by a given model $\mathcal{M}$, satisfy the inspected property described as the formula φ. In order to generate all signals or paths, the whole state-space has to be stored and evaluated. Owing to this need, model checking generally suffers from the *state-space explosion problem*. There exist several approaches to reduce this problem, e.g., efficient symbolic representation, state-space reductions or iterative abstraction refinement. Model checking algorithms differ depending on the temporal logic employed. An extensive review of model checking methods is given in [24]. In this section we restrict ourselves only methods relevant for the thesis.

LTL Model Checking

There are several approaches to model checking a finite state Kripke structure $\mathcal{K}$ against an LTL formula φ [24]. The goal is to check $\mathcal{K} \models \varphi$, in particular, whether for every $s_0 \in S_0$ every path π in $\mathcal{K}$ starting at s_0 satisfies π. In this thesis we use the automata-based approach based on the following ideas:

1. construct a Büchi automaton over 2^{AP} such that it accepts all words satisfying $\neg\varphi$,
2. convert $\mathcal{K}$ to a Büchi automaton which accepts words of the form $L(s_0)L(s_1)\cdots$ for each run $s_0 s_1 \cdots$ in $\mathcal{K}$,
3. compute the synchronous product of the two automata, and
4. decide whether the resulting automaton accepts an empty language.

Definition 2.44 *A (non-deterministic)* Büchi automaton *is a tuple $\mathcal{B} = (\Sigma, S, s_0, \delta, F)$, where Σ is the input alphabet, S is a finite non-empty set of states, $s_0 \in S$ is a distinguished initial state, $\delta \subseteq S \times \Sigma \times S$ is a transition relation, and $F \subseteq S$ is the set of accepting states.*

The automaton $\mathcal{B}$ accepts the infinite word $w \in \Sigma^\omega$, if there is a sequence of states $s_0 s_1 \ldots$ such that $(s_i, w_i, s_{i+1}) \in \delta$ and there is a state $f \in F$, which appears in the sequence infinitely often.

There are well-known sequential and parallel algorithms performing automata-based model checking of LTL (e.g., [358, 32, 81, 80]). These algorithms are implemented in several tools [222, 36, 34, 37]. In our work [35, 37], we have focused on transferring the LTL model checking to biological systems (cBNMs) by employing discrete abstractions.

CTL Model Checking

There are several approaches to model checking a finite state Kripke structure $\mathcal{K}$ against a CTL formula φ. The goal is to check $\mathcal{K} \models \varphi$, in particular, whether for every $s_0 \in S_0$, $s_0 \models_{\mathcal{K}} \varphi$. In this thesis we use the enumerative fixed point approach [117, 311]. The basic procedure for CTL model checking is the following:

- the set of all states satisfying φ, denoted $Sat(\varphi)$, is computed recursively, and
- $\mathcal{K} \models \varphi$ if and only if $S_0 \subseteq Sat(\varphi)$.

In fact, the procedure of CTL model checking checks $s \models_{\mathcal{K}} \varphi$ for every state $s \in S$. Therefore it provides a *global* model checking technique.

The recursive computation of $Sat(\varphi)$ is performed by a bottom-up traversal of the parse tree of the formula φ. The individual nodes in the parse tree represent subformulae of φ. At an intermediate node of the parse tree, representing a subformula ψ, the results of the computations of its children are used and combined to establish the states of its associated subformula. The type of computation at a particular node (the so-called *node algorithm*) depends on the logical operator at the top level of the subformula ψ. The children of the node stand for the maximal genuine subformulae of the formula ψ. In case ψ has the form of **EU** or **EG** (respectively **AU** or **AG**) operator, the satisfaction set $Sat(\psi)$ is obtained iteratively by computing a fixed point (backward traversal of the transition relation is employed in case of **EU** and **EG**, for details see [24], page 349).

Termination of the entire model checking procedure is ensured due to the fact the Kripke structure is required to be finite. For large-scale Kripke structures, a distributed variant of the fixed point algorithm has been introduced in [91].

Examples of tools employing the above mentioned approach are EMC [117], CESAR [311], and CADP [188].

CSL Model Checking

The aim of the global CSL model checking technique is to efficiently compute for any CSL formula φ and any state $X \in \mathbb{X}$ of the given labelled CTMC $\mathcal{C} = (\mathbb{X}, X_0, \mathbf{Q}, L)$ the probability of satisfying $s \models \varphi$. To formally capture the satisfaction probability for every state, an auxiliary function $Eval(\mathcal{C}, \varphi)$ is defined in the following way:

$$Eval(\mathcal{C}, \varphi) = \begin{cases} Prob^{\mathcal{C}}(X_0, \phi) & \text{if } \varphi \equiv \mathsf{P}_{\star}[\phi] \\ Exp^{\mathcal{C}}(X_0, \mathbf{X}_{\mathsf{C}^{\leq t}}) & \text{if } \varphi \equiv \mathsf{R}_{\star}[\mathsf{C}^{\leq t}] \\ Exp^{\mathcal{C}}(X_0, \mathbf{X}_{\mathsf{I}^{=t}}) & \text{if } \varphi \equiv \mathsf{R}_{\star}[\mathsf{I}^{=t}] \end{cases} \tag{2.2}$$

where $\star \in \{=?, \sim r\}$.

The goal of local model checking technique is to compute $Eval_{\varphi}^{\mathcal{C}}(X)$ for a single state $X \in \mathbb{X}$. The crucial advantage of the global approach is the fact that it has the same asymptotic and also practical complexity as the local approach. Therefore, the global model checking technique is much more suitable for robustness analysis over perturbations of initial conditions that are encoded as the initial state of the corresponding CTMC.

Global model checking returns the vector of size $|\mathbb{X}|$ such that the *i*th position contains the model checking result provided that X_i is the initial state. Now let $\mathcal{C} = (\mathbb{X}, \mathbf{Q}, L)$ be a labelled CTMC where the initial state is not specified. The crucial part of this method is to compute the vector of probabilities $\overline{Prob}^{\mathcal{C},\phi}$ for any path formula ϕ and the vector of expected rewards $\overline{Exp}^{\mathcal{C},\mathbf{X}}$ for $\mathbf{X} \in \{\mathbf{X}_{\mathsf{I}=t}, \mathbf{X}_{\mathsf{C}\leq t}\}$ such that for all $X \in \mathbb{X}$ the following holds:

$$\overline{Prob}^{\mathcal{C},\phi}(X) = Prob^{\mathcal{C}}(X,\phi) \wedge \overline{Exp}^{\mathcal{C},\mathbf{X}}(X) = Exp^{\mathcal{C}}(X,\mathbf{X})$$

In local model checking the computation of $Prob^{\mathcal{C}}(X,\phi)$ and $Exp^{\mathcal{C}}(X,\mathbf{X})$ is reduced to the computation of the transient probability distribution $\pi^{\mathcal{C},X,t}$, see [23, 260] for more details. Thus, for different initial states $X \in \mathbb{X}$ we have to compute the corresponding transient probability distributions separately. The key idea of the global model checking method is to use *backward transient analysis*. The result of backward transient analysis is the vector $\rotatebox{180}{$\pi$}^{\mathcal{C},\mathbb{A},t}$ such that for arbitrary set of states $\mathbb{A} \subseteq \mathbb{X}$, the value $\rotatebox{180}{$\pi$}^{\mathcal{C},\mathbb{A},t}(X)$ is the probability that $\mathbb{A}$ is reached from X at the time t. Without going into details the vector $\rotatebox{180}{$\pi$}^{\mathcal{C},\mathbb{A},t}$ can be computed in a very similar way using the uniformised DTMC $\mathsf{unif}(\mathcal{C})$ as in the case of vector $\pi^{\mathcal{C},X,t}$. Only vector-matrix multiplication is replaced by matrix-transposed-vector multiplication and $\rotatebox{180}{$\pi$}^{\mathcal{C},\mathbb{A},0}(X) = 1$ if $X \in \mathbb{A}$, and 0, otherwise.

The vector $\overline{Prob}^{\mathcal{C},\phi}$ is computed using backward transient analysis. Since the definition of next operator $\mathbf{X}\ \varphi$ does not rely on any real time aspects of CTMCs, its evaluation stems from the probability of the next event that can be easily obtained from the transition matrix $\mathbf{Q}$. The evaluation of the until operator $\varphi_1 \mathsf{U}^I \varphi_2$ depends on the form of the interval I and is separately solved for the cases of $I = [0, t_1]$ and $I = [t_1, t_2]$ where $t_1, t_2 \in \mathbb{R}_{\geq 0}$. It is based on a modification of the uniformised infinitesimal generator matrix $\mathbf{Q}^{\mathsf{unif}}$ where certain states are made absorbing. This means that all outgoing transitions are ignored in dependence on the validity of φ_1 and φ_2 in these states.

For any CSL formula φ, let $\mathcal{C}[\varphi] = (\mathbb{X}, X_0, \mathbf{Q}[\varphi], L)$, where $\mathbf{Q}[\varphi](X, X') = \mathbf{Q}(X, X')$, if $X \models \varphi$, and 0, otherwise. The formula $\phi = \varphi_1\ \mathsf{U}^{[0,t]}\ \varphi_2$ can be evaluated using the vector $\rotatebox{180}{$\pi$}^{\mathcal{C},\mathbb{A},t}$ in the following way:

$$\overline{Prob}^{\mathcal{C},\phi} = \rotatebox{180}{π}^{\mathcal{C}[\neg\varphi_1\wedge\varphi_2],\mathbb{A},t} \text{ where } X \in \mathbb{A} \text{ iff } X \models \varphi_2.$$

For the formula $\phi = \varphi_1 \, \mathsf{U}^{[t_1,t_2]} \, \varphi_2$ the evaluation is split into two parts: staying in states satisfying φ_1 until time t_1 and reaching a state satisfying φ_2, while remaining in states satisfying φ_1, within time $t_2 - t_1$. The formula ϕ can be evaluated using the vector $\amalg^{\mathcal{C},\overline{v},t}$ that takes a vector $\overline{v}$ instead of a set $\mathbb{A}$ (i.e., $\amalg^{\mathcal{C},\overline{v},0} = \overline{v}$) in the following way:

$$\overline{Prob}^{\mathcal{C},\phi} = \amalg^{\mathcal{C}[\neg\varphi_1],\overline{v},t_1} \text{ where } \overline{v} = \amalg^{\mathcal{C}[\neg\varphi_1 \wedge \varphi_2],\mathbb{A},t_2-t_1} \text{ and } X \in \mathbb{A} \text{ iff } X \models \varphi_2.$$

The backward transient analysis can be also used in the case of reward computation. Since operator $\mathsf{R}_{\sim p}[\mathsf{I}^{=t}]$ expresses the expected reward at time t, the vector $\overline{Exp}^{\mathcal{C},\mathbf{X}_{\mathsf{I}=t}}$ can be computed as $\overline{Exp}^{\mathcal{C},\mathbf{X}_{\mathsf{I}=t}} = \amalg^{\mathcal{C},\overline{v},t}$ where $\overline{v}$ encodes the given state reward structure.

For evaluation of the operator $\mathsf{R}_{\sim p}[\mathsf{C}^{\leq t}]$ we have to use *mixed Poisson probabilities* (see, e.g., [259, 260]) in the backward transient analysis. It means that during the uniformisation the Poisson probabilities $\gamma_{i,q\cdot t}$ are replaced by the mixed Poisson probabilities $\bar{\gamma}_{i,q\cdot t}$ that can be computed as:

$$\bar{\gamma}_{i,q\cdot t} = \frac{1}{q} \cdot \left(1 - \sum_{j=1}^{i} \gamma_{j,q\cdot t}\right).$$

Using the given state reward structure we can compute the vector $\overline{Exp}^{\mathcal{C},\mathbf{X}_{\mathsf{C}\leq t}}$ as $\overline{Exp}^{\mathcal{C},\mathbf{X}_{\mathsf{C}\leq t}} = \amalg^{\mathcal{C},\overline{v},t}$ where $\overline{v}$ encodes the state reward structure and the mixed Poisson probabilities $\bar{\gamma}_{i,q\cdot t}$ are used.

To review the overall method of stochastic model checking of CTMCs over CSL formulae we summarise the procedures from an abstract perspective. The evaluation of a structured formula φ proceeds by bottom-up evaluation of a set of atomic propositions, probabilistic or expected reward inequalities and their Boolean combinations. This evaluation gives us a discrete set of states that are further used in the following computation. The process continues up the formula until the root is reached. The final verdict is reported either in the form of a Boolean yes/no answer or as the actual numerical value of the probability or the expected reward.

The robust implementation of the technique is available in the widely used tool PRISM [261] and recently also in the tool STORM [137].

2.6.4 Software for Model Checking of Biological Models

The main tools that allow to apply model checking methods to biological models interface the encoding specific for biological models to an internal language of some basic model checking tool.

cBNM

One of the first tools that has supported model checking of cCRNMs is BIOCHAM [93], it is a general framework for formal analysis of biological networks and it has been continuously developed until now [166]. It has introduced verification of CTL properties on qualitative models of cCRNMs employing a unique asynchronous Boolean semantics working well with the standard symbolic model checker NuSMV [111]. A feature that is original to BIOCHAM is model checking of both LTL and CTL over finite discrete approximations of signals generated by cBNMs [162, 163, 164].

In [290], the authors identify CTL-based patterns of formulae relevant for cINMs. The tool GNA [135] adopts NuSMV and CADP model checkers [50, 52] for piece-wise affine abstractions of cINMs [162, 163, 164]. In [49] the authors introduce an approach employing the abstraction method described in 2.4.6. It is used with LTL model checking to address liveness properties of cINMs, the method has been incorporated in a Matlab-based tool RoVerGeNe [54]. In our work [57], the technique has been transferred to general biological systems (cBNMs) [82]. Although in both cases there is a more complex technical framework working with parameterised models, the techniques can be directly used for models with fixed parameters.

dINM

Considering dINMs, CTL model checking has been widely employed targeting analysis of gene regulatory networks [62, 282, 352] and signalling networks [325, 158, 203]. The tool GINsim [106] is continuously developed to address modelling and analysis of dINMs, it passes the models to the standard symbolic model checker NuSMV [111]. The tool Antelope [17] goes beyond CTL by introducing hybrid operators. The dedicated model checker is capable of verifying properties such as presence of an attractor in the system without the previous knowledge on its location. Extensions of CTL for biological models have been also targeted in [280].

LTL model checking of dINMs has been addressed in dINMs [341, 184, 342] with a prototype tool support. The tool TREMPPI [340] provides a comprehensive collection of methods based in LTL model checking of dINMs. An interface PyBoolNet targeting analysis and visualisation of dINMs has been presented in [243]. A framework aimed at addressing biologists as direct users of dINMs is provided within the BMA tool [59] including a module for LTL model checking [2]. The recent valuable work has been invested in development of a Python-based interface utilising Jupyter Notebooks for interoperability of tools working with dINMs [294]. An overview of some tools dedicated to model checking of dINMs is given in [289].

sCRNM

The general tools existing for model checking CSL properties over CTMCs can be directly used for analysis of sCRNMs. Several versions of the tool

PRISM [261] have been successfully applied to analysis of sCRNMs, e.g., [258, 253, 255, 268]. The main advantage of the tool is it incorporates a large set of different techniques ranging from exact to approximate techniques and using symbolic and explicit representations or their combinations. The tool is also integrated within higher level frameworks. In [113] the PRISM is used as a model checker for models specified in BioPEPA process algebraic modelling framework for biological models. In [320] a web service for biological models analysis integrating several tools including PRISM is presented. Another probabilistic CSL model checker addressing biological models is MARCIE [328] based on its own symbolic techniques. In future we can also expect the usage and adaptation of the STORM [137] model checker that also supports CSL model checking.

Probabilistic model checking has also been employed to the analysis of stochastic models of influence networks where an automatised translation of models into a CTMC, based on quasi-steady-state approximation (QSSA), has been proposed [274, 363]. The related tool iBioSim gives a powerful technology for analysis of stochastic models of genetic networks.

As an alternative to exact probabilistic model checking which can be computationally inefficient for large models, statistical approaches can be employed for properties making a fragment of CSL. The recently developed tool U-Check [72] implements smooth model checking based on Gaussian noise approximation and is very promising for statistical analysis of cCRNMs. The tools developed in the past are [329, 270].

In [92], the authors consider signal transduction in the RKIP-inhibited ERK pathway. They overcome the state-space explosion problem of probabilistic model checking by rescaling model component quantities to lower numbers of population levels. The main problem with statistical model checking is caused by rare events, i.e., temporal formulae whose satisfaction probability is very small. When estimating the probability of such formulae, the number of simulations needed to ensure a good estimate becomes unfeasible. In [116], the authors show that the importance sampling, a variance-reduction technique for the Monte Carlo method, and the cross-entropy method, a general Monte Carlo approach to combinatorial and continuous multi-extremal optimisation and importance sampling, can efficiently address this problem. They use Bounded Linear Temporal Logic, a variant of LTL where the temporal operators are equipped with time bounds, to reason about biochemical reactions in systems biology.

2.6.5 Simulation and Monitoring

Monitoring techniques are based on the satisfaction test of a formula over an individual simulation trace (signal). The responsibility for exhaustive coverage is delegated to the procedure that generates the traces. The key observation behind efficiency of monitoring performed over simulations is that for large and complex systems, the simulation is generally easier and faster than building a concise representation of global transition systems required for the exhaustive model checking approach. However, since a single simulation generates a single trajectory out of all the possible executions of a system, usually the average values among several simulations need to be considered to achieve the necessary level of confidence in the results obtained. Owing to the fact that the individual monitoring problem targets a single path, linear temporal logics are employed (they are typically extended with quantitative operators evaluating the time-aspects of the trace such as MTL, MITL or STL as discussed in Section 2.5).

A possible way to improve the accuracy of monitoring techniques is to employ the statistical model checking [116] that addresses general stochastic systems in terms of statistical inference. It samples the behaviours (simulations) of a model, verifies their conformance with respect to a temporal formula (i.e. performs the membership test), and finally applies a statistical estimation technique to compute an approximate value for the probability that the formula is satisfied. The accuracy of statistical model checking is affected by the accuracy of stochastic simulations techniques that are employed and also by the structure of the model or more precisely by the level of details (initial conditions, parameters, etc.) we have about the system under study.

A comprehensive survey of monitoring techniques is given in [44]. The techniques are divided to two main groups: offline monitoring (working on a generated trace) or online monitoring (working in real time). In the basic setting, the method computes Boolean satisfaction of a formula (e.g., computing the qualitative semantics of STL or STL* over a signal as stated in Sections 2.5.3 and 2.5.4). Additionally, the quantitative semantics (e.g., as of Definition 2.37) can be evaluated. In such a case, we speak about *robust monitoring*.

On the practical side, there exist several tools targeting monitoring, one of the first tools were Temporal Rover [154] (qualitative satisfaction of LTL/MTL) and JavaPath Explorer [213] (qualitative satisfaction of temporal properties over Java byte-code executions). The tool AMT [298, 297] provides algorithms for the qualitative satisfaction of STL. Several tools have been developed regarding robust monitoring with quantitative semantics: Breach [145] (Matlab-based, piece-wise constant signals wrt STL), S-TaLiRo [14] (Matlab-based, finite timed sequences wrt MTL), and U-check [72] (standalone, monitors simulations of CTMCs wrt STL). Our own contribution implemented in the tool Parasim (standalone) is described in Section 6.2. The tool BIOCHAM [166] mentioned in Section 2.6.3 also provides offline monitoring algorithms for finite timed sequences wrt LTL and even CTL. All of these tools with the exception of U-check are directly applicable to cBNMs as they work with the differential semantics of models. U-check is directly applicable to sCRNMs. There exist several interesting applications to biological problems conducted with Breach and BIOCHAM [148, 351, 279].

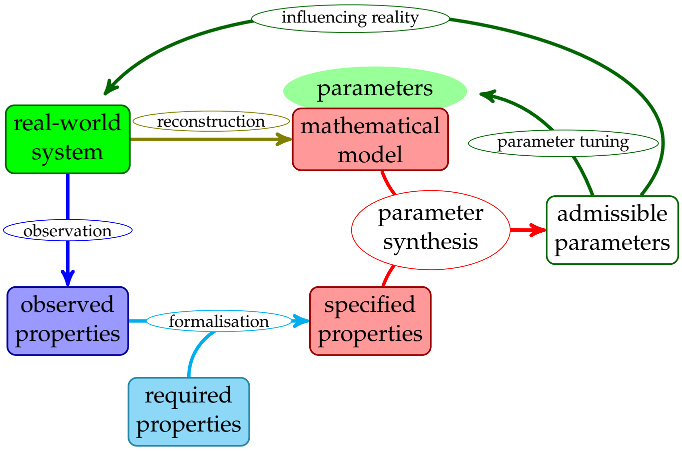


Figure 2.9: A general scheme of parameter synthesis methods based on system properties formalised in a temporal logic.

It is worth noting that the monitoring procedure can be also applied to experimentally measured time-series data. Monitoring then enables methods of automatised inference of a logical specification of real-world systems dynamics from experimental data [295].

2.6.6 Parameter Synthesis

The problem of parameter synthesis is for a given parameterised model to identify parameter valuations that guarantee the system satisfies the given global property formalised in some temporal logic. The technique provides an alternative way to more traditional method of fitting model parameters to experimental data (also called parametric systems identification [314] or parameter estimation [288]). In contrast to the traditional approaches to tackle the inverse problem (e.g., [179, 183, 193, 314, 288]) parameter synthesis methods typically focus on identifying reliable subsets of parameter space instead of finding singular parameter values.

The overall settings of property-driven parameter synthesis are depicted in Figure 2.9. Hypotheses mined from biological literature as well as time-series experiments from wet-labs can be considered as temporal properties restricting the admissible set of parameter valuations.

dINM

In the case of discrete models, usage of model checking for guiding synthesis of admissible parameters has been originally targeted for CTL in [62] supported with the tool SMBioNet [237] (employing NuSMV). The approach uses the naïve algorithm of separate model checking tasks run for every parameter valuation of the model. Decisions over parameter space of models with synchronous semantics can be done very efficiently using SAT, SMT [192] or answer-set programming [175, 300]. The problem becomes more challenging for semantics with asynchronous updates as is considered in our case for dINMs. Parameters can be synthesised iteratively by using a model checker with an appropriate symbolic encoding of parameters as is done in [51] by employing NuSMV.

Parameter synthesis for LTL has been first addressed in our work presented in Section 4.2 where also a detailed discussion of related work is given.

cBNM

Several techniques have been developed for parameter synthesis of continuous-time models. In the case of linear-time TL, the dominating approach is based on monitoring simulated trajectories of the system [37, 86, 148, 312, 319]. These techniques rely on numerical solvers which are well-developed for systems with fixed parameters or small parameter spaces (perturbations). An advantage of these techniques is that they consider the function defining the systems dynamics as a black box provided that there is basically no limitation on the form of parameterisation of the system. The algorithm implemented in Breach [145] utilises robust monitoring of STL. It is based on a hybrid approach built on sensitivity analysis [126]. The parameter space is traversed with respect to sensitivity of the quantitative semantics of STL wrt changes in parameters. The tool BIOCHAM [166] supports exploration of parameters with respect to LTL formulae with constraints over reals. It brings the methods known from traditional parameter estimation to the settings based on temporal logics. Parameter space is searched with a covariance matrix evolutionary strategy that attempts to identify a gradient in quantitative formula satisfaction wrt changes in parameters.

The main drawback of the simulation-based methods is the need to sample the parameter space and initial states while losing robust guarantees for the results. This drawback can be overcome by replacing numerical solvers with Satisfiability Modulo Theories (SMT) solvers that can cope with non-linear functions and real domains up to required precision [187]. However, these techniques are limited to reachability analysis (e.g., the tool BioPsy [275]) and their extension to work with general TL specifications is a non-trivial task yet to be explored.

Techniques based on finite abstractions of cBNMs employ an exact (symbolic) representation of the maximal set of parameterisations satisfying a given TL property. These techniques are based on model checking performed directly on a qualitative finite quotient or hybridisation of systems dynamics (e.g., [49, 51, 67] and our approaches [33, 82] described in Chapter 4). Several approaches encode parameter sets symbolically in terms of polytopes [49, 205]. Another solution is to encode parameterisation sets by means of predicate formulae with non-linear arithmetics over real numbers and use SMT to reason on them.

sCRNM

In the case of sCRNMs, parameter synthesis methods are based on probabilistic model checking or variants of statistical model checking. Most statistical approaches to parameter synthesis [318, 12, 124] rely on approximating the maximum likelihood wrt to a given set of data. The advantage is the possibility to analyse infinite state spaces [12] (employing dynamic state space truncation with numerically computed likelihood) or even models with no prior knowledge of parameter ranges [124] (using Monte-Carlo optimisation for computing the likelihood). The state-of-the-art moment closure approaches are able to cope with multi-modal distributions occurring in multi-stable systems [12, 273, 22]. The technique implemented in the tool U-check uses so-called smoothed model checking [73] that brings an improved Bayesian statistical algorithm which performs model checking over paths simultaneously for all values of the model parameters from observations of truth values of the formula over individual paths of the model obtained for isolated parameter valuations.

Approaches based on Markov Chain Monte-Carlo sampling and Bayesian inference [200, 232, 247] can be extended to sample-based approximation of parameter values, but at the price of undesired inaccuracy and high computational demands [61, 27].

2.6.7 Robustness Degree

The concept of robustness addresses aspect of the functional evaluation of a dynamical system by considering a weighted average of every behaviour across a space of perturbations affecting the model parameters (hence its behaviour) and in a particular way, having a certain probability of occurrence. The general definition of robustness [242] gives us *robustness degree* that quantitatively characterises to what extent is the evaluated system functionality preserved under considered perturbations:

$$\mathrm{rd}^{\mathcal{M}}_{\mathcal{A},\mathbf{P}} \overset{def}{=} \int_{\mathbf{P}} \psi(\mu) D^{\mathcal{M}}_{\mathcal{A}}(\mu) d\mu$$

where $\mathcal{M}$ is the system (represented by an appropriate model), $\mathcal{A}$ is the function under scrutiny, $\mathbf{P}$ is the set of perturbations, $\mathbf{P} \subseteq \mathcal{P}_{\chi_{\mathcal{M}}}$, $\psi(\mu)$ is the probability of the perturbation $\mu \in \mathbf{P}$ and $D^{\mathcal{M}}_{\mathcal{A}}(\mu)$ is an *evaluation function* stating how much the function $\mathcal{A}$ is preserved under a perturbation μ.

Kitano proposed that the evaluation function $D^{\mathcal{M}}_{\mathcal{A}}(\mu)$, stating how much the functionality $\mathcal{A}$ is preserved in perturbation μ, should be defined using a subspace $\mathbf{B}$ of all perturbations, where the system's function is completely missing and the remaining $\mathbf{P} \setminus \mathbf{B}$ where the function's viability is somehow altered. Formally, the definition is the following:

$$D^{\mathcal{M}}_{\mathcal{A}}(\mu) = \begin{cases} 0 & \mu \in \mathbf{B} \subset \mathbf{P} \\ f_{\mathcal{A}}(\mu)/f_{\mathcal{A}}(0) & \mu \in \mathbf{P} \setminus \mathbf{B} \end{cases}$$

For perturbations $\mu \in \mathbf{P} \setminus \mathbf{B}$ where the system maintains its function at least partially, Kitano proposes to express the evaluation function $D^{\mathcal{M}}_{\mathcal{A}}(\mu) = f_{\mathcal{A}}(\mu)/f_{\mathcal{A}}(0)$ relatively to the ground (unperturbed) state $f_{\mathcal{A}}(0)$. This is meaningful, e.g., for naturally living systems where the ground state is measurable and is considered as an optimal performance state. Such a definition enables the comparison of some common property of different species. On the other hand, in cases when no ground state is given, the absolute value can provide more adequate measure of robustness.

On the technical side, the concept of robustness has been widely studied in the deterministic modelling framework based on ordinary differential equations (ODEs). There exist several well-established analytic techniques based on static analysis as well as dynamic numerical methods for effective robustness analysis of ODE models.

Several studies have brought formal methods into this concept. The idea is to formalise the analysed property in terms of temporal logic formulae and to employ quantitative satisfiability to measure the robustness of formula satisfaction. The first such framework has been implemented on the top of the BIOCHAM tool [162, 165] employing LTL extended with constraints (predicates) over reals to express properties of timed traces of real-valued states. The approach is "property-oriented" provided that the robustness is understood as a measure indicating how much the predicates in the formula can be altered in order to affect the validity of the formula wrt the given trace. The approach introduced in [168] and implemented in the TaLiRo tool [14] brings "behaviour-oriented" robustness degree measured against MTL formulae interpreted over discrete-time and continuous-time traces and discrete state space. The quantitative semantics of MTL for a given trace (signal) $\mathfrak{x}$ is defined as a distance of $\mathfrak{x}$ from the set of signals violating the formula. The approach using STL [151] extends the concept of behaviour-oriented quantitative reasoning to work with real-valued signals and to consider a perturbation of traces in time (time robustness), state space (space robustness), and the mix of both. The methods are implemented in the tool Breach [145]. All of these methods are applicable to cCRNMs, cINMs and cRINMs.

In the case of stochastic models (sCRNMs), the concept of robustness is not yet as established as in the case of ODE models. Komorowski etal. [249]

introduce the notion of robustness in the context of linear noise approximation of stochastic kinetics. To the best of our knowledge, our approach discussed in Chapter 6 is the first work addressing robustness in stochastic systems from the perspective of temporal properties (CSL). Alternative techniques employ statistical approaches and simulation working with STL specifications [40, 72].

2.7 Summary

In this chapter, we have set a unifying framework targeting the model classes most frequently used in computational systems biology. The framework can be easily extended to include other models used in biology (e.g., more complicated kinetics models of INs or RINs, discrete models of CRNs, etc.) Additionally, we have introduced the background on the semantics of the models with respect to a set of selected basic temporal logics and their extensions that have been already successfully used in systems biology.

Finally, we have given a brief overview of model analysis techniques based on formal methods. The selection is focused on the analysis of models under parameter uncertainty by means of formal methods based on model checking and monitoring procedures. These methods have been described in more detail as they are discussed in next chapters from the perspective of our original methods.

Chapter 3

Solved Problems

In this chapter we formulate three key problems that address the analysis of models included in the framework discussed in Chapter 2. These methods target analysis of parameterised models. In particular, we revisit the following problems from the perspective of formal methods: parameter synthesis and robustness analysis. Moreover, we introduce a novel problem of so-called parameter exploration that adapts the framework of parameter synthesis to the settings of stochastic models.

3.1 Parameter Synthesis

3.1.1 Problem Formulation

In general, the problem is, for a given parameterised biological network model and a given temporal property, to find all parameter values satisfying the given property. For non-stochastic models, the problem is precised as *parameter synthesis problem* formally stated in the following definition. In our case we target an instance of the problem that is stated from the perspective of the basic temporal logics.

Definition 3.1 (Parameter Synthesis Problem) *Let $\mathcal{M}_{\mathcal{P}_\chi}$ be a parameterised cCBNM or dINM. Let φ be a formula of LTL or CTL.*

The parameter synthesis problem *for $\mathcal{M}_{\mathcal{P}_\chi}$ is to find the maximal set $\mathbf{P} \subseteq \mathcal{P}_{\chi\mathcal{M}}$ of parameter valuations such that $\mathcal{M}_{\mathcal{P}_\chi}(\mu) \models \varphi$ for all $\mu \in \mathbf{P}$.*

3.1.2 Significance

The problem is highly relevant for systems biology because the models are usually built from (bio)physical first-principles that are mathematically represented by means of parameterised kinetic functions. Parameters typically reflect thermodynamics conditions in a particular cell and are very difficult to be obtained *in vivo*. Parameter synthesis gives the unique opportunity to analyse the model globally with unknown parameters. In particular, by parameter synthesis it can be decided if a given parameterised model can or cannot display behaviour described by a given temporal property. In positive case, all satisfying parameter values settings are returned. If stochastic dynamics is considered, the parameter synthesis procedure gives all parameter values that satisfy the property with a probability greater or equal than a given probability threshold.

3.1.3 Existing Solutions

The naïve solution to the problem is to enumerate the individual parameter valuations of the given parameters domain by running a separate model checking task for every parameter valuation. Such approach requires the set of admissible parameter valuations to be finite. This is obviously ensured in case of dINMs [62]. However, it is worth noting that in that case the number of admissible parameter valuations can grow combinatorially with respect to the number of influences affecting a certain species (indegree of the influence network nodes).

In the case of cBNMs and sCRNMs, the parameter domain is uncountable. Apparently, the naïve approach suffers from fundamental problems in such cases. In order to achieve exact approximation of the results with formal guarantees, a suitable finite partitioning of the parameter space is inevitable.

The very first algorithm for parameter synthesis of cBNMs employing model checking over discrete abstractions has been introduced by Batt et al. in [49, 368]. The algorithm in [49] (implemented in the RoVerGeNe tool) is sequential and relies on execution of two model checking procedures per each class of valuations. In the average case, the number of analysed parameter classes can be reduced by a suitable BDD representation of the parameter space. In [51] the authors target the piece-wise affine framework [135] adapted to regulatory networks. The notion of uncertainty is lifted to a higher level of abstraction provided that qualitative (symbolic) valuation is considered. That allows to efficiently employ the symbolic NuSMV model checker [112] for the parameter synthesis procedure. However, the problem of such solution is that it does not construct the satisfying parameter valuations set. In particular, for a given model with a symbolically encoded parameter perturbation it either answers $true$ provided that the perturbation entirely satisfies the formula, or $false$, provided that a single counter-example is returned by the model checker. The latter case first requires to check if the counter-example is not a false-positive path introduced by the abstraction. If the false-positiveness (spuriousness) is falsified, the procedure is further iterated with a new parameter perturbation excluding the non-satisfying parameter valuation inferred from the counter-example. As the counter-examples gives just a single such parameter valuation, the approach can be very close to the naïve solution.

Alternatively, parameter synthesis for cBNMs can be done approximately by using sampling of the parameter space joint with numerical simulation of the individual (approximate) signals as is done in [166, 145]. Such approach does not give formal guarantees but rather a result interpreted up-to some numerical error (the finite sampling of the parameter space gives an approximate under-approximation of the synthesised parameter valuations sets as the individual signals are simulated with the numerical error of the solver, as used, e.g., in the BIOCHAM tool [166]). To avoid the exhaustive (and unnecessary) sampling of the parameter space, sensitivity analysis of quantitative semantics of a temporal formula wrt changes in model parameters offers an efficient systematic way [150] (the approach is employed in the tool Breach [145]).

The problem of parameter synthesis has been also addressed in the settings of discrete time dynamical systems that are significantly different to models considered in this thesis. An example is technique based on the Bernstein polynomial representation that targets polynomial dynamical systems [127]. It is implemented in the tool SAPO [152].

3.1.4 Our Contribution

In our work, we have targeted the parameter synthesis problem with a fully automatised procedure working on common parameterised biological models. Firstly, we have developed *coloured model checking* – a method that adapts parallel model checking algorithms for several temporal logics and works with dINMs (directly) and cBNMs (by utilising the discrete abstraction procedures of ODEs). The method and its mapping to the problem formulated in Definition 3.1 is described in Chapter 4. The method expects encoding of the models by means of PKS and introduces symbolic representation of parameter values that overcomes the problems of large (or infinite) parameter spaces.

3.2 Parameter Exploration

3.2.1 Problem Formulation

In the case of stochastic models, the presence of a specified property in a given (parameter-uncertain) model is evaluated quantitatively with a given probability measure. To that end, the problem of parameter synthesis is reformulated in the quantitative context provided that the goal is to compute all parameter values that satisfy the property with probability greater or equal than the given threshold. Since it is necessary in this context to quantitatively evaluate parameter valuations with respect to a given formula, we speak about a more fundamental problem of *parameter exploration*. This problem is formally stated in the following definition.

Definition 3.2 (Parameter Exploration Problem) *Let $\mathcal{M}_{\mathcal{P}_\chi}$ be a sCRNM parameterised with a parameterisation $\chi_{\mathcal{M}}$. Let φ be a formula of the bounded time fragment of CSL with rewards.*

The parameter exploration problem *for a parameterised sCRNM $\mathcal{M}_{\mathcal{P}_\chi}$ is to compute for each state $X \in \mathbb{X}$ of the parameterised CTMC $[\![\mathcal{M}_{\mathcal{P}_\chi}]\!]_{\mathrm{sCRNM}}$ the* landscape function $\lambda_X^{\varphi,\mathcal{P}_{\chi_{\mathcal{M}}}} : \mathcal{P}_{\chi_{\mathcal{M}}} \to \mathbb{R}_{\geq 0}$ *that for each parameter valuation $\mu \in \mathcal{P}_{\chi_{\mathcal{M}}}$ returns the numerical value of the probability or the expected reward for the formula φ.*

3.2.2 Significance

From the same reasons as mentioned in Section 3.1.2, the problem of parameter exploration of stochastic models is highly relevant for systems biology. Parameters reflect thermodynamics and physical conditions that control the amount of noise that makes the modelled reactive events uncertain. Since these parameters target the very detailed mechanistic aspects of chemical reactions occurring *in vivo*, it is even more difficult to obtain their values than in the deterministic case. Parameter exploration enables analysis of stochastic model dynamics without requiring precise knowledge of the parameter values.

3.2.3 Existing Solutions

The parameter synthesis problem for a parameterised CTMC and a bounded reachability specification was introduced in [212]. The authors reduce the problem to the analysis of the polynomial function describing how the reachability probability depends on the parameter values. The degree of the polynomials is determined by the number of uniformisation steps which is typically very high. Therefore only an approximate solution is obtained through the discretisation of the parameter space. An alternative solution is provided by statistical approaches based on assuming simplifying restrictions on the stochasticity of systems dynamics. In that case, statistical guarantees are obtained instead of exact guarantees. Approaches based on Markov Chain Monte-Carlo sampling and Bayesian inference [200, 232, 247] can be extended to sample-based approximation of the evaluation function, but at the price of undesired inaccuracy and high computational demands [27, 61]. A promising approach working with STL is proposed in [73] where it is shown that satisfaction probability is a smooth function of model parameters. A Bayesian statistical algorithm is employed to perform model checking for all isolated parameter values simultaneously. This is done by exploiting the smoothness of the quantitative satisfaction function. Correlations are modelled explicitly through a prior distribution over a space of smooth functions (a Gaussian Process). Observations at individual parameter values are collected to construct an analytical approximation of the satisfaction function itself. The method is implemented in the tool U-Check [72]. It is considerably faster than traditional approaches to statistical model checking [232, 116].

3.2.4 Our Contribution

We have given first steps towards exact (precise) probabilistic model checking of sCRNMs under parameter uncertainty by introducing *parameterised uniformisation* method that enables quantitative model checking of CTMCs with uncertain parameters against formulae specified in bounded-time fragment of CSL with rewards. The method computes an approximation of the landscape function and it is unique in giving formal guarantees for the errors caused by the approximation. The methods and its mapping to the problem formulated in Definition 3.2 is described in Chapter 5.

3.3 Robustness Analysis

3.3.1 Problem Formulation

Assuming the concept of robustness as declared by Kitano, the goal of robustness analysis is to evaluate a certain functionality of systems dynamics with respect to perturbation in systems parameters. The approach employing formal methods understands such functionality as an arbitrary (qualitative or quantitative) temporal property.

It is worth noting that in following definitions, the robustness degree is simplified by excluding the probability of perturbation occurrence. However, a uniform distribution of perturbations can be considered without any limitations when computing the robustness degree. In cases when the distribution of perturbations is not uniform, computation of the robustness degree needs to be revised accordingly. In all cases studies we have conducted in our research, independent perturbations or a uniform distribution of perturbation is used.

Owing to the fact the robustness degree is inherently a quantitative measure, the problem of evaluating robustness of a given temporal property requires to introduce a computable evaluation function that can quantitatively characterise the amount of work that is needed to perturb the system in such a way that the property becomes violated. Such a function has to reflect whether the systems behaviour is deterministic or stochastic.

In the case of deterministic systems, every behaviour (represented as a signal) has a unique set of parameter values under which it is present in the system. Cardinality of such set then intuitively reflects the robustness of the considered behaviour. We target the notion of space robustness introduced in [151] from the perspective of the expressive variant of the value-freezing signal temporal logic STL* interpreted over continuous-time signals. The problem is formally defined as follows.

Definition 3.3 (Deterministic Robustness for STL*) *Let $\mathcal{M}_{\mathcal{P}_\chi}$ be a parameterised cBNM and let $\mathbf{P} \subseteq \mathcal{P}_{\chi_\mathcal{M}}$ be a bounded set of perturbations. Let φ be a formula of STL*. The* robustness problem for $\mathcal{M}_{\mathcal{P}_\chi}$ wrt φ and $\mathbf{P}$ *consists of the following parts:*

1. *to define and approximately compute the* evaluation function $D_\varphi^\mathcal{M} : \mathbf{P} \to \mathbb{R}$ *such that for any $\mu \in \mathbf{P}$, $D_\varphi^\mathcal{M}(\mu)$ evaluates quantitative semantics of φ on the signal $[\![\mathcal{M}_{\mathcal{P}_\chi}(\mu)]\!]_{\text{cBNM}}$,*

2. *to approximately compute the* robustness degree of φ in $\mathcal{M}_{\mathcal{P}_\chi}$ wrt $\mathbf{P}$ *defined as:*
$$\text{rd}_{\varphi,\mathbf{P}}^\mathcal{M} \overset{def}{=} \int_\mathbf{P} D_\varphi^\mathcal{M}(\mu) d\mu$$

In the case of stochastic systems, the situation is more difficult. Since every behaviour is distorted with noise and randomness of relevant events occurring in time, we have to deal with the shape of probabilistic distributions of system states and its development in time (transient analysis) or its shape in the steady state (steady state analysis). In our setting, we assume the behaviour is described by means of a probabilistic temporal property. The evaluation function reflects the probability of the property being satisfied under the given perturbation. The robustness is inversely proportional to the amount of noise that causes potential violation of the formula. In contrast to the deterministic case, in this case the robustness is understood as an inherent property of the systems stochasticity.

Definition 3.4 (Probabilistic Robustness for CSL) *Let $\mathcal{M}_{\mathcal{P}_\chi}$ be a parameterised sCRNM. Let φ be a formula of CSL with rewards. The* robustness problem for $\mathcal{M}_{\mathcal{P}_\chi}$ wrt φ and $\mathbf{P}$ *is to approximately compute the* robustness degree $\text{rd}_{\varphi,\mathbf{P}}^\mathcal{M} \overset{def}{=} \int_\mathbf{P} D_\varphi^\mathcal{M}(\mu) d\mu$ *where the* evaluation function $D_\varphi^\mathcal{M} : \mathbf{P} \to \mathbb{R}$ *is defined in a way that for every $\mu \in \mathbf{P}$, $D_\varphi^\mathcal{M}(\mu)$ reflects the quantitative semantics of φ in a suitable manner.*

3.3.2 Significance

The problem of evaluating robustness is important to obtain deeper understanding of the role of parameters on the presence of the specified behaviour [343]. Robustness degree provides a tool that significantly helps to compare several models presenting the same behaviour [48]. There are many examples in systems biology literature that have used robustness analysis to get mechanistic insights into a certain phenomenon. For example, in [4, 30] the concept of robustness is used to correctly reconstruct the mechanism of bacteria chemotaxis control. The analysis conducted in [338] provides a study of the organisation of a two-component signalling pathway in *E. coli* from the perspective of global concentration robustness of individual components in the pathway.

The robustness degree also provides a computable measure allowing targeted optimisation of cellular processes by means of tuning of known parameters [148, 319]. This is an inevitable precursor of successful design and engineering of synthetic cellular circuits [238, 132].

3.3.3 Existing Solutions

In the context of deterministic models, robustness analysis with respect to functionality specified in terms of temporal formulae has been recently introduced [168, 319]. There exist two major approaches of defining and analysing robustness. If only the parameters of the model are perturbed, we speak of a *behaviour-oriented approach* to robustness. This approach has been explored by Fainekos and Pappas [168], further extended by A. Donzé et al. [151] and implemented in the toolbox Breach [145]. Another way to look at perturbations is from the perspective of property uncertainty. If the system is considered fixed and all parameters exactly known, the uncertainty then lies in the property of interest. For a specific property it can be explored how much would have the atomic constraints in the formula to be altered in order to affect the property's validity in the given model. This approach has been adopted by Fages et al. [319] and implemented in the tool BIOCHAM [167]. When only the parameters of the property are perturbed, it is the case of a property-oriented approach to robustness.

The work presented in [40] is based on the idea of directly adapting the concept of behaviour-oriented robustness to stochastic models. Individual simulated trajectories of the CTMC are locally analysed with respect to an STL formula. For each simulated trajectory, the so-called satisfaction degree representing the distance from being (un)satisfied is computed, thus resulting into a randomly sampled distribution of the satisfaction degree. This distribution thus gives another source of information in addition to the probability of formula satisfaction (the fraction of valid trajectories in the sampled set).

3.3.4 Our Contribution

Our contribution to robustness analysis with formal methods is two fold. Firstly, we have extended the set of techniques used to analyse robustness on timed traces (signals) generated from continuous-time models (cRINMs, cINMs and cCRNMs). In particular, we have defined an extension of signal temporal logic called STL* that allows to express freezing of values referred within temporal operators. The extension is important especially to express several aspects of signals that cannot be expressed in plain STL (e.g., presence of local extremes and their mutual relationships). In consequence, we have defined quantitative semantics of STL* that fits well in the existing robustness analysis framework developed for plain STL. The algorithm for computing the quantitative semantics is implemented within the PARASIM tool. The method and its mapping to the problem formulated in Definition 3.3 is described in Chapter 6. Our result makes a fundamental step towards fully automatised analysis of complex properties over timed sequences and is universally applicable in the wide domain of (cyber-)physical and (cyber-)biological systems. It is already received well by the community [44, 102, 296].

Secondly, we have introduced one of the first property-based formulations of robustness degree in the context of continuous-time stochastic processes. To that end, we have adopted the concept of parameterised uniformisation as the underlying computational machinery and have used the landscape function as the main tool for inferring robustness of stochastic dynamics. The method and its mapping to the problem formulated in Definition 3.4 is described in Chapter 6. In the community, our result is considered as one of the fundamental approaches to the problem [373].

Chapter 4

Coloured Model Checking

As described in Chapter 3, to solve the parameter synthesis problem, we have developed a method called coloured model checking (CMC) that is based on enumerative model checking. In this chapter, we summarise our results regarding the development, implementation, and evaluation of the CMC method. In particular, we give definitions of the variants of the CMC problem targeting different temporal logics (LTL and CTL).

4.1 Overview

In Figure 4.1, the general scheme of the coloured model checking is shown. The method assumes the model to be encoded as a parameterised Kripke structure. The method is used to solve the parameter synthesis problem I (see Definition 3.1). It works for cBNMs and dINMs (encoding of cBNMs and dINMs in terms of PKSs is described in Chapter 2).

As a general method, coloured model checking works on any parameterised Kripke structure for which the local parameterisation problem can be decided for every pair of states. The individual variants of the method differ in the temporal logic employed. The chosen logic significantly affects the algorithmics as well as the interpretation of the results. We employ enumerative algorithms for both kinds of logics. CTL characterises the temporal properties in individual states and that way allows to capture global properties of systems dynamics. Validity of a given formula is typically evaluated in all states of the system. On the other hand, LTL focuses on properties of paths and the analysis is typically localised into a given initial state (or a set of initial states) from where all the reachable paths are explored. In the context of the parameter synthesis problem, such difference affects the interpretation of the obtained results for the original biological network model.

Since the parameter space cardinality increases exponentially with the number of unknown parameters, our primary goal is to reduce the com-

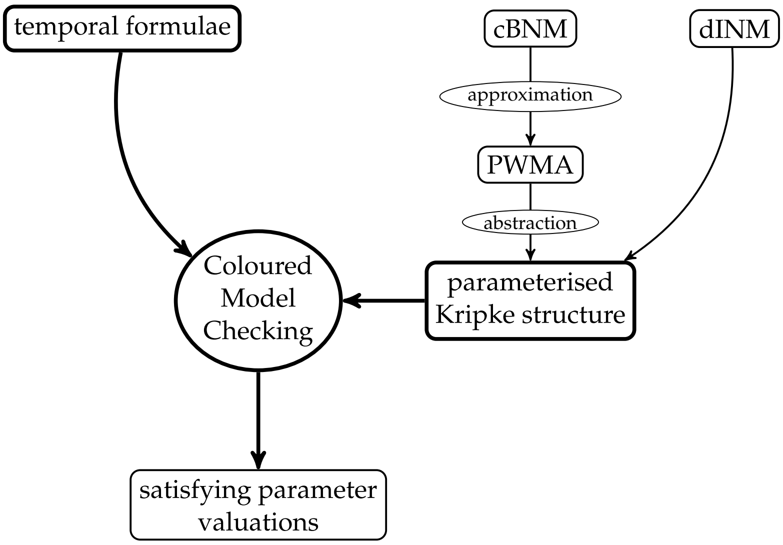


Figure 4.1: General workflow of parameter synthesis by coloured model checking.

plexity for the expected case. To this end, the method is appropriate for models satisfying the following requirements:

1. The transition-enabling parameter values sets $\mathcal{P}(s, s')$ can be computed efficiently from the knowledge of the model and the state s. In particular, we suppose there exists an implicit representation of these sets.

2. A small change in the value of a single parameter causes only a local change in the transition relation. This implies that sets $\mathcal{P}(s, s')$ represent significant portions of the parameter space (transition relations of respective non-parameterised Kripke structures are, to a large extent, similar).

Note that cBNMs and dINMs exhibit such properties. The primary goal of CMC is to provide an algorithm which in practice performs parameter synthesis on suitable models reasonably fast and which is effectively parallelisable.

4.2 Coloured LTL Model Checking

The problem of coloured LTL model checking for a given PKS and a given formula is to explore every path starting in some initial state and to find an exact set of parameter values for which the formula is satisfied on the path.

Formally, the problem is formulated using the automata-based LTL model checking theory. In the standard way, the given LTL formula is translated to a Büchi automaton (a so-called never-claim) that accepts the language covering exactly the Kripke structure paths not satisfied by the formula. The intuition behind the LTL CMC is given in Figure 4.2. The notion of Kripke structure extended with the set of accepting states F is considered here. The following definition provides a formal description of the problem.

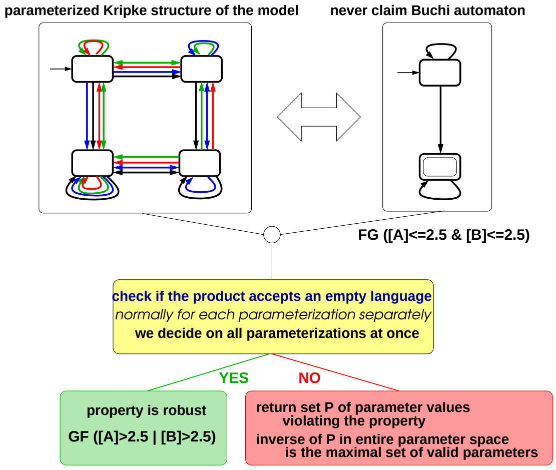

parameterized Kripke structure of the model
never claim Buchi automaton
FG ([A]<=2.5 & [B]<=2.5)
check if the product accepts an empty language
normally for each parameterization separately
we decide on all parameterizations at once
YES
NO
property is robust
GF ([A]>2.5 | [B]>2.5)
return set P of parameter values
violating the property
inverse of P in entire parameter space
is the maximal set of valid parameters

Figure 4.2: Intuition behind the coloured LTL model checking method.

Definition 4.1 *Let $\mathcal{K} = (\mathcal{P}, S, S_0, F, \rightarrow, L)$ be a parameterised Kripke structure over* AP*. Let further φ be an LTL formula over* AP*. We define the* coloured LTL model checking problem *as a problem of finding maximal set of parameter values $P \subseteq \mathcal{P}$ such that for all $p \in P$, all $s_0 \in S_0$ and every fair infinite path $\pi_p(s_0)$ in $\mathcal{K}$, $\pi_p \models \varphi$ holds.*

4.2.1 Algorithms

The basic step is to construct the synchronous product $\mathcal{K} \times \mathcal{B}$ of the PKS $\mathcal{K}$ and the never-claim automaton $\mathcal{B}_{\neg\varphi}$. We have presented the details of the construction in [38] where it is shown the product is a Büchi automaton that contains exactly those infinite paths of $\mathcal{K}$ on which the automaton $\mathcal{B}$ accepts. Hence every π_p in $\mathcal{K} \times \mathcal{B}_{\neg\varphi}$ makes a path in $\mathcal{K}$ that violates the formula φ. As a result, the problem of finding the maximal $P \subseteq \mathcal{P}$ such that for all $p \in P$ every π_p in $\mathcal{K}$ satisfies φ can be efficiently reduced to the problem of finding the maximal set $P \subseteq \mathcal{P}$ such that for every $p \in P$, there exists no π_p in the product $\mathcal{K} \times \mathcal{B}_{\neg\varphi}$.

A path π_p is in PKS $\mathcal{K}$ if and only if there exist states $s_0 \in S_0$, $s \in F$ such that $s_0 \xrightarrow{p}{}^{*} s \xrightarrow{p}{}^{+} s$. The problem is decomposed into two subtasks:

1. For each $s \in F$ compute the maximal set of parameter values $P \subseteq \mathcal{P}$ such that for all $p \in P$ there is an initial state $s_0 \in S_0$ such that $s_0 \overset{p}{\to}^{*} s$.
2. For each $s \in F$ and the corresponding parameter value set $P \subseteq \mathcal{P}$, determine if there is $p \in P$ and a path π_p that cycles on s.

We need to compute sets of parameter values (subsets of $\mathcal{P}$, each intuitively represented as a distinct *colour*) under which an accepting execution trace is enabled. To this end, colours are accumulated in states provided that each state is assigned all colours under which it is reached. We define *colouring* – a mapping of parameter value sets to states – as a function $Col \colon S \to 2^{\mathcal{P}}$. To detect a path cycling on a state $s \in F$, we compute the set of colourings $\mathrm{Succ}(s, P)$ which is a function $\mathrm{Succ}(s, P) : S \to 2^{\mathcal{P}}$ assigning to each state $s' \in S$ the maximal set of parameter values $P' \subseteq P$ such that $s \overset{p}{\to}^{+} s'$ for all $p \in P'$.

Algorithm 1 performs the computation in the manner described above. The algorithm iteratively accumulates the parameter value set P and can terminate as soon as $P = \mathcal{P}$, which allows for a fast discovery of paths violating a given property.

Algorithm 1 Coloured LTL Model Checking over the product BA $\mathcal{K} \times \mathcal{B}$

```
Require: $\mathcal{K} \times \mathcal{B}$
Ensure: $p \in P$ iff $s_0 \overset{p}{\to}^{*} s \overset{p}{\to}^{+} s$ for some $s_0 \in S_0$ and $s \in F$
$P \leftarrow \emptyset$
compute $\mathrm{Succ}(s_0, \mathcal{P})$
assign each state $s \in S$ a colouring $Col(s) \leftarrow \mathrm{Succ}(s_0, \mathcal{P})(s)$
for all $s \in F$, $Col(s) \setminus P \neq \emptyset$ do
    $P \leftarrow P \cup \mathrm{Succ}(s, Col(s) \setminus P)(s)$
```

Algorithm 2 computes $\mathrm{Succ}(s, P)$. It starts with an empty colouring and incrementally updates it. An update is a tuple (s, P), meaning that the set of parameter values $P \subseteq \mathcal{P}$ should be added to the colouring for the state $s \in S$. The set of pending updates is stored in Q. The algorithm terminates as soon as there are no more pending updates.

By $Q(s) = \bigcup\{P \subseteq \mathcal{P} \mid (s, P) \in Q\}$ we denote the set of parameter values that are currently scheduled for addition to $Col(s)$. To prevent Q from containing multiple updates for the same state, we use the merge operation $Q \oplus Q'$ defined as $Q \oplus Q' = \{(s, P) \mid P = Q(s) \cup Q'(s) \wedge P \neq \emptyset\}$ to update Q (line 11).

4.2.2 Interpretation of Parameter Synthesis Results

Coloured LTL MC provides a solution to the parameter synthesis problem stated in Definition 3.1 and can be thus applied to dINMs and cBNMs. In

Algorithm 2 Compute $\mathrm{Succ}(s_0, P)$ over the product BA $\mathcal{K} \times \mathcal{B}$

```
Require: $\mathcal{K} \times \mathcal{B}$; $P \subseteq \mathcal{P}$; $s_0 \in S$ ($s_0$ can be an arbitrary state)
Ensure: $\forall s \in S : Col(s) = \mathrm{Succ}(s_0, P)(s)$
for all $s \in S$ do
    $Col(s) \leftarrow \emptyset$
$Q_{new} \leftarrow (s_0, P)$
while $Q_{new} \neq \emptyset$ do
    $Q_{old} \leftarrow Q_{new}$
    $Q_{new} \leftarrow \emptyset$
    while $Q_{old} \neq \emptyset$ do
        remove $(s, P)$ from the head of $Q_{old}$
        if $P \nsubseteq Col(s)$ then
            $Col(s) \leftarrow Col(s) \cup P$
            $Q_{new} \leftarrow Q_{new} \oplus \{(s', P \cap \mathcal{P}(s, s')) \mid s \rightarrow s', s' \in S\}$
```

both cases, it is important to comment on how to correctly interpret the results. In particular, we consider a universally interpreted LTL (the formula holds in a given initial state of a Kripke structure if it holds for all infinite paths starting in that state). This affects the results interpretation. In general, the result of the procedure is a set of all parameter values violating the formula. Its complement in the entire parameter space is the set of all parameter values that satisfy the formula.

It is important to note that the product automaton $\mathcal{K} \times \mathcal{B}$ compactly represents the (infinite) set of paths that covers exactly witnesses of violation of the formula (so-called counterexamples). This witnesses can be used for further processing of the results.

In any case, when interpreting the results for the individual types of models as depicted in Figure 4.1, we also need to reflect the particular encoding of the models in PKS.

dINM

In the case of a dINMs $\mathcal{M} = (\mathcal{N}, \nu_{dIN}, \mathbb{X}, X_0, \mathbf{m})$ with a parameterisation $\chi_{\mathcal{M}}$ the parameterised Kripke structure encodes the model semantics ($[\![\mathcal{M}_{\mathcal{P}_\chi}]\!]_{\mathrm{dINM}}$). In consequence, the results of parameter synthesis using LTL CMC can be directly interpreted on parameterised dINMs.

In [244] we explore the enumeration of finite (lasso-shaped) witnesses for every satisfying parameterisation. In particular, we propose several criteria allowing to classify parameterisations based on sets of finite witnesses associated to them.

cBNM

In the case of cBNMs, the parameterised Kripke structure does not reflect the model dynamics directly due to the approximation and abstraction steps employed on a given cBNM. Especially, the uncountable parameter space of the cBNM is partitioned into a finite number of equivalence classes with respect to a particular settings of approximation/abstraction procedures. There are several issues to be discussed in such case in order to correctly interpret the results of CMC for the original model.

First, the approximation step affects the vector field of the original dynamical system that determines a given cBNM when turning it into the PWMA system. In particular, the trajectories of the original system are distorted. To the best of our knowledge, no techniques have yet been developed to characterise qualitatively or quantitatively such distortion. In [205], there is introduced an algorithm for optimal approximation that associates every approximated vector field function with an approximation error. However, it is not known how to project the computed error onto the trajectories.

Second, the abstraction technique produces a Kripke structure that makes an over-approximation of the PWMA system. In consequence, some of counterexamples found by model checking can be false-positive paths. The reasons are discussed in Sections 2.4.6 and 2.5.6. Therefore for universally interpreted LTL and ACTL the resulting parameter values in the unsatisfying set computed by LTL CMC performed on a PKS $\mathcal{K}^{abst}$ obtained by abstracting a cBNM $\mathcal{M}$ make an over-approximated set of parameter values that really satisfy the formula in the intermediate PWMA. Analogously, the resulting parameter values in the satisfying set are under-approximated with respect to the abstraction (see the solid polygon in Figure 2.8). In conclusion, the CMC method can be safely applied to PWMAs of cBNMs while carefully considering the over/under-approximation issues. However, the result of CMC cannot be exactly interpreted at the level of the original cBNM.

4.2.3 Performance Evaluation

On the theoretical side, the worst case time complexity of the overall Algorithm 1 for a given product $\mathcal{K} \times \mathcal{B} = (\mathcal{P}, S, S_0, \rightarrow, L, F)$ automaton belongs to $\mathcal{O}(|F||S|^2\zeta(|\mathcal{P}|))$ as discussed in detail in [33]. The linear dependence on the number of accepting states (affected by the given formula) comes from lines 4-5 in Algorithm 1. Note that the LTL formula is expected to be already translated into the Büchi automaton (LTL model checking is exponential in the size of the formula and so is LTL CMC). The quadratic dependence on the state space size comes from line 11 in Algorithm 2. The dependence on the size of the parameter space $\mathcal{P}$ is included in the function

ζ. It depends on the model type and complexity of the colouring maintenance (the union and intersection on lines 10-11 in Algorithm 2). In the case of cBNMs with mathematically independent parameters, these operations on parameter sets can be performed in constant time due to representation of parameter values as continuous intervals of reals. In the case of dINMs, we have considered explicit representation of the parameter sets encoded by means of bit vectors. Implementation of these operations is reduced in this case to operations on bit vectors.

From the practical point of view, we have performed several experiments by employing two independent prototype multi-core implementations working on cBNMs and dINMs respectively. In the case of cBNMs, we were able to compute models up-to 200 thsd states (7-dimensional model) with up to 3 parameters unknown (3-dimensional parameter space) in 30 minutes on a common hardware (details in [38]). In the case of dINMs, we have employed enumerative computation of the intersection/union operations by using binary encoding of the parameter values. This has allowed us to compute large parameter spaces having up-to 100 thousand billions of parameter values on models with thousands of states in several seconds [244]. The low numbers of states are typical for this kind of models.

4.2.4 Publications Summary

First results introducing this method have been published in [38] primarily focused on cBNMs. The paper also provides complexity analysis of the problem and discusses a mechanism for reducing the extent of over-approximation by means of fairness. Consequently, in [33] the method has been generalised for both cBNMs and dINMs and applied to several biological case studies. Please note that for purpose of this thesis we have simplified and changed some of the notation.

The work has been realised in collaboration with my colleagues Luboš Brim, Jiří Barnat with whom I have developed the method. Bringing the method on a paper and in a code has been realised with a significant help of master students Martin Vejnár, Tomáš Vejpustek, Adam Streck and Adam Krejčí. My contribution is in setting up the problem for biological networks, guiding and partially realising the first formalisation of the method including the formal analysis of its correctness. In the case study part, I have made a selection of models useful for fine-tuning and demonstrating of the method to systems biologists. Finally, I have built up a scenario followed in experimentation procedures realised with all these models.

The work on LTL CMC for dINMs has been realised with the help of my master's students Adam Streck and Juraj Kolčák. The first implementation has made a part of the Esther web-based tool [341]. Based on this result, we have started a collaboration with Hannes Klarner and Heike Siebert (FU Berlin) resulting with a paper [244] that gives a new view on using

CMC for dINMs by means of introducing quantitative measures allowing to rank the synthesised parameter values concerning their fitness with a given set of experimental data. My contribution to that work was in taking a significant part in designing and setting up the problem and adapting the LTL CMC for the new problem.

4.3 Coloured CTL Model Checking

The problem of coloured CTL model checking for a given PKS and a given formula is in our case to obtain a table where every system state is associated with an exact set of parameter values for which the formula is satisfied. Formally, such table is formalised using a function $\mathcal{F}$, formally defined in the following definition.

Definition 4.2 *Let* $\mathcal{K} = (\mathcal{P}, S, S_0, \rightarrow, L)$ *be a parameterised Kripke structure over* AP*. Let further* φ *be a CTL formula over* AP*. The* coloured CTL model checking problem *is, given* $\mathcal{K}$ *and* φ*, to find the function* $\mathcal{F}$ *described as follows:*

$$\mathcal{F}(s) = \{p \in \mathcal{P} \mid s \models_{\mathcal{K}_p} \varphi\}. \tag{4.1}$$

4.3.1 Algorithms

To efficiently compute the problem, we developed the distributed-memory semi-symbolic CMC algorithm that solves the problem of finding the function $\mathcal{F}$. The algorithm for computing $\mathcal{F}$ is a modification of the (explicit) labelling CTL model checking algorithm [117]. It labels states with "coloured" subformulae of φ that are satisfied in the state of the Kripke structure $\mathcal{K}_p$ for the "colour" $p \in \mathcal{P}$.

Algorithm 3 Compute φ-satisfying Colouring

```
Require: parameterised KS $\mathcal{K}$ and CTL formula $\varphi$
Ensure: $\mathcal{F}$
  for all $i \leq |\varphi|$ do      ▷ compute the sets $ColSat(\varphi) = \{(p,s) \in \mathcal{P} \times S \mid s \models_{\mathcal{K}(_p} \varphi\}$
      for all $\psi \in cl(\varphi)\ with\ |\psi| = i$ do
          compute $ColSat(\psi)$ from $ColSat(\psi')$     ▷ for maximal genuine $\psi' \in cl(\psi)$
  return $\{(p,s) \in \mathcal{P} \times S \mid (p,s) \in ColSat(\varphi)\}$
```

State Colouring

Similarly to the LTL case, the notion of the colouring is employed to represent the assignment of parameter values to states. In contrast to the LTL case where the LTL formula is encoded implicitly in the product automaton, here the Colouring is defined as a function over formulae. One of the reasons is that the labelling algorithm computes satisfaction for all subformulae of the given formula. To that end, for a given CTL formula φ the notion of Colouring is lifted to a relation called *φ-satisfying colouring*, denoted $ColSat(\varphi)$, defined as $ColSat(\varphi) = \{(p, s) \in \mathcal{P} \times S \mid s \models_{\mathcal{K}_p} \varphi\}$. Intuitively, it assigns the formula with a set of pairs containing a parameter value p and a state s such that the formula is satisfied in s under the parameter value p. The algorithm for computing the function $\mathcal{F}$ operates recursively on the structure of φ starting from atomic propositions and iteratively computing $ColSat(\varphi)$. The computation is done in standard way using the labelling algorithm [117]. Its basic idea is described in Algorithm 3. The computation of $ColSat(\psi)$ from the results obtained for subformulae of ψ (having the size smaller by one wrt ψ) reflects the particular temporal operator applied to the subformulae. Since here we focus on parallel algorithms that require further extensions to the standard labelling-based model checking algorithm, description of the labelling procedure extended to the parameterised settings is included as a part of the following steps.

Parallelisation with Assumption-Based Semantics

To parallelise the procedure, we adapt the assumption-based distributed CTL model checking algorithm [91]. By following that approach, the algorithm is run on a cluster of n computational nodes (workstations). Each workstation owns a part of the original PKS as defined by a partition function. This part is extended with the so-called *border states*. Intuitively, border states are states that in fact belong to another computational node and represent the missing parts of the state space. They serve as a proxy between two parts.

More precisely, the fundamental notion is a *PKS fragment $\mathcal{K}_i$* which makes a substructure of the PKS $\mathcal{K}$ satisfying the property that every state in $\mathcal{K}_i$ has either no successor in $\mathcal{K}_i$ or it has exactly the same successors as in $\mathcal{K}$. The states without any successors in $\mathcal{K}_i$ are called the *border states* of $\mathcal{K}_i$. A partition of the PKS $\mathcal{K}$ is a finite set of PKS fragments $\mathcal{K}_1, \ldots, \mathcal{K}_n$ such that every state of $\mathcal{K}$ is present in exactly one $\mathcal{K}_i$ as a non-border state; it may be present in several other $\mathcal{K}_j$ as a border state. In fact, every border state is stored several times: as original one on the node that owns it and as duplicates on nodes that own its predecessors.

The information in border states is considered with the notion of the truth under assumption. For a given formula φ, this is captured by the so-called *assumption function*, $\mathcal{A} : \mathcal{P} \times S \times cl(\varphi) \rightarrow Bool$ that for a given parameter value, state and a subformula of φ returns the *assumption* on satisfaction of the subformula in the given state and under the given parameter value. An important fact is that the value returned by $\mathcal{A}$ can be considered undefined. The intuition is the following: $\mathcal{A}(p, s, \varphi) = \texttt{tt}$ if we can assume that φ holds in the state s under parameter value p, $\mathcal{A}(p, s, \varphi) = \texttt{ff}$ if we

Algorithm 4 Main Idea of the Parallel Algorithm

```
Require: parameterised KS $\mathcal{K}$, CTL formula $\varphi$, partitioning function
Ensure: $\mathcal{F}$
  Partition $\mathcal{K}$ into $\mathcal{K}_1, \ldots, \mathcal{K}_n$
  for all $\mathcal{K}_i$ where $i \in \{1, \ldots, n\}$ do in parallel
      Take the initial assumption function
      repeat
          Update the assumption function using the node algorithm;
          Exchange relevant information with other nodes;
          Modify assumption function;
      until all processes reach fixpoint
```

can assume that φ does not hold in the state s under parameter value p, and $\mathcal{A}(p, s, \varphi) = \perp$ if we cannot assume anything.

The coloured model checking problem is then revisited with the assumption function in the following way:

$$ColSat(\varphi) = \{(p, s) \in \mathcal{P} \times S \mid \mathcal{A}(p, s, \varphi) = \texttt{tt}\}$$

In general, computation of the assumption function adopts the labelling algorithm. It starts from setting its values for all states to $\perp$ and updating the results iteratively in individual fragments. Results depending on the bordering states are completed once all the necessary information is collected. Finally, the assumption function is completely defined and must return either true or false for every state.

In more detail, the main idea of the entire parallel computation, summarised in Algorithm 4, is the following. Each fragment $\mathcal{K}_i$ is managed by a separate process (node) $Proc_i$. These processes are running in parallel (simultaneously on each node). Each process $Proc_i$ initialises the assumption function $\mathcal{A}_i$ to the undefined assumption function $\mathcal{A}_\perp$. After initialisation, it computes the new assumption function from the initial assumption function using the so-called node algorithms described below. The node algorithms extend the standard labelling algorithms defined for individual temporal operators to the assumption-based and parameterised setting.

Once the algorithm has finished computing the assumptions, the node exchanges information about border states with other nodes. It sends to each other node the information it has about that node's border states and receives similar information from other nodes. After this exchange is completed, the computation is restarted. These steps are repeated until the whole network reaches a fixpoint, i.e., until no update is received by any node.

Assumption Function Computation

The main operation of the parallel algorithm is the iterative computation of the assumption functions starting from the simplest subformulae of (the atomic propositions) and moving towards φ by structural induction. The algorithm takes into account the assumptions of border states, initially set to $\perp$. Since the algorithm traverses the formula by means of structural induction, the central role in the algorithm play the so-called node algorithms that compute the assumption function for a subformula of a particular shape (determined by a specific temporal operator). Each of these algorithms assumes that all possible assumptions for all subformulae have been already computed (given the current assumptions on border states).

Algorithm 5 Compute explicit assumptions for **EX**ψ

Require: PKS fragment $\mathcal{K}_i$, CTL formula $\varphi = \mathbf{EX}\psi$, initial assumptions $\mathcal{A}_{in}$
Ensure: new assumptions $\mathcal{A}$

```
$\mathcal{A} := \mathcal{A}_{in}$
set $\mathcal{A}(p, s, \varphi) := \texttt{ff}$ for all non-border states $s$ and all $p \in \mathcal{P}$
$init := \{(s, P) \mid P = \{p \in \mathcal{P} \mid \mathcal{A}_{in}(p, s, \psi) = \texttt{tt}\}\}$
for $(s, P)$ in $init$ do
    for $(s', P')$ such that $P' = P \cap \mathcal{P}(s', s) \neq \emptyset$ do
        for all $p \in P'$ do
            $\mathcal{A}(p, s', \varphi) := \texttt{tt}$
```

To give an intuition on the role of a node algorithm, we show two variants for temporal subformulae of the form **EX**ψ (existential next) presented in Algorithm 5 and Algorithm 6. The variants differ in encoding of the parameter values sets. In general, the assumption function is initiated to "false for all parameter values". Then, the algorithms iteratively collect assumptions about ψ and propagate the information into predecessor states. This propagation extends the set of parameters for which the assumption is true and reduces the set of parameters for which the assumption is false. This ensures that if a state under a given parameter value has at least one successor that satisfies ψ (under the same parameter value), this value is going to be included in the true assumption formula for that state. Moreover, if all successors of a state under a given parameter value refute ψ, that value is going to be included in the false assumption formula for that state. Finally, if a state under given parameter value has no successors that satisfy ψ and at least one successor whose satisfaction of ψ is undefined in the current assumption, this parameter value is not going to be included in either the true or false assumption function.

In the case of interval-based parameter encoding, the assumption function is represented explicitly. In particular, the explicit assumption function requires the evaluation of the formula for a given parameter value.

Operations over the parameter space that are typically performed in node algorithms are union and intersection of parameter value sets. More specifically, both operations can be demonstrated on an example of the node algorithm presented for formula of the shape **EX**ψ (Algorithm 5). In that case, all parameter values satisfying ψ in a state s are intersected with the set of parameter values enabling a transition from s', a predecessor of s, to s (line 5). That way the satisfying parameter values are back-propagated to predecessors of s and the assumption function is updated for the formula **EX**ψ.

Algorithm 6 Compute symbolic assumptions for **EX**ψ

```
Require: PKS fragment $\mathcal{K}_i$, CTL formula $\varphi = \mathbf{EX}\psi$, initial assumptions $\widetilde{\mathcal{A}}_{in}$
Ensure: new assumptions $\widetilde{\mathcal{A}}$
1: $\widetilde{\mathcal{A}} := \widetilde{\mathcal{A}}_{in}$
2: set $\widetilde{\mathcal{A}}(s,\varphi) := (\mathtt{ff},\mathtt{tt})$ for all non-border states $s$
3: $\mathit{init} := \{(s,\Phi_t,\Phi_f) \mid \widetilde{\mathcal{A}}_{in}(s,\psi) = (\Phi_t,\Phi_f)\}$
4: for $(s,\Phi_t,\Phi_f)$ in $\mathit{init}$ do
5:     for $s' \in S_i$ such that $\Phi_{s',s}$ is satisfiable do
6:         $\widetilde{\mathcal{A}}^t(s',\varphi) := \widetilde{\mathcal{A}}^t(s',\varphi) \vee (\Phi_t \wedge \Phi_{s',s})$
7:         $\widetilde{\mathcal{A}}^f(s',\varphi) := \widetilde{\mathcal{A}}^f(s',\varphi) \wedge (\Phi_f \vee \neg\Phi_{s',s}))$
```

In the case of SMT-based parameters encoding, the assumption function is represented symbolically. The *symbolic assumption function* $\widetilde{\mathcal{A}}$ is a function that assigns to each pair (s,φ) a pair of SMT formulae (Φ_t,Φ_f) such that for all $p \in \mathcal{P}$: $p \models \Phi_t$ iff $\mathcal{A}(p,s,\varphi) = \mathtt{tt}$ and $p \models \Phi_f$ iff $\mathcal{A}(p,s,\varphi) = \mathtt{ff}$. Each such function thus divides the set of all parameter values into three sets: those parameters that ensure the satisfaction of φ (Φ_t), those that ensure that φ is not satisfied (Φ_f), and finally those parameter values under which the satisfaction of φ is undefined ($\neg\Phi_t \wedge \neg\Phi_f$). Algorithm 6 shows how the symbolic assumption function is updated in the case of **EX**ψ formula. The algorithm deals with the two parts (true and false) of the symbolic assumption function separately by using the notation $(\widetilde{\mathcal{A}}^t(s,\varphi),\widetilde{\mathcal{A}}^f(s,\varphi)) = \widetilde{\mathcal{A}}(s,\varphi)$. To that end, the lines 2 and 3 are updated accordingly with respect to the explicit case. Lines 6-7 show the advantage of symbolic encoding. The formula $\Phi_{s',s}$ encodes the set of parameter values $\mathcal{P}(s',s)$ enabling the transition from s' to s. Finally, union and intersection operations on parameter sets are replaced with disjunction and conjuction, respectively. Instead of enumerating the parameter value sets the computation is reduced to manipulating with the formulae and performing satisfiability checks. All these steps are delegated to calls of an SMT solver.

It remains to note that node algorithms for other temporal operators are constructed in similar way with the exception of the operator **AU** where the computation can end up with the assumption function left undefined for some states as discussed in the case of distributed labelling algorithm in [91]. Therefore additional computation is needed in that case.

4.3.2 Interpretation of Parameter Synthesis Results

The employment of the labelling approach implies exhaustiveness in two aspects. First, the satisfying parameters are synthesised for all subformulae of the given formula. Second, the resulting parameter values reflecting all subformulae are obtained for all states of the Kripke structure. In particular, for a given PKS and a given formula, the method gives for every subformula a table where every state is associated with a set of satisfying parameter values.

It is important to discuss how the CMC results can be interpreted when applied for parameter synthesis of dINMs and cBNMs. In the former case, there is no limitation and the results in the form of a satisfying/unsatisfying parameter values sets for every state and formula are directly interpretable. However, in the case of cBNMs the approximation and abstraction steps affect the interpretation. For same reasons as in the LTL case, the effect of approximation of cBNMs by means of a PWMA system cannot be characterised. It is therefore necessary to evaluate the results at the level of PWMA system. The over-approximating abstraction affects CTL model checking results in the following way:

- For a formula in the universal fragment of CTL (ACTL), the abstraction causes satisfying parameter values synthesised by model checking to be under-approximated wrt the entire set of parameter values for which the formula is exactly valid in the PWMA system. Unsatisfying parameter values are over-approximated.

- For the existential fragment (ECTL), we obtain over-approximation of the exact set of satisfying parameter values. Unsatisfying parameter values are under-approximated in this case.

4.3.3 Performance Evaluation

On the theoretical side, the worst case time complexity of the overall CTL CMC algorithm for a given formula φ and the PKS $\mathcal{K}$ does not critically affect the complexity of the common labelling algorithm which is linear in the size of φ and the size of $\mathcal{K}$. However, the size of the PKS is significantly extended wrt the size of unparameterised KS. In particular, the added aspect is the size of the parameter space $\mathcal{P}$. Regarding its role in the most of the node algorithms, this is critical in the explicit assumption function case (e.g., the intersection at line 5 of Algorithm 5). In the case of cBNMs where the parameters are represented as continuous closed intervals of reals, the intersection of two such intervals can be done in constant time as well as their union. This cannot be done easily in case of dINMs where the parameters do not represent numbers but different settings of the regulatory logic. In the worst case, the parameter space size can affect the time complexity quadratically. In case of symbolic encoding, the enumeration of $\mathcal{P}$ is avoided but replaced with calls to the SMT solver. In the worst case, the overall computation depends on the number of SMT solver calls linearly.

On the practical side, we have performed several experiments on different parallel platforms to evaluate scalability of the algorithms. The prototype implementation has been provided for cBNMs, we have not yet explored the performance of CTL CMC on dINMs. The results we have obtained show that both multi-core and distributed implementation scale with the number of states and the size of the parameter space almost linearly. In general, despite the compactness of the parameter space in PKS representations, the challenging issue is the size of the parameter space that can be very large in practise. On a common hardware, considering the explicit assumptions, we were able to tackle cBNMs up-to three millions states (6-dimensional models) with up-to 6 unknown parameters in times less than 15 minutes (see [82]). Considering the symbolic assumptions, the performance has been reduced due to a large number of SMT solver calls (running in parallel). In particular, we have used Microsoft Z3 with common settings (see [56]). The profiling has shown the majority of computation is performed in SMT solver calls.

4.3.4 Publications Summary

Results overviewed in this section have been first published in [82] where the algorithm with the explicit assumption function is introduced and analysed. The symbolic assumption function and the respective algorithm is published in [56]. A set of biological case studies providing a deeper view into performance of CTL CMC using both assumption functions is provided in [139]. Please note that for the purpose of this thesis we have simplified and changed some of the notation.

The presented work is a joint work with my colleagues Nikola Beneš and Luboš Brim. The implementation of the work, the performance experiments and biological case studies have been realised with significant help of students Samuel Pastva and Martin Demko. My contribution has been in revising the problem of CMC in the context of cBNMs and CTL and guiding the work on several steps leading to a functional prototype of the method. I have also contributed in setting up the experiments and interpreting the achieved results.

4.4 Software Tools

Several prototype software tools implementing coloured model checking have been developed. They focus on efficient computation on common models and support scalable execution in terms of parallel implementation.

To address the LTL-based parameter synthesis for discrete abstractions of cBNMs (Section 4.2), the prototype tool PEPMC [38] has been developed. It is implemented to support multi-core parallelisation of Algorithm 1. The tool has been merged with BioDiVinE [35] (a previously developed tool supporting distributed LTL model checking of finite abstractions of cBNMs, implemented at the top of the DiVinE model checker [36]). The resulting NewBioDiVinE[1] supports the procedures of plain model checking and parameter synthesis wrt LTL specifications. The parameter synthesis module utilises multi-core parallelisation whereas the plain model checking module employs distributed state space exploration. The implementation is entirely done in C++.

CTL-based parameter synthesis supporting discrete abstractions of cBNMs has been first addressed by the prototype tool BioDiVinECTL[2] implemented in Java and supporting the distributed algorithm as described in Section 4.3. This version supports the interval-based encoding of parameter sets and is therefore limited to cBNMs with independent parameters. The tool has been further extended with hybrid CTL enhanced with action-labels operators and utilising the SMT-based encoding allowing to analyse models with interdependent parameters. The resulting stable release called Pithya[3] embedded with a web GUI[4] has been approved by the artefact evaluation procedure of the CAV conference [57]. The tool is publicly available as an online service[5].

We have also targeted the parameter synthesis procedure for dINMs wrt LTL. The prototype tool Parsybone[6] has been developed to provide an efficient implementation of the algorithms presented in Section 4.2 working directly on parameterised dINMs. Parsybone is entirely developed in C++. It has been embedded within a web GUI Esther [341]. The implementation has been further improved in collaboration with Adams Streck and the group of Heike Siebert at Free University Berlin. The resulting web service TREMPPI[7] is publicly available.

[1] https://github.com/sybila/NewBioDiVinE

[2] https://github.com/sybila/biodivineCTL

[3] https://github.com/sybila/pithya-core

[4] https://github.com/sybila/pithya-gui

[5] https://pithya.ics.muni.cz

[6] https://github.com/sybila/Parsybone

[7] http://tremppi.fi.muni.cz

4.5 Applications

The coloured model checking technique has been successfully applied to several biological case studies ranging from small models describing basic biological mechanisms (enzyme kinetics, cell-cycle control, λ-phage) to large-scale models providing analysis of several complex biological systems.

In [33] a cBNM model of ammonium transport system in *E. coli* is analysed using the LTL-based approach. In Chapter 7 we describe a couple of case studies performed with CMC techniques and published in [139] – synthesis of a synthetic TCP-degradation pathway in *E. coli* (Section 7.2) and analysis of the well-known bistable switch present in a cell cycle of mammalian cells (Section 7.3.3). Additionally, in [210] we have provided a preliminary analysis of signalling pathways of FGFR3 fibro-blast growfactor in rat cells that also employs CMC. This case study is overviewed in Section 7.4.

4.6 Discussion

To compare the approach presented in Sections 4.2 and 4.3 with the closest previously existing work, it is worth noting that the algorithm in [49] is sequential and relies on execution of two model checking procedures per each class of valuations. In the average case, the number of analysed parameter classes can be reduced by a suitable BDD representation of the parameter space. On the contrary, our approach is supposed to be based on a principally different idea. Enumerative LTL model checking procedure (reduced to detection of accepting cycles in the model-property product automaton) is employed directly on a graph that compactly represents the dynamics of all valuations. The computational effort can be significantly reduced in the average case due to small variations in subgraphs corresponding to different valuations. Moreover, we take the advantage of the choice of enumerative model checking and provide a parallel algorithm that accelerates the computation with the increasing number of CPUs.

The parameter synthesis approaches based on finite abstraction have been studied also in the field of hybrid systems. Owing to the non-linear character, these methods (including the abstraction procedure considered in this thesis) are usually applicable only to certain subclasses of biological models. The approach of [146] supports arbitrary non-linear ODE models. It is based on a hybrid approach built on sensitivity analysis. Approaches [218, 177, 368] inherently rely on reachability analysis in hybrid systems and are sufficiently applicable only to piece-wise affine or multiaffine systems. These algorithms technically rely on computing reachability for parameterised systems by means of polyhedral operations. This restricts the parameter synthesis procedure to safety properties only. Liveness properties, e.g. oscillations, for which any contradicting counterexample is an infinite path requiring monotonous time progress, can be considered only with a specific care [49]. This problem is also related to our method for cBNMs, in particular, the extent of over-approximation of the possibility of non-exiting a rectangle in finite time as discussed in Section 2.4.6 significantly affects the practical application of the rectangular abstraction to liveness analysis of non-linear systems. An improvement targeting this problem is proposed in [67] by lifting the rectangular abstraction to the framework of multi-affine hybrid automata. Besides a general increase in precision, this extension enables the universal handling of time-dependent properties. However, the price is the higher time complexity of the parameter synthesis algorithm and worse scalability.

Another approach to parameter synthesis for cBNMs is given in [269] where δ-complete decision procedures [186] for first-order logic (FOL) formulae are employed to overcome undecidability issues. In this setting, a FOL formula describes the states reachable with a finite number of steps. The parameter identification problem is reduced to finding a satisfying valuation of the parameters for this formula. The approach requires enumerating all the discrete paths of a particular length, which leads to performance degradation for large models.

4.7 Future Work

In the case of cBNMs the critical place for improvement is the approximation/abstraction procedure. First, an important aspect is to have a tool that will allow to characterise or predict the distance between a signal generated by a cBNM and a signal generated by its PWMA approximation. It will enable to take the approximation method closer to the concept of numerical error known from predictor-corrector algorithms used for numerical simulation of ODEs. Second, the abstraction makes a significant bottleneck of the entire procedure due to the large extent of over-approximation introduced by the rectangular discretisation. Some of the methods discussed in Section 2.4.6 partially solve the problem. However, they are not yet prepared to be employed for models with uncertain parameters.

The technique of CTL CMC has not been yet brought to the framework of dINMs. The reason for that is mainly that symbolic model checking works well with those kinds of models including parameterised versions. However, as shown in [17], it is definitely useful to bring hybrid extensions of CTL to the framework of parameterised dINMs. The reason for that is the necessity of abstract reasoning about various attractors in systems dynamics that might coexist in the system under certain settings of parameters. The case study in Section 7.3 makes an important example of such a system. To that purpose, it seems very relevant to adapt the techniques we have introduced in [58] (bifurcation analysis) and [31] (detection and counting of terminal strongly connected components in parameterised Kripke structures).

Chapter 5

Parameterised Uniformisation

To solve the parameter exploration problem for stochastic models described in Chapter 3, we have developed a method called *parameterised uniformisation*. The method is built upon the probabilistic model checking of quantitative properties of the continuous-time systems dynamics occurring in finite time horizon – the so-called transient analysis. The fundamental uniformisation technique is extended to the parameterised setting and works on properties belonging to an appropriate (bounded time) fragment of CSL with rewards.

5.1 Overview

The general workflow of parameter exploration of a sCRNM employing the parameterised uniformisation procedure is depicted in Figure 5.1. We assume that for a given sCRNM $\mathcal{M}$ parameterised by a parameterisation $\chi_{\mathcal{M}}$ within a perturbation space $\mathbf{P} \subseteq \mathcal{P}_{\chi_{\mathcal{M}}}$, the semantics of $\mathcal{M}$ is represented by a parameterised CTMC $\mathbf{C} = \{\mathcal{C}_\mu \mid \mu \in \mathbf{P}\}$. As declared in Chapter 3, the problem of *parameter exploration* is as follows: for each state $X \in \mathbb{X}$ compute the *landscape function* $\lambda_X^{\varphi,\mathbf{P}} : \mathbf{P} \to \mathbb{R}_{\geq 0}$ that for each parameter valuation $\mu \in \mathbf{P}$ returns the numerical value of the probability or the expected reward for the formula φ. It means that we consider "quantitative" formulae in the form $\varphi ::= \mathsf{P}_{=?}[\varphi] \mid \mathsf{R}_{=?}[\mathsf{C}^{\leq t}] \mid \mathsf{R}_{=?}[\mathsf{I}^{=t}]$, i.e., the topmost operator of the formula φ returns a quantitative result, as used, e.g., in PRISM [261]. In general, the formula φ can contain nested probabilistic and reward operators whose evaluations define discrete sets of states further used in the computation of the resulting numerical value. In consequence, the corresponding landscape function is not in general continuous but only piecewise continuous. An example of a simple landscape function describing a bimodal transient distribution in a given time varying with a parameter is shown in Figure 5.2. The distribution describes a snapshot of behaviour of the model described in Example 2.4. The bimodal distribution is typical for

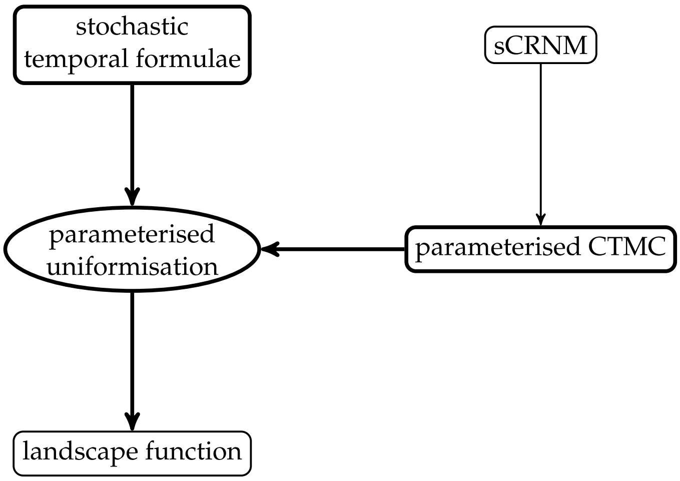


Figure 5.1: General workflow of parameter exploration by parameterised uniformisation.

reaction networks with positive feedback. The model is parameterised in such a way that k_1 denotes the parameter $\nu_{sRN}(\varrho_1)$. The corresponding formula CSL that exactly specifies the states distribution in time t is $\mathsf{P}_{=?}[\mathbf{F}^{\leq t}]$.

To solve the parameter exploration problem we extend global quantitative model checking techniques described in Section 2.6.3. The most crucial part of the problem is given by the fact that the perturbation space $\mathbf{P}$ is continuous and thus the family $\mathbf{C}$ is infinite. Especially, to compute the problem we cannot run the global quantitative model checking techniques for each CTMC $\mathcal{C}_\mu \in \mathbf{C}$ individually.

Our solution to the problem is based on a technique called *min-max approximation*. The key idea is to approximate the landscape function $\lambda_X^{\varphi,\mathbf{P}}$ using a lower bound $\overline{min}_X^{\varphi,\mathbf{P}} = min\{\lambda_X^{\varphi,\mathbf{P}}(\mu) \mid \mu \in \mathbf{P}\}$ and an upper bound $\overline{max}_X^{\varphi,\mathbf{P}} = max\{\lambda_X^{\varphi,\mathbf{P}}(\mu) \mid \mu \in \mathbf{P}\}$. Since the computation of the exact bounds is computationally infeasible, we further approximate these bounds, i.e., we compute approximations $min_X^{\varphi,\mathbf{P}}$ and $max_X^{\varphi,\mathbf{P}}$ such that $min_X^{\varphi,\mathbf{P}} \leq \overline{min}_X^{\varphi,\mathbf{P}}$ and $max_X^{\varphi,\mathbf{P}} \geq \overline{max}_X^{\varphi,\mathbf{P}}$. Although the proposed min-max approximation provides the lower and upper bounds of the landscape function, it introduces a numerical *error* with respect to parameter exploration, i.e., such approximation can be insufficient for the inspected perturbation space $\mathbf{P}$ and the given formula φ. Formally, the inaccuracy for a state X is given as the difference $max_X^{\varphi,\mathbf{P}} - min_X^{\varphi,\mathbf{P}}$. In Figure 5.3 there are shown two approximation variants of a landscape function from the example depicted in Figure 5.2. It can be seen that the approximation error can significantly affect the result.

An advantage of the min-max approximation is that it allows us to it-

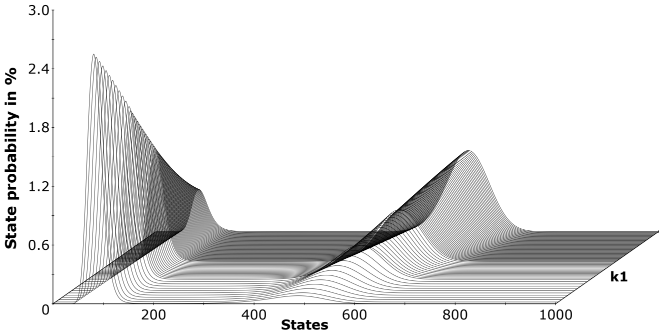


Figure 5.2: Example of a landscape function describing a snapshot of transient distribution of states in a parameterised CTMC varying with a particular perturbation of a single parameter $k1$.

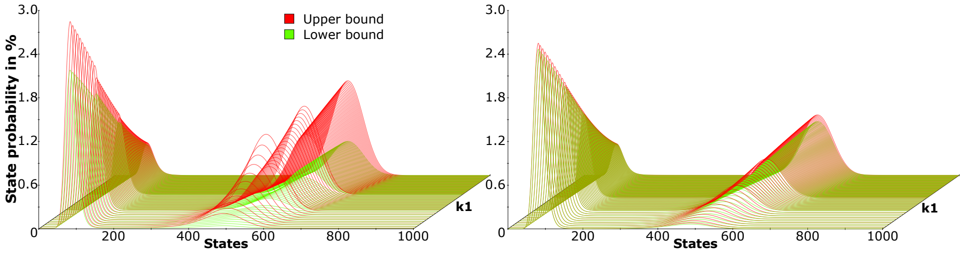


Figure 5.3: Min-max approximation of states distribution at a certain time of transient analysis of a parameterised CTMC varying with a particular perturbation of a single parameter $k1$. The two presented cases differ in absolute error bound: (left) 0.01, (right) 0.001.

eratively reduce the error to a required bound. The key idea is based on *iterative parameter space decomposition* where the perturbation space $\mathbf{P}$ is divided into subspaces that are processed independently. The result of such computation is an approximation of the lower bound $min_X^{\varphi,\mathbf{P}_i}$ and the upper bound $max_X^{\varphi,\mathbf{P}_i}$ for each subspace $\mathbf{P}_i$. Such decomposition provides more precise approximation of the landscape function $\lambda_X^{\varphi,\mathbf{P}}$ and enables to reach the required accuracy bound.

To effectively compute the min-max approximation for the given formula the method called *parameterised uniformization* is developed. In general, it allows to efficiently approximate the transient probabilities for the parameterised CTMCs. The key idea is a modification of *standard uniformization* [201] so that an approximation of the minimal and maximal transient probability with respect to the family $\mathbf{C}$ can be computed.

5.2 Method Principles

5.2.1 Local Transient Analysis

The parameterised uniformisation is a modification of the standard uniformisation [260], a widely used technique for transient analysis of CTMCs. For the given parameterised CTMCs $\mathbf{C}$, the initial state $X_0 \in \mathbb{X}$ and time $t \in \mathbb{R}_{\geq 0}$, the parameterised uniformisation returns vectors $\pi_{\top}^{\mathbf{C},X_0,t}$ and $\pi_{\bot}^{\mathbf{C},X_0,t}$, such that for each state $X' \in \mathbb{X}$ the following holds:

$$\pi_{\top}^{\mathbf{C},X_0,t}(X') \geq max\{\pi^{\mathcal{C}_\mu,X_0,t}(X') \mid \mathcal{C}_\mu \in \mathbf{C}\}$$
$$\wedge\, \pi_{\bot}^{\mathbf{C},X_0,t}(X') \leq min\{\pi^{\mathcal{C}_\mu,X_0,t}(X') \mid \mathcal{C}_\mu \in \mathbf{C}\}$$

where $\pi^{\mathcal{C}_\mu,X_0,t}$ denotes the transient state distribution of CTMC $\mathcal{C}_\mu$ in the time t. The key idea is to compute for each state X' the local maximum (resp. minimum) of $\pi^{\mathcal{C}_\mu,X,t}(X')$ over all $\mathcal{C}_\mu \in \mathbf{C}$ with respect to the current computation step of $\pi^{\mathcal{C}_\mu,X,t}$. It means that only the maximal (resp. minimal) values of predecessors of X' from the preceding step are considered. To obtain the local maximum (resp. minimum) of $\pi^{\mathcal{C}_\mu,X,t}(X')$ we define a function returning for the parameter valuation $\mu \in \mathbf{P}$ the difference of probability mass inflow and outflow to/from state *s*.

Remark 5.1 *In [89] we have shown that if all reactions are described by mass action kinetics the function is monotonous with respect to any single perturbed stochastic rate parameter included in the parameterisation $\chi_{\mathcal{M}}$. This allows us to efficiently identify $\mu \in \mathbf{P}$ that maximises (minimises) the value $\pi^{\mathcal{C}_\mu,X,t}(X')$ and thus to obtain the vectors $\pi_{\top}^{\mathbf{C},X_0,t}$ and $\pi_{\bot}^{\mathbf{C},X_0,t}$.*

5.2.2 Global Transient Analysis

The aforementioned parameterised uniformisation can be employed also for backward transient analysis that is used for the global model checking procedure discussed in Section 2.6.3. For the given set of states $\mathbb{A} \subseteq \mathbb{X}$ and time t we can efficiently compute the vectors $\rotatebox{180}{$\pi$}_{\top}^{\mathbf{C},\mathbb{A},t}$ and $\rotatebox{180}{$\pi$}_{\bot}^{\mathbf{C},\mathbb{A},t}$ such that for each state $X \in \mathbb{X}$ the following holds:

$$\rotatebox{180}{π}_{\top}^{\mathbf{C},\mathbb{A},t}(X) \geq max\{\rotatebox{180}{π}^{\mathcal{C}_\mu,\mathbb{A},t}(X) \mid \mathcal{C}_\mu \in \mathbf{C}\}$$
$$\wedge\, \rotatebox{180}{π}_{\bot}^{\mathbf{C},\mathbb{A},t}(X) \leq min\{\rotatebox{180}{π}^{\mathcal{C}_\mu,\mathbb{A},t}(X) \mid \mathcal{C}_\mu \in \mathbf{C}\}$$

where $\rotatebox{180}{$\pi$}^{\mathcal{C}_\mu,\mathbb{A},t}(X)$ denotes the probability that the set $\mathbb{A}$ is reached from X at time *t* in the CTMC $\mathcal{C}_\mu$.

5.2.3 Model Checking

The min-max approximation employs the results of the parameterised uniformisation (i.e., the vectors $\pi_{\top}^{\mathbf{C},X_0,t}$, $\pi_{\bot}^{\mathbf{C},X_0,t}$ $\rotatebox{180}{$\pi$}_{\top}^{\mathbf{C},\mathbb{A},t}$ and $\rotatebox{180}{$\pi$}_{\bot}^{\mathbf{C},\mathbb{A},t}$) to approximate the *largest set of states satisfying* φ, and the *smallest set of states satisfying* φ with respect to the perturbation space $\mathbf{P}$. It computes the approximation $Sat^{\top}_{\mathbf{C}}(\varphi)$ and $Sat^{\bot}_{\mathbf{C}}(\varphi)$ such that

$$Sat^{\top}_{\mathbf{C}}(\varphi) \supseteq \bigcup_{\mathcal{C}_\mu \in \mathbf{C}} Sat_{\mathcal{C}_\mu}(\varphi) \;\wedge\; Sat^{\bot}_{\mathbf{C}}(\varphi) \subseteq \bigcap_{\mathcal{C}_\mu \in \mathbf{C}} Sat_{\mathcal{C}_\mu}(\varphi),$$

where $X \in Sat_{\mathcal{C}_\mu}(\varphi)$ iff X satisfies the formula φ in CTMC $\mathcal{C}_\mu$. To obtain such approximations we extended the standard satisfaction relation for CSL logic [89]. For more details about the min-max approximation see [89].

5.2.4 Numerical Errors of Parameterised Uniformisation

It is worth noting that the parameterised uniformization for the parameterised CTMC $\mathbf{C}$ in general does not correspond to standard uniformization for any individual CTMC $\mathcal{C}_\mu \in \mathbf{C}$. The reason is that we consider a behaviour of a parameterised CTMC that has no equivalent counterpart in any particular $\mathcal{C}_\mu$. First, the parameters (minimising/maximising the inspected value) are determined locally and thus independently for each state. Second, the parameters are determined independently for each computational step.

In a given state X, the inaccuracy of the min-max approximation related to the computation of parametrised uniformization, called *unification error*, denoted as $\mathsf{uniErr}^{\varphi,\mathbf{P}}_X$, and defined:

$$\mathsf{uniErr}^{\varphi,\mathbf{P}}_X \overset{df}{=} (max^{\varphi,\mathbf{P}}_X - \overline{max}^{\varphi,\mathbf{P}}_X) + (\overline{min}^{\varphi,\mathbf{P}}_X - min^{\varphi,\mathbf{P}}_X)$$

Apart from the unification error the min-max approximation introduces an inaccuracy related to approximation of the landscape function $\lambda^{\varphi,\mathbf{P}}_X$, called *approximation error*, denoted as $\mathsf{appErr}^{\varphi,\mathbf{P}}_X$, and defined:

$$\mathsf{appErr}^{\varphi,\mathbf{P}}_X \overset{df}{=} \overline{max}^{\varphi,\mathbf{P}}_X - \overline{min}^{\varphi,\mathbf{P}}_X$$

Finally, the *overall error* of the min-max approximation, denoted as $\mathsf{Err}^{\varphi,\mathbf{P}}_X$, is defined as a sum of both errors:

$$\mathsf{Err}^{\varphi,\mathbf{P}}_X \overset{df}{=} \mathsf{uniErr}^{\varphi,\mathbf{P}}_X + \mathsf{appErr}^{\varphi,\mathbf{P}}_X = max^{\varphi,\mathbf{P}}_X - min^{\varphi,\mathbf{P}}_X$$

It is important to note that we are not able to effectively distinguish the proportion of the approximation error and the unification error nor to reduce the unification error as such. Therefore, we have developed a method that provides parameter space decomposition. That way the overall error of the min-max approximation can be effectively reduced to a user specified *absolute error bound*, denoted as ERR.

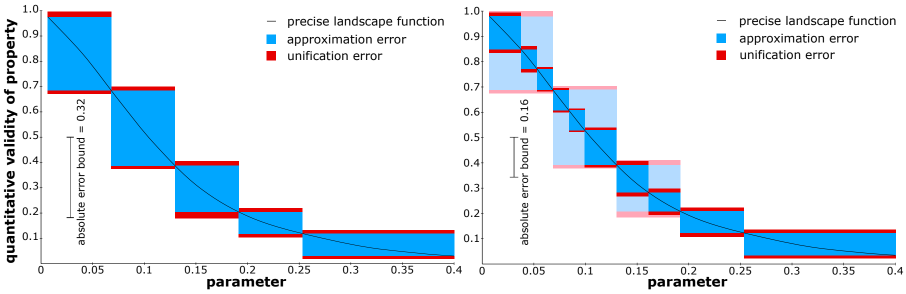


Figure 5.4: Illustration of the min-max approximation computation of the landscape function $\lambda_X^{\varphi,\mathbf{P}}$ for an initial state X_0, property φ and the perturbation space $\mathbf{P}$ assigning a single parameter a value from the range $[0, 0.4]$. Left graph shows the decomposition of $\mathbf{P}$ into 5 subspaces for absolute error bound ERR = 0.32. Right graph shows a more refined decomposition for ERR = 0.16 resulting in 10 subspaces. This decomposition reduces both types of errors in each refined subspaces. The exact shape of $\lambda_X^{\varphi,\mathbf{P}}$ is visualised as the black curve.

5.2.5 Parameter Space Decomposition

In order to ensure that the min-max approximation meets the given absolute error bound ERR, we iteratively decompose the perturbation space $\mathcal{P}$ into finitely many subspaces such that $\mathbf{P} = \mathbf{P}_1 \cup \ldots \cup \mathbf{P}_n$ and each partial result satisfies the overall error bound, formally written:

$$\forall X \in \mathbb{X} : max_X^{\varphi,\mathbf{P}_i} - min_X^{\varphi,\mathbf{P}_i} \leq \textsc{Err}$$

In consequence, the overall error for each state $X \in \mathbb{X}$ satisfies the following equation:

$$\mathsf{Err}_X^{\varphi,\mathbf{P}} = \sum_{i=1}^{n} \frac{|\mathbf{P}_i|}{|\mathbf{P}|} \left(max_X^{\varphi,\mathbf{P}_i} - min_X^{\varphi,\mathbf{P}_i} \right) \leq \sum_{i=1}^{n} \frac{|\mathbf{P}_i|}{|\mathbf{P}|} \textsc{Err} = \textsc{Err}$$

Figure 5.4 illustrates an example of a parameter space decomposition and demonstrates convergence of $\mathsf{Err}_X^{\varphi,\mathbf{P}_i}$ to 0 provided that the function $\lambda_X^{\varphi,\mathbf{P}_i}$ is continuous.

An optimal decomposition with respect to the perturbation space $\mathbf{P}$ has minimal number of subspaces m such that $\mathbf{P} = \mathbf{P}_1 \cup \ldots \cup \mathbf{P}_m$ and for each subspace $\mathbf{P}_j$ where $1 \leq j \leq m$ holds that $\mathsf{Err}_X^{\varphi,\mathbf{P}_j} \leq \textsc{Err}$. The existence of such decomposition is guaranteed only if the landscape function $\lambda_X^{\varphi,\mathbf{P}}$ is continuous. If the landscape function is continuous there can exist more than one minimal decomposition. To search for an optimal decomposition we have considered and implemented several heuristics allowing to iteratively compute a decomposition satisfying the following: (1) it ensures the required error bound whenever $\lambda_X^{\varphi,\mathbf{P}}$ is continuous, (2) it guarantees the refinement termination in the situation where $\lambda_X^{\varphi,\mathbf{P}}$ is not continuous and the discontinuity causes that ERR can not be achieved. To ensure the termination, an additional parameter has to be introduced as a lower bound on the subspace size. Hence this parameter provides a supplementary termination criterion.

5.3 Performance Evaluation

On the theoretical side, the asymptotic time complexity of standard uniformization is not increased by the parameterised extension. An important fact is that we have restricted the admissible specification to bounded-time CSL formulae only. In particular, there always exists a maximal bound on the number of uniformisation iterations that is linear in the uniformisation rate multiplied by the time bound occurring in the formula [176]. Additionally, every single iteration is quadratic in $|\mathbb{X}|$ (the size of the state space of the parameterised CTMC). It is worth noting that the number of states is exponential in the number of species. In consequence, storing of the state space or the generating matrix can have high demands on the computer memory.

On the practical side, we have implemented the prototype of parameterised uniformisation at the top of numerical algorithms in PRISM 4.0 [261]. Evaluation on several models has shown that the computation becomes demanding with increasing the requirements on the precision. E.g., in case of simple models (two-dimensional state space, up to eight reactions, two perturbed parameters) the computations took several hours on a single common hardware computer.

The conducted experiments have shown that for a given sCRNM, the critical complexity factor is the number of uniformisation iterations that depends on the time scale of the reaction rates. If the rates in individual states significantly differ in the order, the fast rates dictate the number of the iterations that must be performed within the given time horizon. Since the actual rate for a given state depends on the value (number of molecules) kept in the state, the fastest rates appear in states representing high numbers of molecules that form reactants of reactions with fast dynamics. Regarding the states space, when considering models where stochasticity is crucial (e.g., protein molecules or RNA/DNA) the numbers of molecules are typically small and models focusing on a certain phenomena requiring only a limited number of species are tractable. However, complex models including species of different scales (e.g., proteins interacting with metabolites) and acting on different time scales (e.g., transcription mixed with enzyme kinetics) typically require further elaborations in terms of model reduction and simplification to allow tractability of parameterised uniformisation.

5.4 Applications

The method has been successfully applied to parameter exploration of signalling pathways in prokaryotic bacteria [359] (overviewed in Section 7.5) and cell cycle G_1/S_1-phase transition genetic switch in mammalian cells [89, 359] (overviewed in Section 7.3.4). Both studies have shown new insights on the dynamics of the cell mechanisms.

In the case of the bacterial signalling pathways, it has been revealed that the synthetic two-component signalling pathway analysed in [338] is more robust wrt to the internal noise than the natural pathway only under a particular settings of the input signal level.

The analysis of the G_1/S genetic switch has demonstrated the importance of stochastic modelling of genetic regulatory mechanisms for which the intrinsic noise in protein/DNA molecules dynamics significantly affects the behaviour of the system and hence the decisions in cells. The analysis under parameter uncertainty gives new understanding on how the parameters affect the molecule distributions and their stabilisation. Such detailed analysis is not possible with commonly used techniques limited to deterministic models [345].

5.5 Publications Summary

The publication [89] contains the first results introducing the parameterised uniformisation, the parameter space decomposition and the related heuristics. Evaluation is done on several smaller but biologically relevant stochastic models. My contribution is in first formulation of the problem, setting the work in the context of related research, collaboration on development of the technique with Milan Češka and preparation of the biological models for the evaluation including formulation of the individual analysis tasks and interpretation of the results. Detailed technical work and implementation has been realised by Milan Češka and Sven Dražan under my co-supervision.

The publication [359] is partially also devoted to the technique. It contains extended results that enable analysis of models with more complex dynamics than is covered by sCRNMs. Moreover, a simpler reformulated description of the technique is presented in the supporting material available with the paper. The main focus is robustness analysis (discussed in Chapter 6).

5.6 Discussion

Although our stochastic dynamics model is restricted to reaction networks, it is worth mentioning that there exist circumstances under which more complicated kinetics or even regulatory influences can be modelled stochastically in terms of CTMCs [322, 159]. In our framework, we can consider generalisation of sCRNMs to more complex kinetic models often used with cBNMs. In such cases, the corresponding stochastic kinetic function does not have to be in general monotonous in the parameter for all states of the CTMC. E.g., such phenomenon appears in Michaelis-Menten functions of enzyme kinetics [322] or in Hill kinetics used to described regulatory influences [159] (see Definition 2.14). In such cases, there are parameters for which the corresponding kinetic function is not monotonous, e.g., Michaelis constant in enzyme kinetics or steepness coefficient in Hill functions). This makes the computation of local extremes more intricate, though still tractable. To that end, the kinetic function needs to be decomposed to partitions of the parameter values on which it is monotonous as shown in [359] (and also used in the case study in Section 7.5).

It is worth noting that the landscape function can be alternatively computed by using standard uniformization to obtain precise values in grid points which could be afterwards interpolated linearly or polynomially. Using adaptive grid refinement such an approach can also provide an arbitrary degree of precision with computation complexity of the same asymptotic class as our method. However, the grid-based approach cannot provide the strict minimal and maximal upper bounds. On the contrary, our min-max approximation guarantees upper and lower estimates without neglecting any singularities caused by possible discontinuities in the landscape function.

Alternative approaches to analysis of sCRNMs under parameter uncertainty are thoroughly based on statistical methods performed on parameterised CTMCs [73] or on correct approximations of stochastic dynamics (mean-field approximation, general moment closures or linear noise approximation). In cases where the landscape function is supposed to be smooth and the properties analysed do not require global transient analysis, those techniques can perform practically very well and can be embarrassingly parallelised. However, in general cases only statistical guarantees are given with the results. On the contrary, our technique gives exact guarantees and can basically cope with arbitrary shapes of the landscape function at the price of higher computational demands and non-trivial parallelisation.

To conclude the discussion, we believe the technique of parameterised uniformisation provides a significant fundamental result that fills the gap in global bounded-time transient analysis of models represented by means of parameter-uncertain continuous-time Markov chains.

In the follow-up work the method has been further extended by Milan Češka and the research group of prof. Marta Kwiatkowska. The general idea of that work has been to focus on the critical parameter decomposition step by combining the min-max approximation with sampling of the evaluation function allowing to avoid unnecessary computation. This idea has been presented in [234, 86] including some other improvements: an extension to support rate functions with multi-affine dependency on the parameters, convergence analysis of the approximation error, and definition of the problems set directly on the top of the parameterised uniformisation – threshold synthesis and max synthesis. The threshold synthesis computes for a given CSL formula with a probability threshold the parameter valuations which meet the threshold. The max synthesis computes parameter valuations maximising the satisfaction probability of a given formula. A prototype parallel implementation on GPU has been implemented [103] and embedded in the GUI of the PRISM 4.0 tool.

5.7 Future Work

For future work, there is a place namely in optimising the method for particular classes of CSL properties (e.g., to identify commonly used patterns of CSL formulae with rewards relevant for sCRNMs) and models (e.g., influence networks based on enzyme or Hill kinetics). Additionally, an open non-trivial problem is to transfer the technique to the domain of hybrid approaches that allow to mix discrete and continuous variables [262] and combine the technique with moment-based methods. It is apparent that techniques based on moment closures [13, 273, 180] are capable of targeting several classes of models with a precision sufficient for biological studies. Important directions of works that fit into the overall workflow consider approximate reductions of models by means of variables lumping [204].

An important point for future work is to fine-tune the method implementation in terms of clarifying restrictions to CSL specifications and the CTMC and deploy it to the next stable version of the PRISM model checker [261].

Chapter 6

Robustness Analysis

In Chapter 3 we have stated the problem of property-based robustness analysis. The open problem for cBNMs was to develop a method capable of quantitative characterisation of expressive formulae encoded in STL*. To that end, we have defined quantitative semantics of STL* interpreted over approximate signals generated by a given cBNM model.

Another open problem was to bring the concept of property-based robustness analysis to stochastic models of biological systems. To that end, we have addressed sCRNMs analysed wrt CSL formulae. Our novel concept of robustness is based on the parameterisation uniformisation method described in Chapter 5.

In this chapter, we summarise our results regarding the development, implementation, and evaluation of both methods for robustness analysis.

6.1 Overview

The general workflow of robustness analysis based on a temporal specification is depicted in Figure 6.1. The input is a given temporal specification expressed in a suitable logic (in our case we will consider STL* or CSL), a model of a biological network (we consider any type of cBNMs or a sCRNM), and a parameter perturbation that specifies the range of parameter uncertainty for which the robustness has to be analysed. The robustness analysis algorithm depends on what kind of model we use. For continuous models (cBNMs) we target the robustness by means of a measure that characterises how a given signal is far from violating a formula and we compute a so-called *deterministic robustness*. For stochastic models (represented in terms of sCRNMs) we address the robustness with respect to the probability of a formula to be satisfied and therefore we speak about a *probabilistic robustness*.

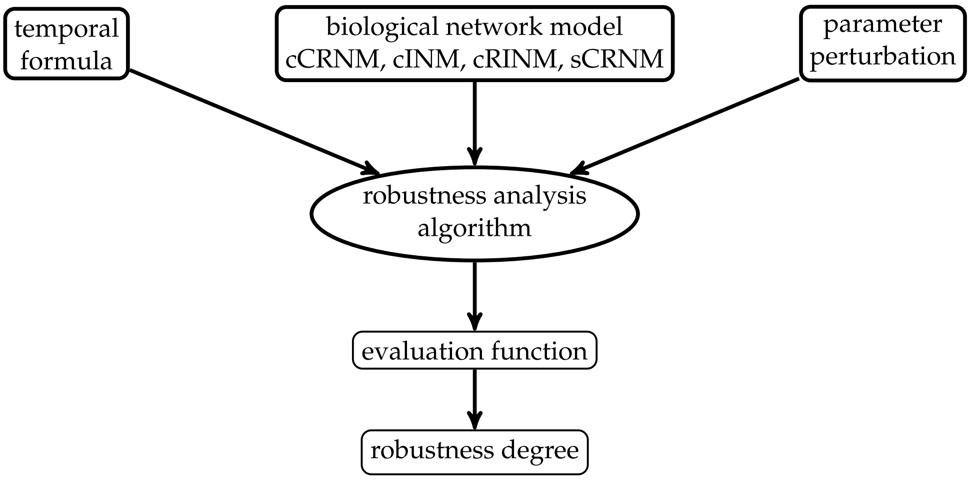

temporal formula
biological network model cCRNM, cINM, cRINM, sCRNM
parameter perturbation
robustness analysis algorithm
evaluation function
robustness degree

Figure 6.1: General workflow of robustness analysis realised with respect to a temporal specification.

6.2 Deterministic Robustness for STL*

For any kind of continuous-time deterministic models including cCRNM, cINM and cRINM (all generating continuous-time signals) we address space robustness analysis with respect to formulae of the value-freezing temporal logic STL* presented in Section 2.5.4. Formally, we target the problem stated in Definition 3.3. In particular, the first goal is to define a robustness measure for approximate signals generated by a model with respect to a given STL* formula and to support it with a suitable algorithm. Second goal is to approximately compute the (global) robustness degree by employing the (local) robustness measure defined for signals.

To solve the first problem, we have extended the continuous and discrete measure defined for MTL by Fainekos et al. [168] to establish the quantitative semantics of STL* formulae over signals. Space robustness of a given signal with respect to an STL* formula delineates the robust neighbourhood of the signal (the maximal "tube" around the signal where the formula is satisfied). The robustness measure is defined inductively wrt the formula structure and is based on a distance metrics employed on the signal domain extended with (multiple) dimensions representing the time points of value freezing. On the practical side, the theoretical framework is computationally supported with an algorithm based on solving the optimisation problem provided that the logic is restricted to *linear predicates* and the signals are considered approximate (discretely sampled by means of finite timed traces that can also be considered as finite piece-wise constant signals with finite number of changes of signal values).

The second problem (computation of the robustness degree) is in our case realised approximately with a simple sampling method. Owing to the fact there exist more sophisticated methods that can be directly used with our framework (e.g., [145]), we have primarily focused on the first problem.

6.2.1 Method Principles

Quantitative semantics of a formula φ interpreted on a signal $\mathfrak{x}$ is given for each time point $t \in T$ and a frozen time vector t^*, denoted $\rho(\varphi, \mathfrak{x}, t, t^*)$. Further we denote $\rho(\varphi, \mathfrak{x}) = \rho(\varphi, \mathfrak{x}, 0, 0)$ the robustness of the entire signal $\mathfrak{x}$ starting at the point $\mathfrak{x}(0)$ measured with respect to formula φ. The value $\rho(\varphi, \mathfrak{x})$ under-approximates the distance of $\mathfrak{x}$ from the set of signals where φ has different truth value. The evaluation function $D_\varphi^{\mathcal{M}}$ is defined directly as the quantitative semantics of φ, in particular, we set $D_\varphi^{\mathcal{M}}(\mu) \overset{df}{=} \rho(\varphi, \mathfrak{x}_\mu)$.

Quantitative Semantics of STL*

To express the notion of signal robustness formally, we first need to define certain basic concepts adopted from [168] (where $\mathfrak{S}$ is a set of signals):

- Distance of signals is given by their maximum point-wise distance: $d(\mathfrak{x}, \mathfrak{x}') = \max_{t \in \mathbb{R}_{\geq 0}} d(\mathfrak{x}(t), \mathfrak{x}'(t))$
- *Set distance* is given by minimum distance to the set: $\mathbf{dist}(\mathfrak{x}, \mathfrak{S}) = \min\{d(\mathfrak{x}, \mathfrak{x}') \mid \mathfrak{x}' \in \mathfrak{S}\}$
- *Set depth* is given by set distance to the complement: $\mathbf{depth}(\mathfrak{x}, \mathfrak{S}) = \mathbf{dist}\left(\mathfrak{x}, \overline{\mathfrak{S}}\right)$
- *Signed distance* is given: $\mathbf{Dist}(\mathfrak{x}, \mathfrak{S}) = \begin{cases} -\,\mathbf{dist}(\mathfrak{x}, \mathfrak{S}) & \mathfrak{x} \notin \mathfrak{S} \\ \mathbf{depth}(\mathfrak{x}, \mathfrak{S}) & \mathfrak{x} \in \mathfrak{S} \end{cases}$

The value $\rho(\varphi, \mathfrak{x})$ under-approximates the signed distance of $\mathfrak{x}$ from the set of all signals satisfying φ, denoted as $\mathcal{L}(\varphi)$, i.e. $|\rho(\varphi, \mathfrak{x})| \leq |\,\mathbf{Dist}(\mathfrak{x}, \mathcal{L}(\varphi))|$ holds while their signs are identical. The absolute value of $\rho(\varphi, \mathfrak{x})$ thus delineates an equidistant tube where all signals satisfy φ if and only if $\mathfrak{x}$ does – the *robust neighbourhood* of $\mathfrak{x}$ (see Figure 6.2).

The quantitative semantics of STL* is defined inductively with respect to formula structure. Boolean functions $\wedge$ and $\vee$ are quantitatively enumerated by real functions $\min$ and $\max$, respectively. Quantifiers in the semantics of operator $\mathbf{U}$ can then be expressed by infinite disjunction or conjunction. The quantitative semantics of a predicate $\mathfrak{p}$ is defined as $\mathbf{Dist}(\mathfrak{x}, \mathcal{L}_{t,t^*}(\mathfrak{p}))$. The quantitative semantics of the freeze operator is defined in the following way:

$$\mathcal{L}_{t,t^*}(*_i\,\varphi) = \{\mathfrak{x} \mid (\mathfrak{x}, t, t^*) \models *_i\,\varphi\} = \{\mathfrak{x} \mid (\mathfrak{x}, t, t^*[i \leftarrow t]) \models \varphi\} = \mathcal{L}_{t,t^*[i \leftarrow t]}(\varphi)$$

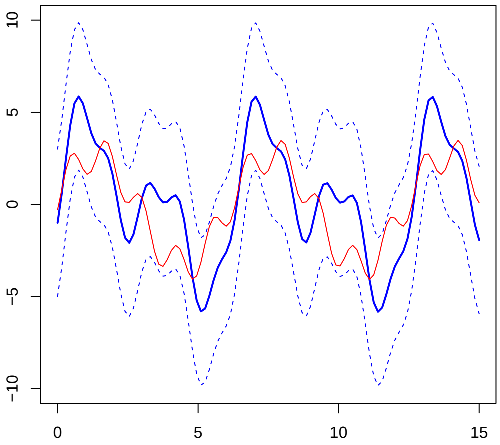


Figure 6.2: Signal $\mathfrak{x}$ (blue, thick) and borders of its robust neighbourhood (blue, dashed) with an example of a signal (red) contained in the robust neighbourhood (adapted from [168]).

In consequence, the robustness of a formula with a freeze operator can be defined in the following manner:

$$\rho(*_i\, \varphi, \mathfrak{x}, t, t*) = \rho(\varphi, \mathfrak{x}, t, t^*[i \leftarrow t])$$

$$\begin{array}{ll}
\rho(true, \mathfrak{x}, t, t^*) & = +\infty \\
\rho(\neg\varphi, \mathfrak{x}, t, t^*) & = -\rho(\varphi, \mathfrak{x}, t, t*) \\
\rho(\varphi_1 \vee \varphi_2, \mathfrak{x}, t, t^*) & = \max\left(\rho(\varphi_1, \mathfrak{x}, t, t^*), \rho(\varphi_2, \mathfrak{x}, t, t^*)\right) \\
\rho(\varphi_1 \mathbf{U}_I \varphi_2, \mathfrak{x}, t, t^*) & = \max\limits_{t' \in t \oplus I} \min\left(\rho(\varphi_2, \mathfrak{x}, t', t^*), \min\limits_{t'' \in [t, t']} \rho(\varphi_1, \mathfrak{x}, t'', t^*)\right) \\
\rho(*_i\, \varphi, \mathfrak{x}, t, t^*) & = \rho(\varphi, \mathfrak{x}, t, t^*[i \leftarrow t])
\end{array}$$

Figure 6.3: Robustness of STL* logical connectives.

Predicates are restricted to be linear. This allows solving the robustness of predicates by convex programming. For predicate $\mathfrak{p}$ with coefficients a_{ij}, b, the problem of finding $\mathbf{Dist}(\mathfrak{x}, \mathcal{L}_{t,t^*}(\mathfrak{p}))$ can be reduced to optimisation of $f(\mathbf{d}) = \max_i \sum_j d_{ij}^2$ (where $i \in \mathcal{I}$ and $j \in \{1, \ldots, n\}$) under the constraint $\sum_i \sum_j a_{ij} d_{ij} + \varepsilon = 0$ for some positive ε. This is a non-trivial problem, since f is not differentiable at point $\mathbf{d}$ where $f(\mathbf{d}) = \sum_j d_{kj}^2 = \sum_j d_{lj}^2$ for some $k \neq l$. A, generalised method of Lagrange multipliers from [118] is used to solve this problem, resulting in the following definition of the robustness ρ:

Definition 6.1 *Let $\mathfrak{p}$ be a predicate with coefficients a_{ij}, b. Then*

$$\rho(\mathfrak{p}, \mathfrak{x}, t, t^*) = \frac{\sum_j a_{0j}\mathfrak{x}_j(t) + \sum_i \sum_j a_{ij}\mathfrak{x}_j\,(t_i^*) + b}{\sum_i \sqrt{\sum_j a_{ij}^2}}$$

for arbitrary $\mathfrak{x}$, t, t^*, i ranging over $\mathcal{I}$, j ranging over $\{1, \ldots, n\}$. The notation $\mathfrak{x}$ denotes the jth component of the vector signal.

The numerator corresponds to the left-hand side value of the predicate.

The algorithm for computing the robustness for a given bounded time signal $\mathfrak{x}(t)$ expects the signal to be sampled into a discrete-time series of values. Numerical simulations have exactly such shape (they can be considered as piece-wise constant signals where the value changes only finitely many times). The algorithm traverses the formula starting from predicates and following the inductive definition of robustness. For every subformula, the signal is processed by analysing all the sampled points.

Robustness Degree

The quantitative semantics provides a robustness measure of a signal wrt a given formula. In particular, it can be understood as a "local" robustness of systems behaviour. On the other end, the robustness degree is given by the integral of the evaluation function over the given perturbation space. It provides a "global" measure of robustness that can be used to compare perturbation spaces or even different models.

In our setting, the integral is approximated by summing finite samples of the evaluation function for signals generated for parameter valuations chosen uniformly from the perturbation space. Such a method gives no guarantees whether the result is under- or over-approximation of the actual robustness degree value. Methods reflecting sensitivity of the evaluation function with respect to changes in parameter values [150, 151] can be employed to improve the precision of robustness degree approximation.

6.2.2 Performance Evaluation

The complexity of robustness computation for STL* is in $\mathcal{O}\left(|\varphi| \cdot n^{2|\mathcal{I}|}\right)$ where n is the size of input timed state sequence (the number of points where the piece-wise constant signal is changed). Space complexity can be bounded by the same function.

The parameter most adversely affecting the algorithm complexity is the number of value-freezing points $|\mathcal{I}|$ required by the formula. In general, $|\mathcal{I}|$

reflects the number of freezing operators appearing in the used formula. In particular, the critical task is to design the formula in an optimal way in order to avoid causing the computation to become unnecessarily hard. In [90] we have proposed several ways of rewriting a formula in such a way that it preserves the quantitative semantics while minimising the number of different freeze operators.

6.2.3 Applications

The quantitative semantics has been applied to robustness analysis of a well-known population model (SIR [236]) predicting an outbreak of an infectious disease in the population. Additionally, we have used the robustness analysis to investigate the well-known model of Lotka-Volterra [272] simulating the predator-prey population behaviour. Results of both applications are described in [90].

To conclude, the method has been applied to known models from systems biology and population dynamics. The purpose of the studies was mainly to demonstrate its unique features and advantages. Applications to currently challenging problems in systems biology are the goal of our future work on this topic. Currently there is an ongoing work in transferring the method to existing software tools for offline monitoring of dynamical systems.

6.2.4 Publications Summary

In paper [143], we have introduced the logic STL* as a novel formalism for expressing advanced characteristics of continuous-time signals. The logic is defined with the Boolean semantics. It is supported with a monitoring algorithm technically based on polyhedra operations. We have implemented a prototype in Matlab employing the MPT toolbox [257]. The work has been done in collaboration with my master student Petr Dluhoš whom I have guided in his master's thesis. He has helped with the formalisation of the semantics and he has provided the prototype implementation.

The paper [86] is an extended version of the previous paper. It contains a refined motivation for STL*, detailed definition of the Boolean semantics and several case studies that support the practicability of the method.

Finally, in paper [90] the quantitative semantics of STL* is formally defined and the algorithm for its computation is given. Several case studies are included to demonstrate the practicability of the method. Most of the work has been done with collaboration with my master's student Tomáš Vejpustek with whom we have jointly formulated the formal fundamentals of the quantitative semantics and the algorithm. The prototype implementation has been realised by Tomáš Vejpustek and Jan Papoušek as a part of a student research project I have managed.

6.2.5 Discussion

It is worth noting that the robustness computation algorithm performs more efficiently than the monitoring algorithm with Boolean semantics [86]. This is due to the fact that operations on polygons are completely avoided. However, the quantitative semantics only approximates the exact true/false answer of the monitoring procedure. However, the exponential complexity wrt the number of freezing operators persists.

Our method of robustness analysis for STL* is in general well-accepted in the community of monitoring and run-time verification and makes an important theoretical base for further research. This is supported by the fact that the method is cited (and explained) in several reviews on run-time verification methods [44, 296] and in relevant doctoral theses [153, 267].

The method has inspired several follow-up works that target the necessity of specification of temporal phenomena that goes beyond expressiveness of the plain STL. The semantics as defined in STL* is specific in including the "memory" of value-freezing time points into the satisfaction relation. Such an approach has been also used in the work on timed regular expressions [355] that provide an alternative approach to capturing (and analysing) complex behaviour of signals and streams in general. The concept of timed regular expressions has been combined with STL including efficient qualitative monitoring algorithms that are implemented in the AMT tool [297]. In [1] the authors target the limitations of STL* when applied to the analysis of continuously occurring peaks in a signal coming to cardioverter defibrillators. Despite the logic is capable of a very compact expressing of the specification of peaks in a signal, it is not that suitable for cases when there is a need to distinguish among many different shapes of peaks all occurring in a single signal in a certain pattern. To that end, the authors adapt the framework of Quantitative Regular Expressions [11]. Robustness evaluation of regular expressions over signals is addressed in [26].

6.2.6 Future Work

There are several directions for future work. On the practical side, the scalability aspects of robustness analysis with respect to the formula complexity need to be improved. Especially, the inevitable exponential computation time with respect to the number of freezing operators needs to be minimised by automatised translation of a formula to an equivalent formula with a minimal number of freezing operators. In general, in lines of [290, 351], identification of computationally-efficient fragments (or individual formulae) of STL* is a necessary step for making the method practicable. A related problem that is needed to be overcome is to make the specification of STL* formulae more accessible to users not familiar with advanced temporal logics. The possible solution is to provide a tool for visual specification of temporal patterns. Several steps have been done in those directions but all are limited to temporal logics without value-freezing [227]. A GUI for plain textual specification of STL formulae extended with timed regular expressions is a part of the AMT tool [297]. Another interesting direction is learning of formulae from data [19, 41, 68, 226, 295, 372]. Yet none of the existing approaches reflects the expressive power of STL*.

To support the usage of STL* in the field of systems biology, the logic has to be embedded within the well-known tools for parameter synthesis and monitoring of biological models [145, 165]. Our tool Parasim is not yet equipped with an adequate support for efficient parameter synthesis as is done e.g. in Breach [145] by employing parameter sensitivity analysis. Another step is to embed STL* in existing tools [14, 297] that are able to work with Simulink models used in systems engineering. An interesting direction is embedding of STL* within existing frameworks for analysis of stochastic systems [74, 73, 71]. The work mentioned in this thesis addresses quantitative monitoring performed on signals generated offline. An important problem is to develop algorithms for quantitative online monitoring. In the domain of real-value signals and STL the problem has been recently addressed with both qualitative [278, 221] and quantitative semantics [140, 141]. However, it remains open for STL*.

On the theoretical side, it remains to be formally proved for selected classes of signals how is the expressive power of STL* related with expressiveness of the plain STL and xSTL [297] (the STL extended with timed regular expressions). In general, the clarification of the expressive power has to be set with respect to the availability of first- and higher-order derivatives of the signal. Finally, a non-trivial open problem is to transfer the notion of time robustness known in the context of STL [151] into the setting of STL*.

6.3 Probabilistic Robustness for CSL

For stochastic models represented by means of sCRNMs we address robustness analysis from the perspective of the bounded-time fragment of Continuous Stochastic Logic (CSL) with rewards (Section 2.5.5). Formally, we target the problem stated in Definition 3.4. In particular, the goal is to compute the robustness degree for a given CSL formula, given parameterised sCRNM, and a perturbation space of interest. In our approach, the robustness analysis is based on an evaluation function that characterises the probability of satisfaction of a given formula under the given parameter valuation.

To solve the problem, we have adopted the concept of parameterised uniformisation described in Chapter 5. The landscape function provided by parameterised uniformisation is considered as the evaluation function for which several notions of robustness degree are introduced. The robustness degree induced by the landscape function can be understood as the property of the systems stochasticity. The probability of a specified formula to be satisfied for a given parameter valuation gives us a measure characterising how robust the specified behaviour is while reflecting the noise in the dynamics (occurrence of random events in continuous time). The notion of robustness degree directly relativises this concept with respect to a given perturbation space.

6.3.1 Method Principles

Consider a reaction network $\mathcal{N}$ and an sCRNM $\mathcal{M} = (\mathcal{N}, \nu_{sRN}, \mathbb{X}, X_0)$ with a parameterisation $\chi_{\mathcal{M}}$ employing valuations from a perturbation space $\mathbf{P} \subseteq \mathcal{P}_{\chi_{\mathcal{M}}}$ as defined in Section 2.4.3. As mentioned in Section 2.4.5, the dynamics of the parameterised model $\mathcal{M}_{\mathcal{P}_\chi}$ of $\mathcal{N}$ is represented by a parameterised CTMC $\mathbf{C} = \{\mathcal{C}_\mu \mid \mu \in \mathbf{P}\}$.

Robustness Degree Variants

To characterise the robustness, we need to define a suitable *evaluation function* that fits in the robustness degree definition stated in Section 2.6.7. To that end, we consider an auxiliary function $Eval(\mathcal{C}_\mu, \varphi)$ that returns the numerical value representing the quantitative model-checking result for the CTMC $\mathcal{C}_\mu$ and the formula φ. In particular, the possible probability threshold $\sim r$ (where $\sim \in \{\geq, >, \leq, >\}$) in the top-most operator of φ is ignored (i.e., it is treated as $=?$). Given these settings the *evaluation function* $D_\varphi^{\mathbf{C}}$ can be set in several different ways:

$$D_\varphi^{\mathbf{C}}(\mu) = \begin{cases} 0 & \mu \in \mathbf{B} \subset \mathbf{P} \quad \vee \quad Eval(\mathcal{C}_\mu, \varphi) \nsim r \\ 1 & \mu \in \mathbf{P} \setminus \mathbf{B} \quad \wedge \quad Eval(\mathcal{C}_\mu, \varphi) \sim r \end{cases} \tag{6.1a}$$

$$D_\varphi^{\mathbf{C}}(\mu) = \begin{cases} 0 & \mu \in \mathbf{B} \subset \mathbf{P} \\ \frac{Eval(\mathcal{C}_\mu, \varphi)}{r} & \mu \in \mathbf{P} \setminus \mathbf{B} \quad \wedge \quad \sim \in \{\geq, >\} \\ \frac{r}{Eval(\mathcal{C}_\mu, \varphi)} & \mu \in \mathbf{P} \setminus \mathbf{B} \quad \wedge \quad \sim \in \{\leq, <\} \end{cases} \tag{6.1b}$$

$$D_\varphi^{\mathbf{C}}(\mu) = \begin{cases} 0 & \mu \in \mathbf{B} \subset \mathbf{P} \\ Eval(\mathcal{C}_\mu, \varphi) & \mu \in \mathbf{P} \setminus \mathbf{B} \end{cases} \tag{6.1c}$$

$$D_\varphi^{\mathbf{C}}(\mu) = \begin{cases} 0 & \mu \in \mathbf{B} \subset \mathbf{P} \\ |Eval(\mathcal{C}_\mu, \varphi) - X|^2 & \mu \in \mathbf{P} \setminus \mathbf{B} \end{cases} \tag{6.1d}$$

where $X = agr\{Eval(\mathcal{C}_\mu, \varphi) \mid \mathcal{C}_\mu \in \mathbf{P}\}$, $agr \in \{min, max, avg\}$ and $\mathbf{B}$ is a subspace of all perturbations, where the system's function is completely missing. The degree of robustness, denoted as $\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P}}$, is defined as the integral of the evaluation function $D^{\mathbf{C}}_{\varphi}$ over the perturbation space $\mathbf{P}$:

$$\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P}} \overset{df}{=} \int_{\mathbf{P}} D^{\mathbf{C}}_{\varphi}(\mu) d\mu.$$

Definitions (6.1a) and (6.1b) are possible for specifications where the topmost operator of the formula φ includes the threshold r. In the first definition (6.1a) the evaluation function $D^{\mathbf{C}}_{\varphi}(\mu)$ returns a qualitative result, therefore the robustness degree $\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P}}$ specifies the measure of all perturbations in $\mathbf{P}$ for which the property holds in a strictly Boolean sense – it is the fraction of $\mathbf{P}$ where the property is valid. This definition can be used, e.g., with the CSL property $\varphi = \mathsf{P}_{\geq 0.8}[\mathsf{F}^{[0,5]}(X > 300)]$, which specifies that in 80% of cases the population X increases above 300 within 5 seconds. For this property and a model with a parameter $k \in [0, 10]$ the robustness gives us the fraction of the parametric interval $[0, 10]$ for which the model satisfies φ.

In the second definition (6.1b) the evaluation function $D^{\mathbf{C}}_{\varphi}(\mu)$ returns the quantitative value that is relative to threshold *r*. Therefore, robustness can be interpreted as the average relative validity of the property over $\mathbf{P}$. If *r* corresponds to the validity of φ in conditions considered natural for the inspected parameterised CTMC $\mathbf{C}$ (i.e., to the unperturbed state) then this interpretation complies with the original definition of Kitano. Let us consider the same property φ_A and the same parametric space $k \in [0, 10]$. If for all values of *k* the model has a 60% change that its behaviour will lead to a population of X larger than 300 within 5 seconds than its robustness is 0.6/0.8 = 0.75. If the probability is different in each *k* then robustness gives us the average value with which our expectations will be met.

The third definition (6.1c) is possible for specifications using the quantitative semantics of formula φ. Here robustness gives the mean validity over all $\mathbf{P}$, regardless of any probability threshold *r*. This interpretation is convenient when there are no *a priori* assumptions about the system's expected behaviour.

Finally, to express the fact that the system behaviour remains the same (with respect to the evaluation function) across the space of perturbations, we introduce the fourth definition (6.1d). It uses an aggregation function to compute a mean value and expresses the variance from the mean. This definition enables us to compare models which have the same numerical values of robustness in the sense of definition (6.1c) but which achieve the average value with very different landscapes of evaluation function.

While the last three definitions require precise computation of the probability value in every $p \in \mathbf{P}$, the first definition is amenable to approximate solutions. In this case it suffices to ensure that the probability is larger or smaller than *r*. In many cases it can be achieved without computing the precise value and thus statistical model checking techniques can be used efficiently.

Robustness Analysis Procedure

Having the definition of the evaluation function $D^{\mathbf{C}}_{\varphi}$ we can describe an effective method for computation of the robustness degree $\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P}}$. Let us first consider the case where the perturbation space $\mathbf{P}$ does not contain different initial states.

The evaluation of $D^{\mathbf{C}}_{\varphi}(\mu)$ includes the computation of $Eval(\mathcal{C}_{\mu}, \varphi)$, i.e., the solution of standard CSL model checking problem. Since the problem can be rather complex even for a single perturbation point p, an explicit computation of the integral over the whole space of perturbations is infeasible. Therefore, we consider an approximation of the evaluation function $D^{\mathbf{C}}_{\varphi}$ using the upper bound $D^{\mathbf{C}}_{\varphi,\mathbf{P},\top}$ and the lower bound $D^{\mathbf{C}}_{\varphi,\mathbf{P},\perp}$ with respect to $\mathbf{P}$ defined as:

$$\begin{array}{l} D^{\mathbf{C}}_{\varphi,\mathbf{P},\top} \geq max\left\{D^{\mathbf{C}}_{\varphi}(\mu) \mid \mu \in \mathbf{P}\right\} \\ D^{\mathbf{C}}_{\varphi,\mathbf{P},\perp} \leq min\left\{D^{\mathbf{C}}_{\varphi}(\mu) \mid \mu \in \mathbf{P}\right\} \end{array} \tag{6.2}$$

This approximation is in most cases too course and thus we use a finite decomposition of the perturbation space $\mathbf{P}$ into perturbation subspaces $\mathbf{P} = \mathbf{P}_1 \cup \ldots \cup \mathbf{P}_n$. This approach allows to effectively compute the upper bound $\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\top}$ and lower bound $\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\perp}$ of the robustness degree $\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P}}$ in the following way:

$$\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\top} = \sum_{i=1}^{n} \frac{|\mathbf{P}_i|}{|\mathbf{P}|} \cdot D^{\mathbf{C}}_{\varphi,\mathbf{P}_i,\top} \qquad \mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\perp} = \sum_{i=1}^{n} \frac{|\mathbf{P}_i|}{|\mathbf{P}|} \cdot D^{\mathbf{C}}_{\varphi,\mathbf{P}_i,\perp}$$

$$\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P}} \simeq \frac{1}{2}\left(\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\top} + \mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\perp}\right) \pm \textsc{Err}^{\mathbf{C}}_{\varphi,\mathbf{P}} \qquad \textsc{Err}^{\mathbf{C}}_{\varphi,\mathbf{P}} = \frac{1}{2}\left(\mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\top} - \mathrm{rd}^{\mathbf{C}}_{\varphi,\mathbf{P},\perp}\right)$$

As we can see, the key step in our approach is computation of the values $D^{\mathbf{C}}_{\varphi,\mathbf{P},\top}$ and $D^{\mathbf{C}}_{\varphi,\mathbf{P},\perp}$ for the given CTMC $\mathbf{C}$, the formula φ and the perturbation space $\mathbf{P}$. In particular, the method of parameterised uniformisation described in Chapter 5 is adapted to effectively approximate the evaluation function $D^{\mathbf{C}}_{\varphi}(\mu)$. Intuitively, for a formula φ the algorithm computes the upper and lower bounds of the function $Eval(\mathcal{C}_{\mu}, \varphi)$ with respect to all perturbation points $\mu \in \mathbf{P}$. In consequence, these bounds are used to obtain the values $D^{\mathbf{C}}_{\varphi,\mathbf{P},\top}$ and $D^{\mathbf{C}}_{\varphi,\mathbf{P},\perp}$ such that Equation 6.2 is satisfied.

6.3.2 Applications

We have conducted two extensive case studies that have proved the method gives valuable insights into biological systems modelled in terms of sCRNMs. The results are published in [359].

First, the method has been applied to a known model of gene regulation of the mammalian cell cycle [345] that explains the transition from G_1-phase to S-phase. In particular, at this point, an irreversible decision is made causing the cell either proceeding with the cell division or ceasing the division. The problem incorporates a so-called biological switch that is in computational systems biology analysed by methods of bifurcation analysis. Since the system is based on interactions among proteins and genes, stochasticity plays a crucial role and therefore it is important to get insights into the robustness of the switch under randomness of individual molecular interactions. After rewriting the original deterministic model (a cINM) into a sCRNM we employed probabilistic robustness analysis of the model. The results have brought new insights into robust stabilisation of the switch in either of the two decisions.

Second, we have applied our technique to provide comparison of two different models (sCRNMs) of a two-component signalling pathway in *Escherichia Coli* with respect to robustness of a signal transfer under randomness of involved molecular interactions and fluctuations of molecule numbers. The analysis gives insights into how robustness of signal transfer can be increased by synthetically modifying the respective reaction network. The systems have been previously analysed in [338] analytically by means of deterministic models confirming higher robustness in a synthetically modified pathway. Our analysis has brought new insights that could not be obtained with the deterministic framework. In particular, we have shown that in the stochastic setting the synthetic pathway gives higher robustness of signal transfer only for low signals whereas in case of strong signals the natural pathway performs significantly better. Thus case study is overviewed in Section 7.5.

6.3.3 Performance Evaluation

The time complexity of our framework in practice depends mainly on the size of the state space, the number of reaction steps that have to be considered, and the number of perturbation sets that have to be analysed to provide the desired precision. The size of the state space is given by the number of species and their populations. The framework is suitable for low populations and is relevant especially in the case of gene regulation. In the first case study we have considered only a single molecule of DNA and thus the state space of resulting CTMC was manageable. In the second case study, we have abstracted the feedback loop mechanism using a sigmoid production function to reduce the state space and to make the analysis feasible. If such an abstraction cannot be used, our framework can be effectively combined with general state space reduction methods for CTMCs, e.g., finite projection techniques [292, 220], dynamic state space truncation [142], and aggregation methods [371]. The number of reaction steps can be reduced using separation of fast and slow reactions as demonstrated in the second case study or using adaptive uniformisation [357, 142].

In the first case study, several hundreds of perturbation subsets had to be analysed and the overall robustness analysis took a few hours. However, in the second case study several thousands of perturbation subsets where required to achieve reasonable precision making the computation demanding.

6.3.4 Publications Summary

The work mentioned in this chapter has been entirely published in the paper [359]. The paper includes a supplementary material including formal details related to the method. The main text of the paper is primarily devoted to the biological case studies as they are of main interest to the community. Additionally, the technical aspects of the method are comprehensively provided in the technical report [83].

This research has been done in collaboration with Luboš Brim, Milan Češka, and Sven Dražan. My contribution was in setting the notion of robustness for the framework of stochastic models and temporal formulae. I have contributed to development of the method and formalisation of the elementary steps. Significant part of my contribution is in the case studies where I have guided the procedure and helped with interpretation of the results (Section 7.5).

6.3.5 Discussion

Our framework for robustness analysis of stochastic biochemical systems is entirely based on the parameterised uniformisation discussed in Chapter 5. It allows us to quantify and analyse how the validity of a hypothesis formulated as a temporal property depends on the perturbations of stochastic kinetic parameters. The framework extends the quantitative model checking techniques and numerical methods for CTMCs and adapts them to the needs of stochastic modelling in biology. Therefore, in contrast to statistical techniques such as Monte Carlo simulation, parameter sampling and adaptive grid refinement [40]), our framework is customisable with respect to the required precision of computation. This is obtained by providing the lower and upper bounds of the results that allow us to rigorously focus on a considered perturbation space of interest and to provide a detailed analysis of the evaluation function.

It is worth noting that the evaluation function can be discontinuous or may change its value rapidly on a very small perturbation interval in situations when the given CSL formula contains nested probability operators. In particular, this leads inevitably to the formulation of a hypothesis requiring a detailed temporal program [369] of the biological system (e.g., temporal ordering of events). This makes another reason why there is a need to guarantee the approximated shape of the evaluation function.

Both applications have demonstrated that the framework can be successfully applied to the robustness analysis of nontrivial sCRNMs. They have shown how to use CSL to specify properties targeting transient behaviour under fluctuations in a finite time horizon. The lesson learned from the applications is that there exist properties that cannot be directly formulated using CSL with rewards. To that end, we have introduced a concept of further processing the evaluation function to express and study the mean quadratic deviation of the molecule population distribution of the signal response regulator protein. This is described within the particular case study in Section 7.5.

6.3.6 Future Work

The crucial points for future work are similar to those stated in Chapter 5. Having robustness analysis techniques deployed on the top of hybrid (combined deterministic and stochastic) dynamics will allow to bring in better scalability necessary for analysis of larger scale models.

Additionally, an ongoing work is to focus on combining the parameterised uniformisation framework with statistical techniques. This extension has the potential to efficiently scan multidimensional parameter spaces and to identify interesting subspaces that can be analysed in detail using the framework. Several steps have been already done in these directions (see Section 5.6).

A standalone task is to implement a software tool that supports robustness analysis with a suitable GUI allowing visualisation of the results.

6.4 Software Tools

The algorithm for deterministic robustness with STL* is implemented on the top of the tool Parasim [86]. The tool is developed as a modular environment for monitoring and robustness analysis of continuous-time deterministic models. It reads a specification in the form of STL and STL* formulae in an XML format and a model in the SBML level 2 format. The main functionality reflects the scheme in Figure 6.1 employed for cCRNM, cINM or cRINM models represented in SBML. Robustness analysis of the model is performed with respect to a given specified perturbation of model parameters which is sampled upto a given minimal distance among sampled points of the perturbation space. To evaluate the robustness, the quantitative semantics of STL and STL* is computed for each sampled parameter valuation. The tool is open source. It is written in Java and available on GitHub[1]. The tool has been developed in several phases supported by student projects and theses supervised by myself. The students that have significantly contributed to the tool development are Jan Papoušek, Tomáš Vejpustek, Samuel Pastva, Vojtěch Brůža, and Aleš Pejznoch.

The algorithm for probabilistic robustness analysis with CSL has been implemented by Milan Češka as a prototype using the procedures of the PRISM 4.0 model checker [261]. An experimental implementation employing parallelisation on GPU that has been implemented as a follow-up work by Petr Pilař can be also used as a back-end for robustness analysis. However, a tool that would implement the specific support for robustness analysis is still missing.

[1] https://github.com/sybila/parasim

Chapter 7

Case Studies

In this chapter, we present a selection of case studies applying the methods presented in previous chapters to several models used in systems biology. Most of the case studies cover a large part of the workflow scheme presented in Figure 2.9 and in most of the cases include reconstruction (or reformulation) of the models. Some of the presented case studies (i.e., Section 7.2 and Section 7.5) present several new insights into the studied biological problem.

7.1 Overview

The presented case studies cover most of the model classes defined in Chapter 2. In particular, biodegradation of 1,2,3-trichloropropane (Section 7.2), regulation of G_1/S cell cycle transition (Section 7.3), mammalian cells signalling pathways (Section 7.4), and population models (Section 7.6) employ the framework of cCRNMs, cRINMs, or cINMs. The study of regulation of G_1/S cell cycle transition also includes a version realised in the framework of sCRNMs. Two-component signalling pathway study (Section 7.5) is done entirely in the framework of sCRNMs.

The case studies in Section 7.2, Section 7.3, and Section 7.4 address the problem of parameter synthesis. Parameter exploration problem is considered in Section 7.3 and robustness analysis is tackled in Section 7.5 and Section 7.6.

7.2 Biodegradation of 1,2,3-trichloropropane in *E. coli*

In this case study, we target a real problem solved in synthetic biology of metabolic pathways. The work has been realised in collaboration with Loschmidt laboratories of the Faculty of Science. The results demonstrate the applicability of the methodology provided in Chapter 4 in systems and synthetic biology workflows. The tool Pythia described in Section 4.4 is employed to perform the analysis presented in this section.

7.2.1 Problem Description

A synthetic pathway for conversion of the highly toxic 1,2,3-trichloropropane (TCP) to glycerol (GLY) in *Escherichia coli* was assembled as described in [256]. TCP is an emerging toxic groundwater pollutant and suspected carcinogen which spreads to the environment mainly due to improper waste management. According to [256] no naturally occurring bacterial pathway is capable of degradation of TCP. A synthetic reaction network consisting of five intermediates with glycerol as a final product and utilising enzymes from other bacterial species was assembled (Figure 7.1). The individual reactions in this pathway are positively influenced (catalysed) by enzymes which we consider as parameters of the respective model (they are considered constant as they reflect the energetical environment for the pathway).

The employed enzymes are haloalkane dehalogenase (DhaA) from *Rhodococcus rhodochrous* and haloalcohol dehalogenase (HheC), epoxide hydrolase (EchA) from *Agrobacterium radiobacter*. They have a major role in this pathway. In order to achieve an efficient implementation of the pathway it is important to quantitatively characterise mutual interplay and optimal concentration levels of these enzymes. In general, the higher is the enzyme concentration the higher is the flux rate. Especially, if a substrate and its intermediates are more or less toxic to a host cell such a requirement becomes critical.

Unfortunately, the solution is not straightforward because each of the enzymes has a distinct rate and some of the intermediate products are much more toxic than the others. Additionally, since these enzymes are not natural proteins in *E. coli*, they have to be produced at the expense of other substances. This is called a *metabolic burden*. In other words, there must be a balance in concentrations of these enzymes in order to degrade TCP as fast as possible while not killing the host by enervation. Therefore, we employ the workflow of parameter synthesis to give preliminary results targeting the complex problem of fine-tuning optimal enzyme concentration levels.

$$\mathbf{TCP} \xrightarrow{\text{DhaA}} \mathbf{DCP} \xrightarrow{\text{HheC}} \mathbf{ECH} \xrightarrow{\text{EchA}} \mathbf{CPD} \xrightarrow{\text{HheC}} \mathbf{GDL} \xrightarrow{\text{EchA}} \mathbf{GLY}$$

$$\begin{aligned}
\frac{d[TCP]}{dt} &= -\frac{k_1 \cdot DhaA \cdot [TCP]}{K_{m,1}+[TCP]} & \frac{d[CPD]}{dt} &= \frac{k_3 \cdot EchA \cdot [ECH]}{K_{m,3}+[ECH]} - \frac{k_4 \cdot HheC \cdot [CPD]}{K_{m,4}+[CPD]} \\
\frac{d[DCP]}{dt} &= \frac{k_1 \cdot DhaA \cdot \cdot [TCP]}{K_{m,1}+[TCP]} - \frac{k_2 \cdot HheC \cdot [DCP]}{K_{m,2}+[DCP]} & \frac{d[GDL]}{dt} &= \frac{k_4 \cdot HheC \cdot [CPD]}{K_{m,4}+[CPD]} - \frac{k_5 \cdot HheC \cdot [GDL]}{K_{m,5}+[GDL]} \\
\frac{d[ECH]}{dt} &= \frac{k_2 \cdot HheC \cdot [DCP]}{K_{m,2}+[DCP]} - \frac{k_3 \cdot EchA \cdot [ECH]}{K_{m,3}+[ECH]} & \frac{d[GLY]}{dt} &= \frac{k_5 \cdot HheC \cdot [GDL]}{K_{m,5}+[GDL]}
\end{aligned}$$

$$k_1 = 1.05, k_2 = 0.751, k_3 = 14.37, k_4 = 2.38, k_5 = 3.96,$$

$$K_{m,1} = 1.79, K_{m,2} = 1.00, K_{m,3} = 0.09, K_{m,4} = 0.86, K_{m,5} = 3.54$$

Figure 7.1: (top) An abstract scheme of the original system. Note that enzymes HheC and EchA are employed twice on the pathway. The reverse mass flow is considered negligible and abstracted away. (bottom) The mathematical model. Enzyme concentrations are considered as constant (and unknown) parameters. Units: $k_x(s^{-1})$, $K_{m,x}(mM)$.

7.2.2 Model Encoding

First we denote the individual reactions in the pathway respectively $\varrho_1, ..., \varrho_5$. Every catalytic reaction is modelled by using Michaelis-Menten kinetics [286] at the place of the respective regulated kinetic function:

$$\kappa'_{\varrho_i}(x) = \nu_{cRIN}(\varrho) \cdot \frac{k_i \cdot x}{K_{m,i} + x}$$

where $k_i, K_{m,i}$ are fixed as shown in Figure 7.1. For each ϱ_i the kinetic function coefficient $\nu_{cRIN}(\varrho_i)$ represents the (constant) concentration of the particular enzyme: $\nu_{cRIN}(\varrho_1)$ represents $DhaA$; $\nu_{cRIN}(\varrho_2) = \nu_{cRIN}(\varrho_4)$ represent $HheC$, and $\nu_{cRIN}(\varrho_3) = \nu_{cRIN}(\varrho_5)$ represent $EchA$. There is a shared parameter for the sets of reactions $\{\varrho_2, \varrho_4\}$ and $\{\varrho_3, \varrho_5\}$ and therefore three parameters are considered uncertain in total.

7.2.3 Analysis Procedure and Results

The original model taken from [155] was reduced in order to minimise the dimensionality of the system. Redundant reactions are eliminated based on their rates and catalytic efficiency (defined as $\frac{k_x}{K_{m,x}}$, see Table 7.1). In general, greater catalytic efficiency means a faster reaction flux towards generation of the product. Since we need to preserve the number of unknown parameters, the very first reaction cannot be omitted. Reaction towards CPD is undeniably the fastest reaction of the model not just due to the best catalytic efficiency but also because of the highest affinity which is an alternative interpretation of the reciprocal Michaelis constant. Therefore this

Table 7.1: Model reactions including enzymes, reaction constants and additional information about catalytic efficiency.

reaction	*enzyme*	*rate (k)*	*Michaelis const. (K_m)*	*cat. efficiency ($\frac{k}{K_m}$)*
TCP→DCP	DhaA	1.050	1.79	0.587
DCP→ECH	HheC	0.751	1.00	0.751
ECH→CPD	EchA	14.370	0.09	159.670
CPD→GDL	HheC	2.380	0.86	2.767
GDL→GLY	EchA	3.960	3.54	1.119

reaction can be omitted in our model. The reaction towards GDL has the second fastest flux and since it is much faster than the last reaction it can be omitted as well. Finally, we have reduced the model to only three reactions which significantly helps to reduce the model state space while making the investigation of the three uncertain parameters tractable.

The desired property is defined verbally as *"complete degradation of TCP as fast as possible with the least accumulated toxicity"*. The notion of toxicity is based on the inhibitory concentration of particular molecules. Our framework is designed for manipulation with differential expressions rather than with numerical assignments. Hence we are not able to directly observe the actual amount of toxicity. But the toxicity has a direct connection to the concentrations of intermediates. To this end, we translate the desired property as *"TCP completely degrades and the concentration of intermediates does not exceed given bounds"*. The bounds are based on experimental data of the original model (Figure 7.2) with the default setting of parameters (DhaA $= 0.003$, HheC $= 0.0036$, EchA $= 0.0029$ (mM)) and initial concentrations ($[TCP]$ $= 2$ mM, $[other\ species]$ $= 0$ mM). Constants are shown in Figure 7.1. The presented data reveal that the considered boundary is reasonable for the concentration 0.5 mM or less. In consequence, we proceed by testing various combinations of bounds for GDL and DCP in the interval $[0, 0.5]$ of mM.

It has been mentioned that the concentration of enzymes cannot be unlimited due to the metabolic burden (which is not the object of investigation in this paper). According to the default values the initial constraints for these parameters are therefore set to the interval $[0.0, 0.02]$ of mM. The parameter synthesis workflow employing the discrete abstraction of the cRINM as presented in Section 2.4.6 is employed with the CTL CMC procedure (Section 4.3) to find parameter values satisfying the desired property. The template of CTL formula expressing the property (denoted as φ) is a combination of several smaller subformulae:

$$\varphi_1 = (\mathbf{AG}\ [TCP] < y),\ \ \varphi_2 = (\mathbf{A}([TCP] > x)\mathbf{U}(\mathbf{AF}\ \varphi_1)),$$
$$\varphi_3 = (\mathbf{AG}\ [GLY] > x),\ \ \varphi_4 = (\mathbf{A}([GLY] < y)\mathbf{U}(\mathbf{AF}\ \varphi_3)),$$

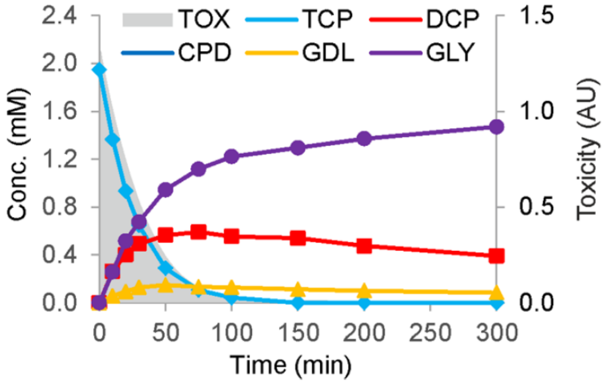


Figure 7.2: Experimental data from the original model [256]. We are interested just in the progress of TCP, DCP, GDL and GLY taken as variables of our reduced model.

$$\varphi_6 = (\mathbf{AG}\ [DCP] < v),\ \ \varphi_7 = (\mathbf{AG}\ [GDL] < w),$$

$$\varphi_5 = (\varphi_2 \wedge \varphi_4),\ \ \varphi_8 = (\varphi_5 \wedge \varphi_6),\ \ \varphi = (\varphi_8 \wedge \varphi_7),$$

where x, y, v and w are estimated values making a particular instance of this property. Here $x = 1.9$ (according to [256] where authors use the value 2 mM), $y = 0.01$ (cannot be zero), $v \in \{0.5, 0.3, 0.1\}$ and $w \in \{0.5, 0.25, 0.1\}$ (variations based on an observation of the experimental data in Figure 7.2).

The result of parameter synthesis is the set of initial states (satisfying φ) each accompanied with a set of respective values of the parameters (DhaA, HheC, EchA). Results are encoded as a formula in the SMT-LIB format 2.5 [39]. Consequently, to compare and visualise satisfactory parameter values in a human-readable form some post-processing is necessary. In this case, we run a script that systematically samples and visualises the parameter space encoded by the formula (by calling the SMT solver iteratively). The result is the graphical representation of the parameter subspace constraint by φ and the initial parameter constraints. In Figure 7.3 the results are shown for a specific initial state.

Note that due to the global nature of our algorithm all states satisfying the property have been found. The concentration of all variables in this case study has been restricted to the interval $[0.0, 5.0]\ mM$. Our framework reveals parameter values satisfying φ also for initial states beyond the singular initial concentration of particular species considered in [256]. The most interesting are the initial states that increase the upper limit of TCP concentration wrt φ (Figure 7.4).

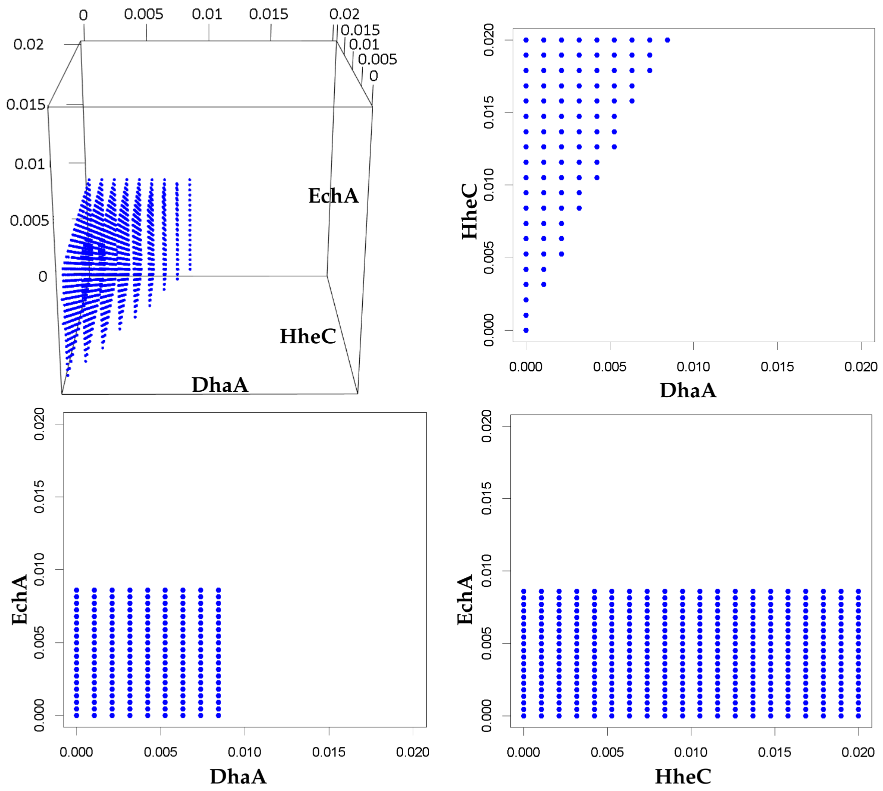


Figure 7.3: A sample of the resulting parameter space for a particular initial state: TCP $\in [1.9, 1.9586]$, DCP $\in [0.448898, 0.5]$, GDL $\in [0.0, 0.0669138]$, GLY $\in [0.0, 0.01]$. Dotted area corresponds to φ ($v = 0.5$, $w = 0.25$). Upper left figure the 3D space sampled with 400 points per a layer. Remaining figures display projections of the 3D plot for every combination of unknown parameters. All values are shown in mM.

7.2.4 Performance

Owing to the global nature of the enumerative CTL model checking algorithm all the subformulae are investigated during the process. However, the computation is time and space demanding and the utility of parallel algorithm has to be employed. The computation took more than one day on a single node while less than 2 hours on twenty nodes (each node equipped with a common hardware – Intel Xeon quad-core 2GHz and 16 GB of RAM).

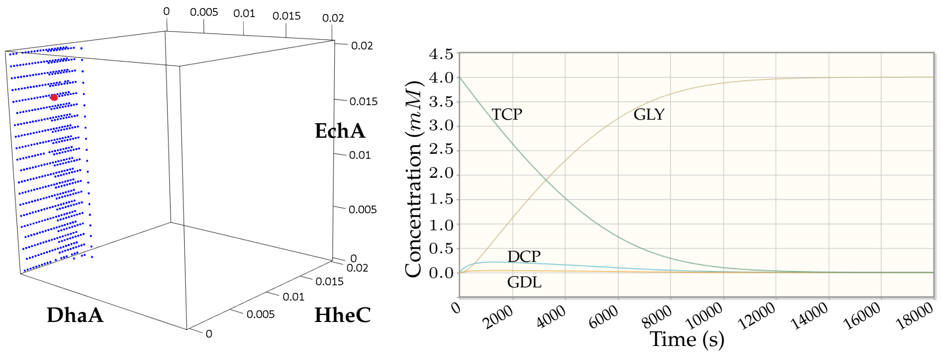


Figure 7.4: (left) Resulting parameter space for a specific initial state: TCP $\in$ $[3.84186, 5.0]$, DCP $\in$ $[0.0, 0.448898]$, GDL $\in$ $[0.0, 0.0669138]$, GLY $\in$ $[0.0, 0.01]$. The red dot shows the selected point for parameters values: DhaA = 0.001, HheC = 0.005, EchA = 0.015. (right) Numerical simulation for the selected point. All values are in mM. Simulation was obtained in BIOCHAM.

7.3 Regulation of G_1/S cell cycle transition

This case study is focused on demonstrating that the methodologies presented in Chapter 4 and Chapter 5 can provide a valuable (fully automatised) alternative to existing numerical and analytical techniques used for exploring how the system reacts to changes in parameters and what are the critical points in the parameter space that cause significant qualitative changes in systems behaviour.

$$\begin{aligned} \frac{d[pRB]}{dt} &= k_1 \frac{[E2F1]}{K_{m1}+[E2F1]} \frac{J_{11}}{J_{11}+[pRB]} - \gamma_{pRB}[pRB] \\ \frac{d[E2F1]}{dt} &= k_p + k_2 \frac{a^2+[E2F1]^2}{K_{m2}^2+[E2F1]^2} \frac{J_{12}}{J_{12}+[pRB]} - \gamma_{E2F1}[E2F1] \end{aligned}$$

$$a = 0.04,\, k_1 = 1,\, k_2 = 1.6,\, k_p = 0.05,\, \gamma_{pRB} = 0.005$$
$$\gamma_{E2F1} = 0.1,\, J_{11} = 0.5,\, J_{12} = 5,\, K_{m1} = 0.5,\, K_{m2} = 4$$

Figure 7.5: An ODE model of the G_1/S transition regulatory network presented in Figure 2.2 (right).

We address the model describing regulatory interactions controlling a transition between two phases of a mammalian cell cycle [345]. In particular, the model explains the core mechanism behind the irreversible decision for cell division described by a two-gene regulatory network of interactions between the tumour suppressor protein pRB and the central transcription factor $E2F1$. The respective influence network is depicted in Figure 2.2 (right) with the corresponding ODE model shown in Figure 7.5.

7.3.1 Problem Description

For suitable parameter valuations of the two parameters, two distinct stable attractors may exist (the so-called *bistability*). The problem is typically targeted by the so-called bifurcation analysis that is performed by using analytical and numerical methods. In [345] an analytical bifurcation analysis of $E2F1$ stable concentration depending on the degradation parameter of pRB (γ_{pRB}) has been provided. Note that traditional methods for bifurcation analysis hardly scale to more than a single model parameter.

7.3.2 Model Encoding

The model does not fit the Definition 2.14 due to the occurence of k_p as a summant in the right-hand side of the second differential equation. It represents a constant inflow (basal transcription) of the protein $E2F1$ and cannot be neglected. To that end, the model is encoded in terms of a cRINM considering the following reactions underlying the influence network given in Figure 2.2 (right):

$$\begin{array}{llll} \varrho_1 : & E2F1 \xrightarrow{\gamma_{E2F1}} & \varrho_3 : & pRB \xrightarrow{\gamma_{pRB}} \\ \varrho_2 : & \xrightarrow{k_p} E2F1 & \varrho_4 : & \rightarrow pRB \end{array}$$

The regulated kinetic functions are defined for the reactions ϱ_2 and ϱ_4 in the following way:

$$\kappa'_{\varrho_2}(x,y) = k_p + \nu_{cRIN}(\varrho_2) \cdot \frac{a^2 + y^2}{K_{m2}^2 + y^2} \cdot \frac{J_{12}}{J_{12} + x}$$
$$\kappa'_{\varrho_4}(x,y) = \nu_{cRIN}(\varrho_4) \cdot \frac{y}{K_{m1} + y} \cdot \frac{J_{11}}{J_{11} + x}$$

where $x \equiv pRB$, $y \equiv E2F1$ and all other parameters are fixed to the values shown in Figure 7.5.

The parameterisation of the model is given by the parameterisation set $\chi = \{\varrho_2, \varrho_4\}$ where the kinetic function coefficients are given as the parameters $\nu_{cRIN}(\varrho_2) \equiv k_1$ and $\nu_{cRIN}(\varrho_4) \equiv \gamma_{pRB}$.

7.3.3 CTL-based Analysis

In this section we employ the algorithm presented in Section 4.3 to parameter synthesis of the above mentioned model. The technique gives an alternative way to perform bifurcation analysis typically solved by analytical methods limited to restricted classes of system dynamics. In particular, we focus on the synthesis of values of two interdependent parameters.

Problem Settings

The property of *bistability* expresses that the system is able to settle in two distinct stable states (i.e., levels of concentration) for specific initial conditions and particular parameter values. It implies existence of a decision-making point (or area) in the system.

The main outcome of the original analysis is shown in Figure 7.6 (left) (produced by numerical analysis) displaying the dependency of stable concentration of $E2F1$ on value of γ_{pRB} (degradation rate). The most interesting area called *unstable* (for $\gamma_{pRB} \in [0.007, 0.027]$) determines feasible values of γ_{pRB} wrt the above property. For $\gamma_{pRB} < 0.007$ the system converges to a lower-concentration stable equilibrium whereas for $\gamma_{pRB} > 0.027$ it converges to a higher-concentration stable equilibrium.

The CTL representation of the property in consideration is $\varphi_1 = (\mathbf{EF}[\mathbf{AG}[low]] \wedge \mathbf{EF}[\mathbf{AG}[high]])$ where $low = (0.5 < E2F1 < 2.5)$ (representing safe cell behaviour) and $high = (4 < E2F1 < 7.5)$ (representing

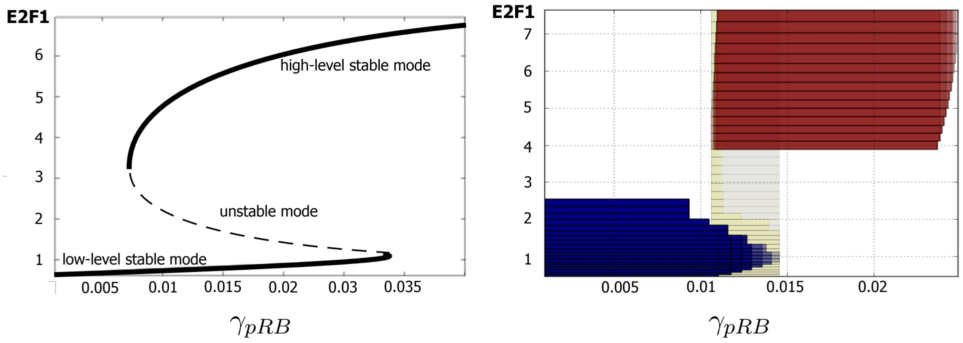


Figure 7.6: (left) Equilibrium diagram reproducing the results achieved in [345]. (right) Visualisation of the parameter synthesis results. The high and low stable regions are represented by the red and blue coloured areas, respectively. The yellow areas denote the states in which the *bistable switch* formula φ_1 is satisfied.

excessive cell division). During the single run of our algorithm all subformulae of φ_1 have been analysed. Let $\varphi_2 = (\mathbf{AG}[low])$ and $\varphi_3 = (\mathbf{AG}[high])$ as the most interesting.

Analysis Procedure and Results

With respect to the three properties, we have investigated perturbations of a single parameter γ_{pRB} with the initial constraint $\gamma_{pRB} \in [0.001, 0.025]$. According to Section 2.4.6 we have first created the PWMA approximation of the original cINM (Figure 7.5 (right)) by approximating each non-linear function in the right-hand side of ODEs with a sum of optimal sequence of piece-wise affine ramp functions (the precision has been set to 70 automatically generated segments per each non-linear function). For such a setting the verification process took less than 10 seconds on twenty nodes. The obtained results are visualised in Figure 7.6 (right). The plot intentionally depicts the same space as the Figure 7.6 (left) to show obvious similarities of these results. The blue area stands for stable concentration of $E2F1$ (y-axis) with a particular value of γ_{pRB} (x-axis) satisfying the property φ_2, whereas the red area satisfies the property φ_3. The yellow area (in the middle) stands for possibility of reaching both stable concentrations. Due to the mixing of existential and universal quantifiers (see Sec. 4.3.2), the results achieved for φ_1 cannot be exactly interpreted. On the contrary, the results for φ_2 and φ_3 are guaranteed due to the conservativeness of the abstraction.

Although the algorithm based on interval-based encoding (Section 2.4.6) performs fast, it is limited to independent parameters only. To overcome this limitation, we have employed the SMT-based encoding (Sec-

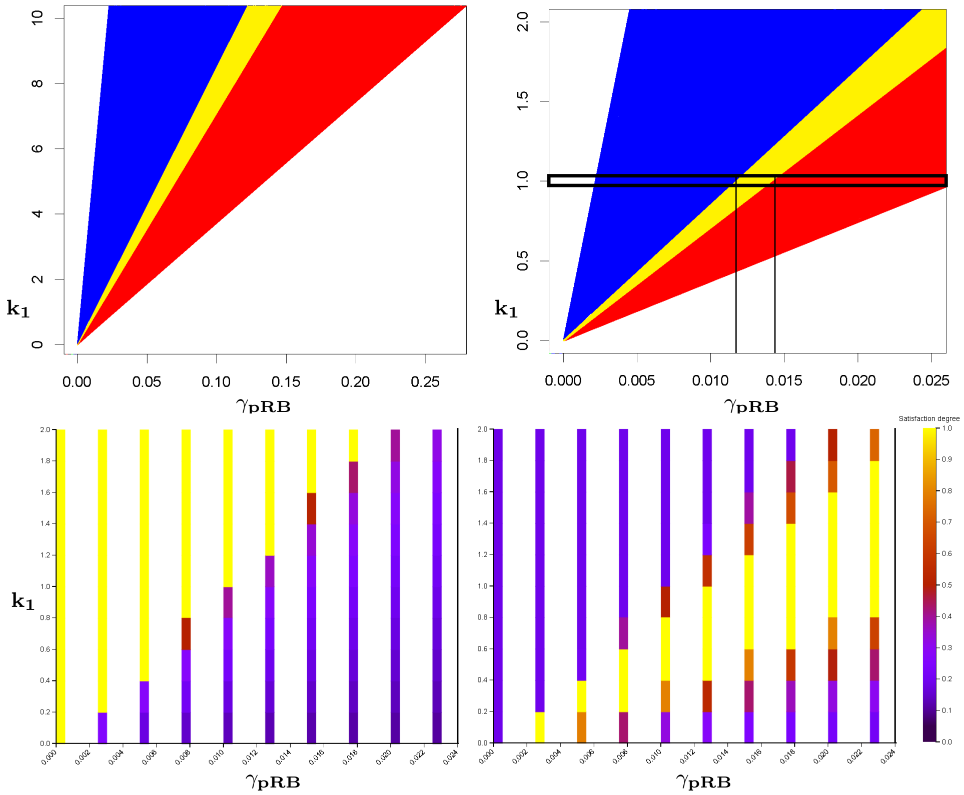


Figure 7.7: (top left) The resulting parameter space merged for all initial concentrations. Each area corresponds to a different property: φ_1 (yellow), φ_2 (blue) and φ_3 (red). (top right) The same parameter space magnified and projected to γ_{pRB}-axis. The framed region agrees with the original numerical bifurcation analysis performed in [345] for γ_{pRB}. (bottom) Landscapes of the parameter space according to the quantitative satisfaction degree computed by BIOCHAM for φ_2 (left) and φ_3 (right), respectively.

tion 2.4.6) to explore two uncertain mutually dependent parameters. The method is computationally more demanding (about one order of magnitude for each pair of dependent parameters). The goal of our extended analysis is to explore the mutual effect of the degradation parameter of pRB (γ_{pRB}) and the production parameter of pRB (k_1) on the bistability. Additionally, we perform post-processing of achieved results by employing additional constraints on the parameter space (e.g., imposing a lower and upper bound on the production/degradation parameter ratio) and show an alternative way of presenting the results.

In particular, we involve the SMT-based tool Symba [266] to obtain an approximated interval of the bounds on valid parameter values. Since the considered parameters are linearly dependent, the resulting intervals cannot be simply combined to display the two-dimensional validity area in the parameter space. To this end, we have employed Symba to explore the ratio of the two parameters. By combining initial parameter constraints with the bounds on the parameter ratio, a more accurate parameter subspace is acquired. Such an outcome has been used with the initial constraint $\gamma_{pRB} \in [0.001, 0.1]$ and $k_1 \in [0.001, 10]$ (Figure 7.7 (top left)). Additionally, we have explored a refined parameter space ($\gamma_{pRB} \in [0.001, 0.025]$ and $k_1 \in [0.001, 2]$) where a one-dimensional projection on the γ_{pRB}-axis is highlighted for $k_1 \approx 1$, the default value of k_1 (Figure 7.7 (top right)).

The analysis took 8 minutes on twenty nodes (excluding post-processing). The obtained results can be used as a base for further analysis. We employ the feature of BIOCHAM [94] to compute the *landscape* function that allows investigation of quantitative satisfaction degree of the properties explored (Figure 7.7 (down)). LTL reformulation of φ_2 and φ_3 has been used (φ_1 cannot be expressed in LTL). The lighter is the colour the higher the satisfaction degree.

7.3.4 CSL-based Analysis

It is important to note that the cINM described in previous section targets the dynamics from the perspective of the average population view. However, it is well known [195] that in cases of low molecular numbers the stochasticity of the dynamics can affect the behaviour of the system significantly. Especially, in the case of networks addressing interactions among proteins through the transcription/translation apparatus (the so-called gene regulation), the stochasticity has to be taken into account due to the fact that the proteins interact with a single copy of DNA. This fact means that the chance of a successful transcription-inducing interaction between a given regulating protein and a given regulated gene to occur in a certain time is affected by the cellular noise.

Model Reformulation

To address the problem of bistability discussed in the previous section, we have reconstructed the model in the stochastic framework of sCRNMs. This is done because we need to fit within the stochastic mass action kinetics. In particular, we have reformulated the original CIN as a CRN that gives an approximate mechanistic view of the mechanism. The resulting CRN is shown in Figure 7.8. The correspondence with the model presented in the previous section is the following: A denotes the protein pRB, a stands for a gene encoding pRB, B denotes the protein $E2F1$, and b stands for a gene encoding $E2F1$. Since the detailed knowledge of elementary chemical reactions occurring in the process of transcription and translation is incomplete, we use the simplified form as suggested in [159]. In the minimalistic setting, the reformulation requires addition of rate parameters describing the transcription factor–gene promoter interaction while neglecting cooperativeness of transcription factors activity. Our pasteurisation is based on time-scale orders known for the individual processes [366] (parameters considered in s^{-1}). Moreover, we assume the numbers of A and B are bounded by 10 molecules. Upper bounds for A and B are set with respect to behaviour of an ensemble of stochastic simulations. We consider minimal population number distinguishing the two stable modes. All other species are bounded by the initial number of DNA molecules (genes a and b) which is conserved and set to 1. The considered parameterisation is limited to the degradation coefficients γ_A, γ_B assigned to the degradation reactions of A and B, respectively. In this settings, the corresponding parameterised CTMC has 1078 states.

Gene a interactions		Gene b interactions	
$a \rightarrow a + A$	1	$b \rightarrow b + B$	0.05
$aB \rightarrow aB + A$	1	$bB \rightarrow bB + B$	1
$A + a \leftrightarrow aA$	$100; 10$	$A + b \leftrightarrow bA$	$100; 10$
$B + a \leftrightarrow aB$	$100; 10$	$B + b \leftrightarrow bB$	$100; 10$
Protein degradation			
$A \rightarrow$	γ_A	$B \rightarrow$	γ_B

Figure 7.8: Chemical reaction network reformulation of the G_1/S regulatory circuit – a, b represent genes, aA, aB, bA, bB represent transcription factor-gene promoter complexes.

Analysis Procedure and Results

We consider three hypotheses: (1) stabilisation in the low mode where $B < 3$, (2) stabilisation in the high mode where $B > 5$, (3) stabilisation in the high mode where $B > 7$ ((3) is more focused than (2)). All the hypotheses are expressed within time horizon 1000 seconds reflecting the time scale of gene regulation response. We employ two alternative CSL formulations to express each of the three hypothesis. According to [345], we consider the parameter space $\gamma_A \in [0.005, 0.5]$ and γ_B to be fixed to the default value.

First, we express the property of being inside the given bound during the time interval $I = [500, 1000]$ using globally operator: (1a) $\mathsf{P}_{\sim ?}[\mathsf{G}^I\,(B < 3)]$, (2a) $\mathsf{P}_{\sim ?}[\mathsf{G}^I\,(B > 5)]$ and $(3a)$ $\mathsf{P}_{\sim ?}[\mathsf{G}^I\,(B > 7)]$. The interval starts from 500 seconds in order to bridge the initial fluctuation region and let the system stabilise. Since the stochastic noise causes molecules to repeatedly escape the requested bound, the resulting probability is significantly lower than expected. Namely, in cases (2a) and (3a) the resulting probability is close to 0 for the whole parameter space. Moreover, the selection of an initial state has only a negligible impact on the result. Therefore, in Figure 7.9 only the resulting probability for case (1a) and a single selected initial state is visualised.

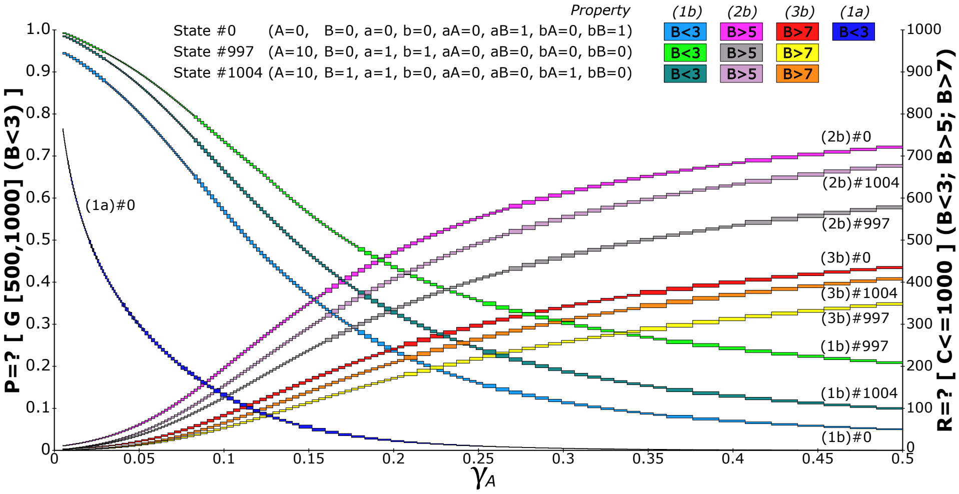


Figure 7.9: Landscape functions of properties *(1a,1b,2b,3b)* for $\gamma_A \in [0.005, 0.5]$ (in s^{-1}) and initial states #0, #997 and #1004. The left Y-axis scale corresponds to *(1a)*, the right to *(1b,2b,3b)*.

Second, we use a cumulative reward property to capture the fraction of the time the system has the required number of molecules within the time interval $[0, 1000]$: $(1b)$ $\mathsf{R}_{\sim ?}[\mathsf{C}^{\leq t}](B < 3)$, $(2b)$ $\mathsf{R}_{\sim ?}[\mathsf{C}^{\leq t}](B > 5)$, $(3b)$ $\mathsf{R}_{\sim ?}[\mathsf{C}^{\leq t}](B > 7)$ where $t = 1000$ and $\mathsf{R}_{\sim ?}[\mathsf{C}^{\leq t}](B \sim X)$ denotes that state reward ρ is defined such that $\forall s \in \mathbb{S}.\rho(s) = 1$ iff $B \sim X$ in s. The result is visualised for three selected initial states in Figure 7.9.

Figure 7.9 also illustrates inaccuracy of our approach with respect to the absolute error bound $\textsc{Err} = 0.01$ by means of small rectangles depicting approximations of the resulting probabilities and expected rewards. The analyses predict that the distribution of the low steady mode interferes with the distribution of the high steady mode. It confirms bi-stability predicted in [345] but in contrast to the deterministic analysis our method shows how the population of cells distributes around the two stable states. Results of computations including the number of iterations performed during parameterised uniformisation, numbers of resulting subspaces and execution times in hours, are presented in Figure 7.10.

Finally, to see how degradation rates of A and B cooperate in affecting property $(3b)$, we explore two-dimensional parameter space $(\gamma_A, \gamma_B) \in [0.005, 0.1] \times [0.05, 0.1]$. The computation also required $4.0 \cdot 10^6$ iterations of the parameterised uniformisation, the parameter decomposition resulted in 143 subspaces for $\textsc{Err} = 0.1$ and the overall execution took 14 hours. Figure 7.11 illustrates the computed upper bound of the landscape func-

Property	# iter.	# subsp.	time[h]
(1a)	1.2·10^6	153	9
(2a)	2.0·10^6	69	5.5
(3a)	2.0·10^6	66	4.5
(1b)	4.0·10^6	159	10.5
(2b)	4.0·10^6	132	8
(3b)	4.0·10^6	80	5

Figure 7.10: Computation performance results.

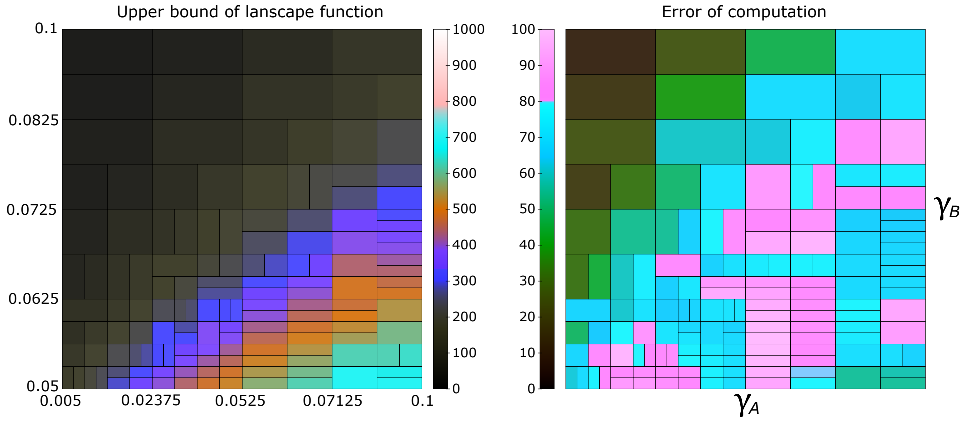


Figure 7.11: Landscape function for property $(3b)$, initial state #0 $(A = 0, B = 0, a = 0, b = 0, aA = 0, aB = 1, bA = 0, bB = 1)$ and two-dimensional parameter space $(\gamma_A, \gamma_B) \in [0.005, 0.1] \times [0.05, 0.1]$ (represented in s^{-1} by X and Y axes, respectively). On the left, the upper bound of the landscape function is illustrated. On the right, the absolute error given as difference between computed upper and lower bounds is depicted. In both cases the color scale is used.

tion for initial state #0 and the absolute error. The result predicts antagonistic relation between the degradation rates which is in agreement with the study provided in [345].

7.4 Sustained vs Transient Modes of Cell Signalling

In this section we employ the CTL-based parameter synthesis methods described in Chapter 4 to differentiate models that display or do not display a particular studied dynamical behaviour expressed in the form of a temporal property. Details on the modelling and analysis provided here have been published in [210].

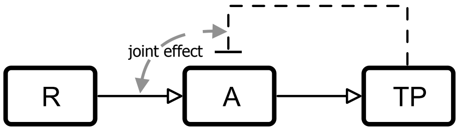
7.4.1 Problem Description

Signalling pathways represent one of the most important biochemical mechanisms studied in current systems biology. In particular, they provide a complex cellular information processing machinery that evaluates input stimuli and transfers them into genome by means of regulation of specific genes expression. A special emphasis is given to distinguishing between monotonous (*sustained*) and non-monotonous (*transient*) time-course behaviour of signalling pathways [324, 365]. It is believed that transition between these two modes may cause a significant change of the nominal cell behaviour leading to serious anomalies of internal cellular processes control.

7.4.2 Model Encoding

We consider three cBNM models falling to the class of cINMs. These models describe a general shape of signalling pathways at a high level of abstraction. In particular, we focus on three topologies of influence networks differing in the presence/absence of feedback mechanisms (Figure 7.12). We use Hill kinetics employing sigmoidal functions to describe the response of a signalling component with respect to the input signal (see equations in Figure 7.12). In particular, our models consist of the species $\Lambda = \{A, R, TP\}$ where R is a receptor, A an adapter, and TP a target protein. The adapter forms the main dynamical entity of the model. It is activated by the receptor and inhibited by the target protein. Receptor concentration is considered constant, hence the positive influence of the adapter from the side of the receptor is considered as constantly active and its dynamics is not considered in the model.

We consider three model variants differing in influences of the adapter. Model 1 reflects the topology with no inhibition (the dashed interaction is removed and Equation 7.3 is employed). Note that the constant activity of the receptor influence is reflected by a constant regulation function ($\sigma^1_A(x) = 1$). Model 2 describes independent inhibition (the dashed interaction is considered to be superposed with the constant activation of A by R, reflected in Equation 7.4). In particular, there is a separate regulation function for each of the influences of A. Model 3 describes dependent inhibition (the effect of the inhibition is multiplied with the activation by R, reflected in Equation 7.5) using a regulation function multiplying the constant activation with the negative influence modelled as a sigmoidal function (Equation 7.2). In this case, there is a single regulation function that combines both influences of A. Target protein dynamics is modelled with a positive regulation function (Equation 7.1) of the adapter in all three models (Equation 7.6).



$$h(X, K_{\mathrm{M}}, n) = \frac{[X]^n}{{K_{\mathrm{M}}}^n + [X]^n} \tag{7.1}$$

$$h^{-}(X, K_{\mathrm{M}}, n) = \frac{{K_{\mathrm{M}}}^n}{{K_{\mathrm{M}}}^n + [X]^n} \tag{7.2}$$

Figure 7.12: Influence networks representing the considered variants of signalling pathways. Dashed line represents optional inhibition (left). Positive and negative regulation function employing Hill kinetics (right).

$$\frac{d[A]}{dt} = V_A - y_A[A] \tag{7.3}$$

$$\frac{d[A]}{dt} = V_A + V_{\mathrm{MAX\_}A} \cdot h^{-}(TP, K_{\mathrm{M\_}A}, n_A) - y_A[A] \tag{7.4}$$

$$\frac{d[A]}{dt} = V_A' \cdot h^{-}(TP, K_{\mathrm{M\_}A}, n_A) - y_A[A] \tag{7.5}$$

$$\frac{d[TP]}{dt} = V_{\mathrm{MAX\_}TP} \cdot h(A, K_{\mathrm{M\_}TP}, n_{TP}) - y_{TP}[TP] \tag{7.6}$$

Parameter V_A' is defined as $V_A \cdot V_{\mathrm{MAX\_A}}$. Default parameters have been set to: $V = V_A' = V_{\mathrm{MAX\_}A} = V_{\mathrm{MAX\_}TP} = 0.001$, $K_{\mathrm{M\_}A} = K_{\mathrm{M\_}TP} = y_A = y_{TP} = 0.1$, $n_A = n_{TP} = 2$. Initial concentrations of all entities have been set to 0: $A(0) = TP(0) = 0$.

Parameterisation of the models is set to the production rate coefficients so that the following coefficients are considered as parameters: V_A in Model 1, $V_{\mathrm{MAX\_}A}$ in Model 2, V_A' in Model 3 and $V_{\mathrm{MAX\_}TP}$ in all three models. These parameters are examined in the range [0.0001,10] and other constants are fixed at default values with the only exception of y_A, y_{TP} (set to 0.5009) and V_A in Model 2 (set to 0.001).

7.4.3 Analysis Procedure and Results

In order to prepare the model for model checking analysis, we have performed the steps described in Section 2.4.6. Firstly, the piece-wise affine approximation (PWMA) of the original non-linear continuous models has been constructed by applying the automatised approximation procedure introduced in [205]. In particular, we have approximated each non-linear function appearing in the right-hand side of the model equations with a sum of ten piece-wise affine ramp functions. Secondly, the abstraction procedure described in Section 2.4.6 has been applied to the PWMA model. The only extension to the procedure is the addition of a new information stored with the transitions. In particular, every transition in the abstracted Kripke structure $\mathcal{K}^{abst}$ is assigned a label representing the index of the model variable affected by the transition and the corresponding direction ($+$,$-$) in which the change occurs.

Properties are formulated in terms of UCTL formulae (the basic CTL logic extended with action-label predicates [346]). The usage of branching time logic is motivated by the fact that the rectangular abstraction produces a non-deterministic system. Combination of action and state predicates is necessary to express local patterns of the dynamics (state predicates) and the character of the transitions (action predicates).

We employ the parameter synthesis based on coloured CTL model checking extended to deal with UCTL operators. Since we restrict ourselves to particular operators as needed in the properties of our interest, the extension is a direct refinement of the algorithm described in Section 4.3. The UCTL operators employed are the following:

- $\mathbf{EX}_{\{i-\}}\varphi$ requiring existence of at least one state satisfying φ to be reached by a transition that decreases the variable i,
- $\mathbf{AX}_{\{i+\}}\varphi$, $\mathbf{EX}_{\{i+\}}\varphi$ requiring all states (resp. existence of at least one state) reached by a transition that increases the variable i to satisfy φ,
- $\mathbf{AF}_{\{\neg i-\}}\varphi$, $\mathbf{EF}_{\{\neg i-\}}\varphi$ requiring all paths (resp. existence of a path) to a state satisfying φ provided that no transition decreases the value of variable i — satisfying paths must be non-strictly increasing,
- $\mathbf{EF}_{\{\neg i+\}}\varphi$ requiring existence of a path to a state satisfying φ on which the variable i is non-strictly decreasing.

The parameter synthesis procedure gives us a global result saying for which initial states and respective parameter values the formula is satisfied. It is important to note that the over-approximating abstraction affects interpretation of the parameter synthesis results. On the one hand, satisfaction is guaranteed for universally quantified formulae. On the other hand, falsification is guaranteed for existentially quantified formulae. Quantifiers cannot be alternated. In both cases of guaranteed results, the obtained parameter values are under-approximated.

To express combined characteristics of the two behaviours of TP, we have employed the following properties in the form of UCTL formulae:

- $\varphi_1 = Init \wedge \mathbf{AX}_{\{TP+\}}(\mathbf{AF}_{\{\neg TP-\}}\, Stable)$,
- $\varphi_2 = Init \wedge \mathbf{EF}(\mathbf{EX}_{\{TP+\}}(\mathbf{EF}_{\{\neg TP-\}}(\mathbf{EF}_{\{\neg TP+\}}(\mathbf{EX}_{\{TP-\}}\, True))))$,
- $\varphi_3 = Init \wedge \mathbf{EF}(\mathbf{EX}_{\{TP+\}}(\mathbf{EF}_{\{\neg TP-\}}(\mathbf{EF}_{\{\neg TP+\}}(\mathbf{EX}_{\{TP-\}}\, NotUp))))$,

where $Init$ stands for the set of initial states (with TP constrained in the range [0.0,0.0001]), $NotUp$ for a state in which the abstracted vector field forbids an increase in TP concentration, $Stable$ for an equilibrium state with both species being stable and, finally, $True$ for any state. Intuitively, property φ_1 specifies the sustained behaviour and its satisfaction guarantees presence of this behaviour in a given model for given initial states. Formally, this formula restricts all admissible runs to start in $Init$ and to increase TP concentration in the very next step and not to decrease it before reaching $Stable$. The properties φ_2 and φ_3 represent necessary conditions for presence of the transient behaviour (formulae are ordered with respect to the strength of the particular condition). Both restrict any feasible run to start in $Init$ and to increase TP concentration at least once on a run before decreasing it also at least once on a run. The parameterisations violating these properties guarantee the absence of the transient behaviour. All results are summarised in Table 7.2.

model type	property	initial concentration of A	$V^* \times V_{\mathrm{MAX\_}TP}$
Model 1	φ_1	[0.22,11.9]	[0.11,5.96]×[0.0,6.81]
	$\neg\varphi_2$	[0.22,12.0]	[0.11,10.0]×[7.23,10.0]
	$\neg\varphi_3$	[0.01,12.0]	[0.23,10.0]×[6.26,10.0] ∪[5.96,10.0]×[0.0,10.0]
Model 2	φ_1	[0.01,11.5]	[0.0,5.76]×[0.0,0.01]
	$\neg\varphi_2$	[0.22,12.0]	[2.29,10.0]×[7.23,10.0]
	$\neg\varphi_3$	[0.0,12.0]	[0.09,10.0]×[7.23,10.0] ∪[4.47,10.0]×[0.0,10.0]
Model 3	φ_1	[0.01,11.5]	[0.01,5.76]×[0.0,0.01]
	$\neg\varphi_2$	[0.22,12.0]	[2.31,10.0]×[7.23,10.0]
	$\neg\varphi_3$	[0.0,12.0]	[0.11,10.0]×[7.23,10.0] ∪ [4.49,10.0]×[0.0,10.0]

Table 7.2: The results obtained for given properties in different models. Initial concentration of A is defined as a union of concentration values in states where the formula holds for stated parameters. V^* is the production parameter of A that represents V_A in Model 1, $V_{\mathrm{MAX\_}A}$ in Model 2, V'_A in Model 3, respectively. Each parameter interval determines a range of all satisfying parameter values across all states of the respective model where the particular property holds.

It is important to note that the abstraction together with UCTL logic allow us to exactly characterise the inevitability of the sustained behaviour.

However, as has been reported in [85], the state predicate $Stable$ is exactly preserved in the abstraction only if a particular equilibrium that must exist in the respective state rectangle is hyperbolic. In consequence, we are not able to fully cover the transient behaviour. More specifically, the transient behaviour might asymptotically converge to an equilibrium that is asymptotically stable. Such a property is not preserved by the abstraction and therefore we limit ourselves only to refuting *absolutely transient* behaviour (transient behaviour without oscillations around the target equilibrium).

7.5 Robustness of Elemental Signalling Pathways

In this section we present a case study utilising the stochastic robustness analysis with CSL described in Section 6.3. The study bring new insights into the mechanism of two-component signalling pathways occuring in bacteria.

Signalling pathways make the main interface between cells and their environment. Their main role is to monitor biochemical conditions outside the cell and to transfer this information into the internal logical circuits (gene regulation) of the cell. Since signal processing is carried out by several dedicated protein complexes (signalling components), it is naturally amenable to intrinsic noise in these protein populations caused by stochasticity of transcription/translation processes. Robust input-output signal mapping is crucial for cell functionality. Many models and experimental studies have been conducted attempting to explain mechanisms of robust signal processing in prokaryotic cells, e.g., [47, 330].

7.5.1 Model Reconstruction

In order to construct robust signalling circuits in synthetically modified prokaryotic cells, Steuer et al. [338] has suggested and analysed a modification of a well-studied two-component signalling pathway that is insensitive to signalling components concentration fluctuations. The study was conducted using a simplified model consisting of the two signalling components each considered in both phosphorylated and unphosphorylated forms. The first component, the histidine kinase H, is a membrane-bound receptor phosphorylated by an external signalling ligand S. In its phosphorylated form Hp, the histidine kinase transfers the phospho-group onto the second component – the response regulator R. That way it activates the response regulator by transforming it into the phosphorylated form Rp, which is diffusible and functions as the internal signal for the cell. The basic topology of the pathway is depicted in Figure 7.13A. The modification suggested by Steuer et al. is depicted in Figure 7.13B. The difference is in the addition of catalytic activation of Rp dephosphorylation by the unphosphorylated histidine kinase H. In [338] it has been rigorously proven that under the deterministic setting this modification leads to globally robust steady-state response of the signalling pathway, which is not achievable with the basic topology.

We reformulate the model in the stochastic setting (in terms of a sCRNM) and employ the stochastic robustness analysis method described in Section 6.3 to provide detailed analysis of the input-output signal response under fluctuations in population of both signalling components. In contrast to [338], where the average steady-state population is analysed with respect to fluctuations in signalling components, our analysis refines the steady population in terms of distributions. That way we obtain for a stable input signal a detailed view of distribution of the output response. In particular, instead of studying the effect of perturbations on the average population, we see how perturbations affect the distribution, i.e., the variance (fluctuation) in the output response. That way the stochastic framework gives a more detailed insight into the input-output signal response mechanism.

The biochemical model of both topology variants is given in Figure 7.13C. The input signal S is considered to be fixed and therefore it makes a constant parameter of the model. The signalling components in both phosphorylated and unphosphorylated forms make the model variables *H*, *Hp*, *R*, and *Rp*.

7.5.2 Problem Description

Depending on which topology is chosen, the original deterministic model [338] exhibits different relationships between the steady-state concentrations of the input signal S and the output signal Rp:

Rp steady-state in model 1

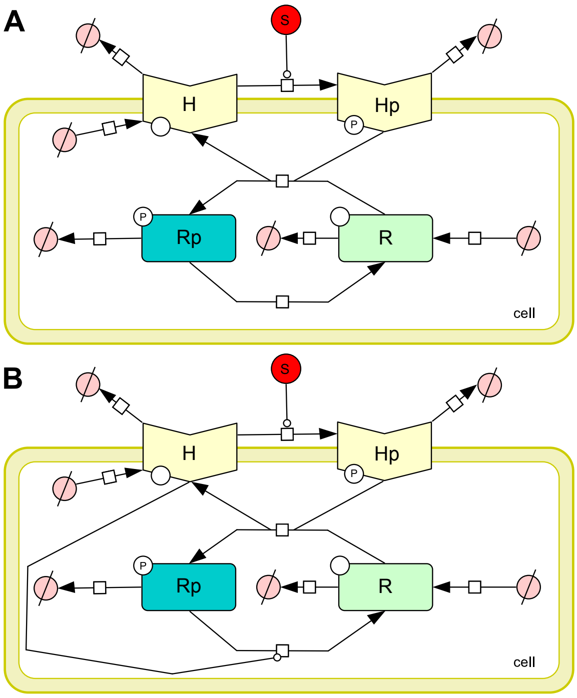
$$[Rp] = \frac{k_1}{k_{31}}[S][H]$$

Rp steady-state in model 2

$$[Rp] = \frac{k_1}{k_{32}}[S]$$

In particular, it can be seen that the steady-state concentration of the output signal $[Rp]$ in model 1 is affected not only by the input signal S but also by the number of unphosphorylated receptors H, which can be interpreted in such a way that the concentration of the signalling components should be kept stable in order to obtain a robust output. This is, however, not an issue in model 2 where Rp depends only on S. Since the steady-state analysis has been carried out under the deterministic setting additionally imposing assumptions of conserved total amounts of *H* + *Hp* and *R* + *Rp*, it is appropriate only for high molecular populations.

The question we want to answer is *"Is there a difference in the way the two models handle noise (fluctuations) for low molecular numbers of signalling components?"* In such conditions, populations of *H* + *Hp* and *R* + *Rp* cannot be considered conserved, since the proteins are subject to degradation and production. Production of proteins from genes, as well as degradation, is inherently noisy as demonstrated in the previous case study. Different levels of noise can be affected by, e.g., regulatory feedback loops or varying numbers of gene copies. Even for a noiseless output signal S these internal fluctuations of protein concentrations transfer noise to Rp.



C

General signal transmission reactions			
$H + S \xrightarrow{k_1} Hp + S$			$k_1 = 0.1$
$Hp + R \xrightarrow{k_2} H + Rp$			$k_2 = 0.1$
Basic topology (model 1)		Modified topology (model 2)	
$Rp \xrightarrow{k_{31}} R$	$k_{31} = 1.0$	$Rp + H \xrightarrow{k_{32}} R + H$	$k_{32} = 15.0$
Signalling components degradation			
$H \xrightarrow{k_d} Ø$	$Hp \xrightarrow{k_d} Ø$		$k_d = 0.01$
$R \xrightarrow{k_d} Ø$	$Rp \xrightarrow{k_d} Ø$		$k_d = 0.01$
Signalling components synthesis			
$Ø \xrightarrow{k_p} R$	$Ø \xrightarrow{k_p} H$		$k_p = 0.3$

Figure 7.13: Model of a two-component signalling pathway. (A) Basic topology of the two-component signalling pathway. (B) Modified topology of the two-component signalling pathway, additionally, histidine kinase H catalyses dephosporylation of the response regulator R. (C) Reactions specifying the biochemical model of the two considered topologies of the two-component signalling pathway.

7.5.3 Post-Processing Functions

To analyse the robustness of signal response, we need to characterise the level of noise (fluctuations) observed on the component *Rp* in a given time. This can be achieved by computing the *mean quadratic deviation* of the distribution of Rp in time.

Standard reward operators do not allow to additionally process the result during the transient analysis. For the purpose of computing the mean quadratic deviation of the distribution of Rp in time, such a procedure is needed. To that end, we extend CSL with an operator of so-called *post-processing functions* denoted $\mathsf{E}_{=?}[\mathsf{I}^t]$ and defined over probability density vectors $\pi^{\mathbf{C},X_0,t}$. For the mean quadratic deviation of noise the post-processing function is denoted $Post(\pi)$ and defined as $Post(\pi^{\mathbf{C},X_0,t}) = \sum_{X\in\mathbb{X}} |\#(A) - mean(\pi^{\mathbf{C},X_0,t}, A)|^2 \cdot \pi^{\mathbf{C},X_0,t}(X)$ where $\#(A)$ gives the population of species A in state X and $mean(\pi^{\mathbf{C},X_0,t}, A)$ is the mean of the distribution $\pi^{\mathbf{C},X_0,t}$ defined as $mean(\pi^{\mathbf{C},X_0,t}, A) = \sum_{X\in\mathbb{X}} \#(A) \cdot \pi^{\mathbf{C},X_0,t}(X)$. The formula $\mathsf{E}_{=?}[\mathsf{I}^t]$ stands for "The mean quadratic deviation of the distribution of species A at time instant t."

7.5.4 Analysis Procedure and Results

For the model to have low numbers of molecules exhibiting stochastic fluctuations and to enable responses to varying levels of S, we have chosen $k_p = 0.3$ *molecules*$\cdot s^{-1}$ and $k_d = 0.01\, s^{-1}$, which leads to an average total population of 30 molecules for both $H + Hp$ and $R + Rp$. To make the analysis straightforward we assume the same speed of degradation of phosphorylated and unphosphorylated variants of each protein.

To reduce the size of the state space we have truncated total populations to $25 \leq H + Hp \leq 35$ and $25 \leq R + Rp \leq 35$, which leads to 116281 states in total. The initial state is considered with populations $s_0 = (H = 30, Hp = 0, R = 30, Rp = 0)$.

Modelling Noise in Signalling Components

In order to control fluctuations in protein production we extend our model with two populations of genes, one for H and one for R, respectively, and for each of the genes we introduce an autoregulatory negative feedback loop via binding the proteins to their corresponding genes. That way we restrict the protein production. By modifying the number of gene copies in the cell and the rate of protein-gene binding, we are able to regulate the overall noise in the transcription. This approach, however, significantly increases the state space size because it introduces new variables representing genes and protein-gene complexes. To make the analysis feasible, we abstract from details of the underlying autoregulatory mechanism and model it using a sigmoid production function, which mimics the desired behaviour accordingly. Using numerical analysis, we have verified that such an approximation can be employed in the stochastic framework. The function is defined in the following way:

$$\emptyset \xrightarrow{sig(k_p,n)} X \qquad\qquad sig(k_p, n) = \frac{2}{1 + \left(\frac{X}{30}\right)^n} \cdot k_p$$

where n is the so-called Hill coefficient controlling the steepness of the sigmoid (caused by cooperativity of transcription factors in protein-gene interactions) and k_P is the maximal production rate. We use this approach for modelling the production of both species H and R by sigmoid coefficients denoted n_H and n_R, respectively. The sigmoid function regulates the population by enabling production when it is below average and repressing it when the population is above the average. Larger *n* is leads to steeper sigmoid functions, which leads to stronger regulation and lower noise. The case $n = 0$ corresponds to an unregulated model and when increased to $n = 20$ it corresponds to over 10 copies of each gene in the fully modelled feedback loop mechanism.

Transient Analysis of Noise in Signalling Components

To see the long-term effects of intrinsic noise we decided to examine the system in the situation when the output response is stabilised. Since the min-max approximation method cannot be employed with steady-state computation, transient analysis in a suitable time horizon has been performed instead. To estimate the closest time *t* when the system's behaviour can be observed as stable, we have computed values of output response noise for the unregulated variant of the model ($n = 0$) using standard numerical steady state analysis (we employed the tool PRISM [261]) and compare it to probability distributions obtained by transient analysis in $t = 20$, $t = 50$ and $t = 100$ seconds. Consequently, we have compared the probability distribution in the steady state with the probability distribution in $t = 100$ seconds. The results clearly show that the difference in distributions is negligible and the transient distribution can be considered stable after $t = 100$.

To further speed up the computation, we have precomputed the distribution of H and R in the time horizon $t = 100$ without enabling phosphorylation reactions. This has led to a significant reduction to 121 states. Starting with the achieved probability distribution, we have subsequently computed the transient analysis with enabled phosphorylation reactions in the next 5 seconds. The rationale behind is that the protein production and degradation are two orders of magnitude slower than phosphorylation. Therefore, the total populations of H and R dictate the time at which the system is nearly stable and thus the next 5 seconds are sufficient for the fast-scale phosphorylation to stabilise the fractions $\frac{H}{Hp}$ and $\frac{R}{Rp}$.

To compute the noise (variance) in Rp we employ the *mean quadratic deviation* post-processing function for state space distributions. Our goal is to compare the levels of Rp noise in both models for different levels of the output signal S and for different values of intrinsic noise appearing in protein production (controlled by sigmoid coefficients n_H and n_R). After computing lower and upper bounds of the state space distributions, we have computed the lower and upper bounds of the post-processing function using the parameterised uniformisation algorithm discussed in Chapter 5. Consequently, we obtain robustness values for the output response R_p over the respective perturbation subspaces in the form *average* ± *error*. Finally, we define the perturbation space of the interest. In particular, for the signal we choose the value interval $S \in [2.0, 20.0]$ and for sigmoid coefficients $n_H, n_R \in [0.1, 10.0]$.

Since the full computation over the 3-dimensional perturbation space has turned out to be intractable, we have to find a way to reduce its dimension. To this end, we focus on a subspace $S = 15.0, (n_H, n_R) \in [3.0, 4.0] \times [3.0, 4.0]$ where both models have symmetric sensitivity to both sigmoid production coefficients n_H, n_R. This symmetry allows us to merge n_H, n_R into a single coefficient *n*. Results of this experiment are visualised in Figure 7.14, where it can be seen that in Model 1 the influence of n_H and n_R is almost perfectly symmetrical with n_H being slightly more influential. In Model 2 the influence is evidently stronger in n_R but the response seems to be symmetrical enough to justify the sigmoid coefficients merging. An interesting property of the parameterised uniformisation and the perturbation space decomposition algorithm can be seen in Figure 7.14, where the decomposition of the perturbation spaces around both sigmoid coefficients set to 3.1 is very dense. This is due to the non-linearity of the sigmoid production functions, which leads to the non-monotonicity of probability inflow/outflow differences in states during parameterised uniformisation. In order to preserve the conservativeness of estimates we have to locally over/under approximate these inflow/outflow rates thus gaining an increase of error. To obtain the desired level of accuracy, we dynamically refine all those subspaces where this has occurred.

Response of Noise to Different Signal Levels

Finally, we inspect selected subintervals of the perturbation space given by five exclusive intervals of the input signal value domain, $S \in [2, 3] \cup [6, 7] \cup [10, 11] \cup [14, 15] \cup [19, 20]$, and three distinct levels of production noise represented by sigmoid coefficient $n \in \{0.1, 4.0, 10.0\}$. The results of this main experiment can be seen in Figure 7.15 and Figure 7.16. The trends that can be seen in Figure 7.15 are that for lower signals up to $S = 10$. Model 2 has

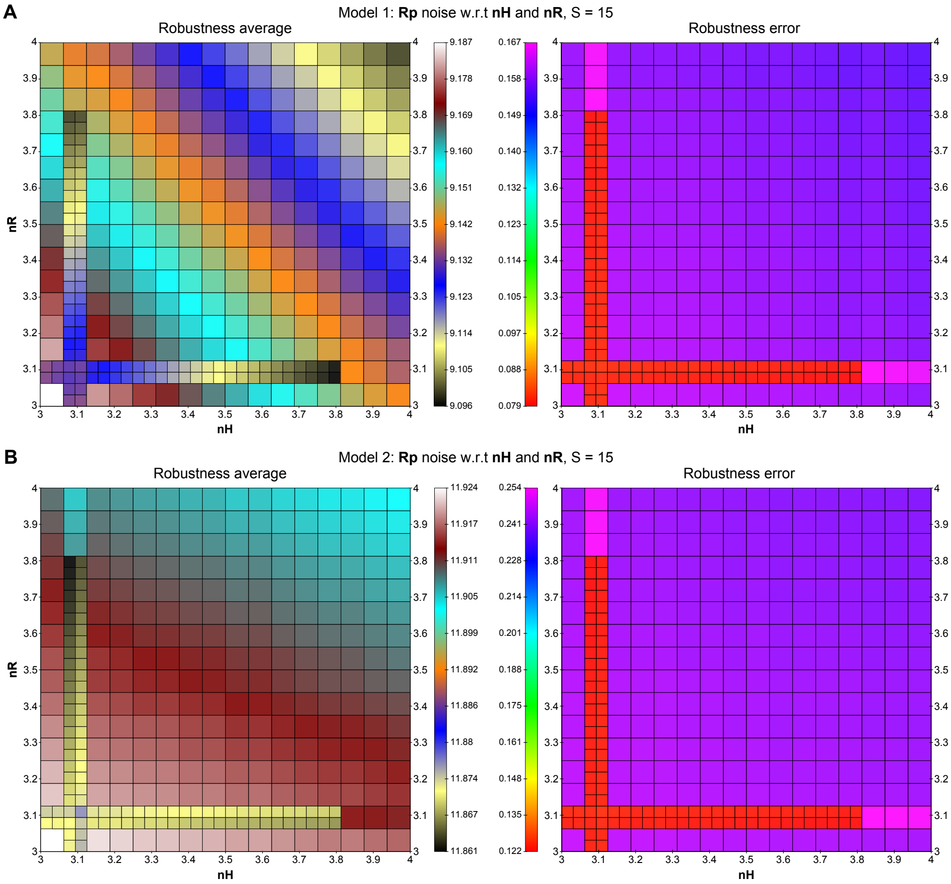


Figure 7.14: Influence of genetic regulation on noise in model 1 and 2. In the upper part, the Rp noise in model 1 is displayed over perturbations of both sigmoid production constants n_H and n_R in $[3.0, 4.0] \times [3.0, 4.0]$. The upper and lower bounds on noise (mean quadratic deviation of the resulting probability distribution projected onto populations of Rp) are recomputed into the form *average* $\pm$ *error*, the average values are shown on the left and errors are shown on the right. In the lower part, the Rp noise in model 2 is displayed.

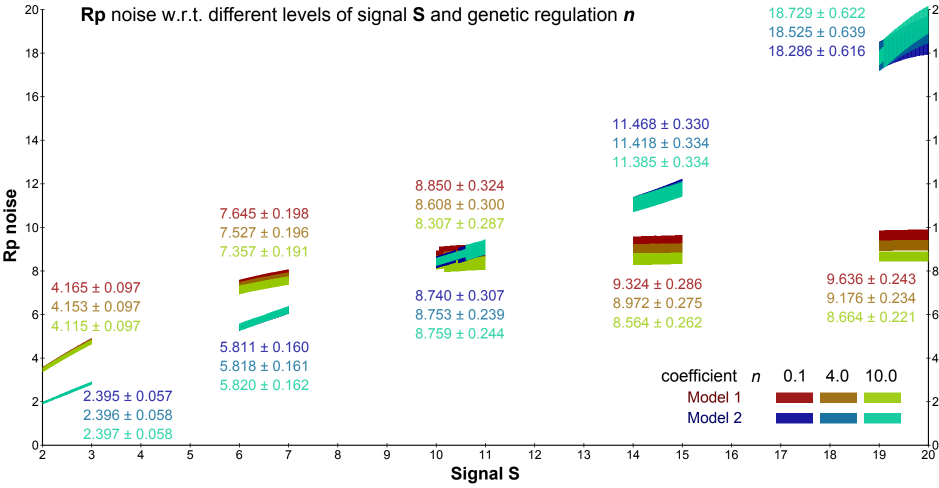


Figure 7.15: Comparison of models by Rp noise robustness. Robustness Rp noise in both models has been computed with respect to perturbations of signal S over five selected intervals of the input signal.

encountered lower noise in Rp than Model 1 but in the higher signal region it is outperformed by Model 1, which quickly converges to values between 8 and 10. However, Rp noise produced in Model 2 linearly increases with increasing value of the input signal S. For most of the inspected subspaces a stronger regulation of H and R production by the sigmoid coefficient n leads to a reduction of Rp noise. An exception to this observation can be seen in Model 2 at the signal interval $[19.0, 20.0]$ where this trend is inverted. To show that this is an emergent behaviour arising from non-trivial interaction between phosphorylation and dephosphorylation reactions not present in the production and degradation of components H and R, their respective influences are displayed in Figure 7.16. There we can see that in Model 1 both H and R follow an initial increase of noise with increasing S but then the noise stabilises. This leads us to a hypothesis that the regulation of noise in signalling components dynamics loses its influence as signal S increases. This is however due to the fact that more S leads to faster phosphorylation of H, which effectively reduces the population of H thus also reducing its absolute noise. In the case of Model 2 the situation is different since we can observe a permanent increase of noise in both H and R populations.

7.5.5 Performance

The parameter space decomposition procedure has iterated through several thousands of perturbation subsets that where required to achieve reasonable precision. In order to speedup the computation we analysed the

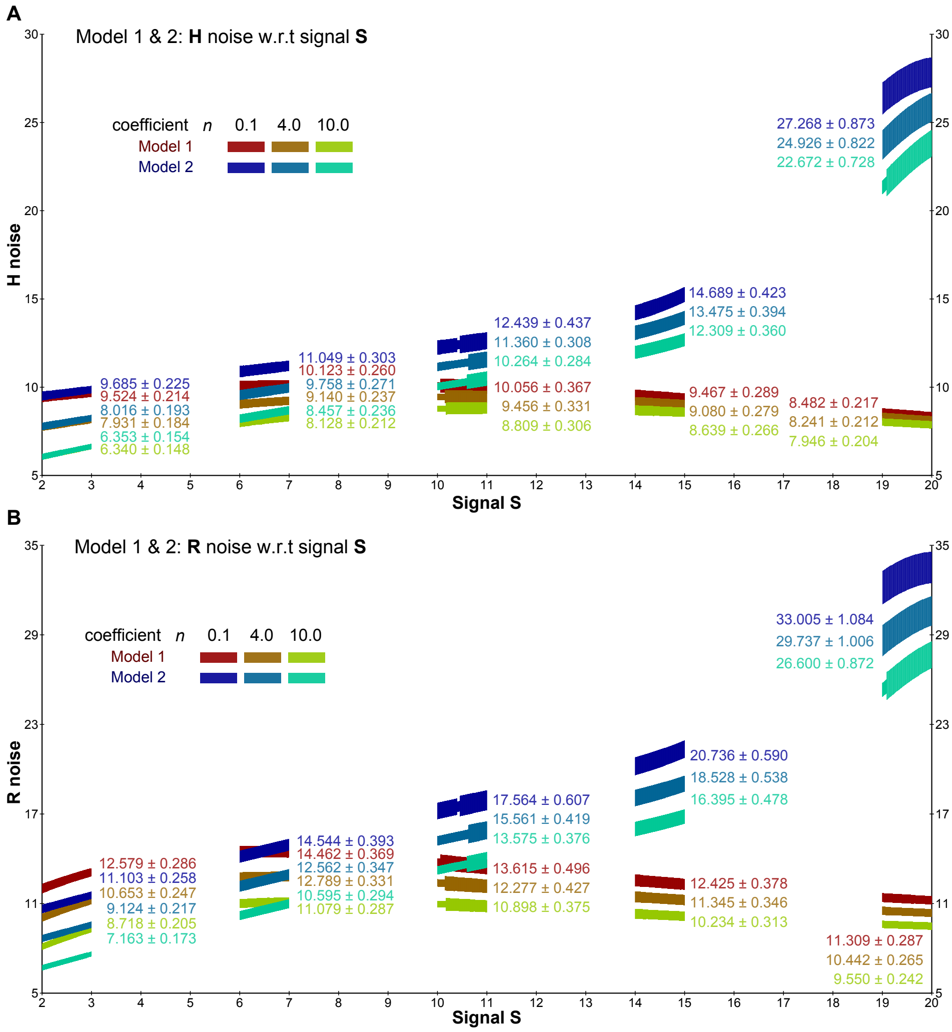


Figure 7.16: Noise in populations or H and R in both models Noise in H (A) and R (B) in both models is displayed with respect to perturbations of signal S over five selected intervals and for three distinct levels of inherent production noise.

subsets in parallel using a high-performance multi-core workstation were the analysis took several hours.

To further improve the accuracy of the robustness analysis without decreasing the performance, piece-wise linear approximation can be employed. It is computed by linearly interpolating the grid points in which the upper and lower bounds of the evaluation function may be computed more precisely as the minimum resp. maximum of the values from all parameter subintervals sharing boundary grid points. It allows us to obtain more precise results without increasing the number of perturbation sets, yet it does not guarantee conservative error bounds.

7.6 Robustness of Population Dynamics

By employing the Parasim tool discussed in Section 6.4 we have conducted two case studies on two basic population dynamics models. The experiments have also served us to evaluate the method presented in Section 6.2 (in the setting of the Parasim tool).

7.6.1 SIR Model

First, we demonstrate the robustness analysis on the model simulating an outbreak of an infectious disease in a population [236]. The simulated population is divided into three categories: *susceptible* (S), *infected* (I) and *recovered* (R). A susceptible individual can become infected by contact with another infected individual and an infected individual may recover. The considered variant of a SIR model is represented as a CRN in the following way:

$$\begin{aligned} S + I &\overset{\alpha}{\rightarrow} I \\ I &\overset{\beta}{\rightarrow} R \end{aligned}$$

where α is the *contact rate* parameter which correlates to probability of disease transmission, while β, the *recovery rate* parameter, takes into account the standard length of recovery. The corresponding parameterised cRNM is given by means of the following system of differential equations:

$$\frac{dS}{dt} = -\alpha SI \qquad \frac{dI}{dt} = \alpha SI - \beta I \qquad \frac{dR}{dt} = \beta I$$

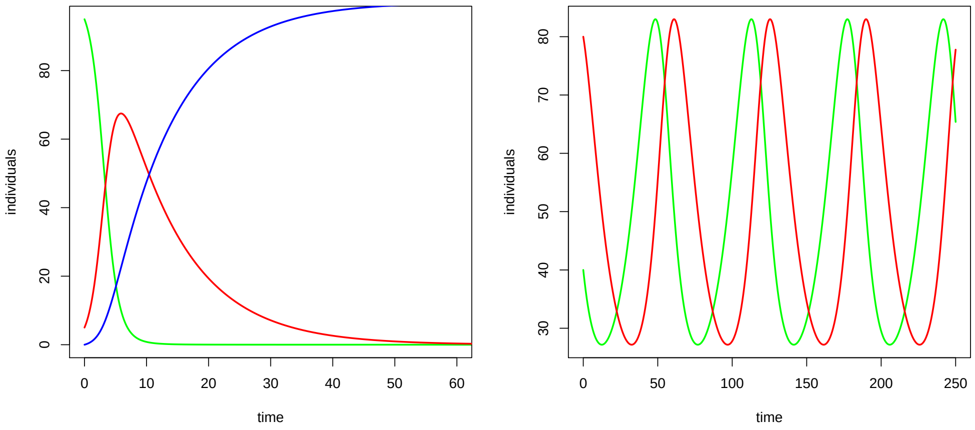
A typical simulation of this model (see Figure 7.17a) includes a rapid increase in infected individuals, which is then followed by their gradual recovery.

In this case study, we compare robustness analysis based on a formula containing value-freezing with respect to a freezing-free formula analysis exploiting a similar behavioural pattern. In particular, we consider the following formulae:

$$\text{STL} : \varphi_1 = \mathbf{F}_{[1,5]}(I \geq 50) \quad \text{STL*} : \varphi_2 = \mathbf{F}_{[1,5]}\left(I \geq 50 \wedge *\mathbf{G}_{[0.25,5]}(I^* \geq I)\right)$$

Both formulae require the number of infected individuals to be greater than 50 at some time in the interval $[1,5]$, while φ_2 also requires this number to be the local maximum (the number of infected individuals is required to decrease after reaching this maximum).

The robustness with respect to both properties was analysed on perturbations of both contact rate and recovery rate. Results are presented in Figure 7.18.



(a) (b)

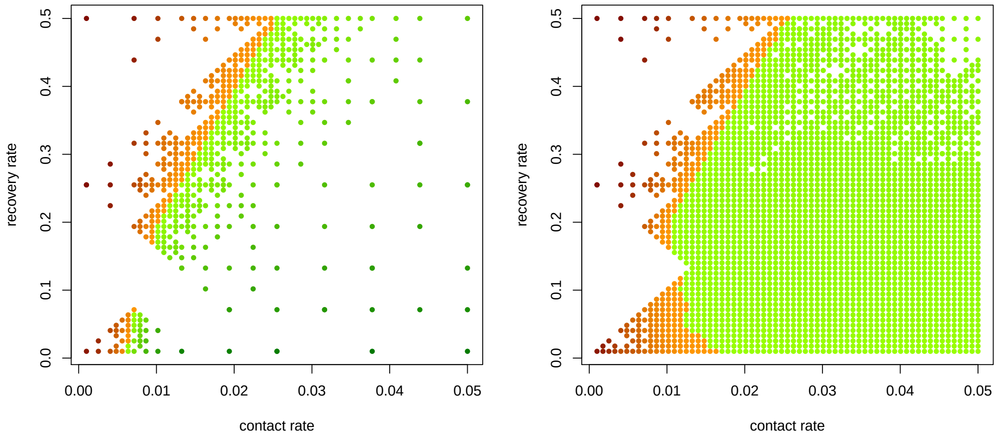
Figure 7.17: (a) Typical development of SIR model, showing the number of susceptible (green), infected (red) and recovered (blue) individuals. (b) Typical development of populations in predator-prey model, showing number of prey (green) and predator (red).

While the satisfaction sets of φ_1 and φ_2 (delineated by positive robustness) are essentially identical, the actual robustness values show a significant difference. Generally, when they are positive, the value of robustness with respect to φ_1 at given point is considerably greater than the corresponding value of robustness with respect to φ_2. In Figure 7.18, this can be seen as lighter shade of green points in Figure 7.18b. Also, lower robustness causes the apparent increase in the number of points.

The reason for the rapid change in robustness comes from evaluation of the subformula $*\mathbf{G}_{[0.25,5]}(I^* \geq I)$ that describes the local extreme. When evaluated in time t, robustness is proportional to the difference $(I[t] - I[t + 0.25])$ (by Definition 6.1). In practise, the difference is small provided that the descent of I is not extremely steep. This causes such formulae to have typically low robustness values on common signals.

7.6.2 Predator-Prey Model

In the second case study we analyse the predator-prey model [272, 360], which attains oscillating behaviour for a wide variety of parameters. We use a variant of the Lotka-Volterra model represented as a CRN in the fol-



(a) Robustness wrt φ_1.

(b) Robustness wrt φ_2.

Figure 7.18: Robustness of SIR model with respect to φ_1 and φ_2 for variable contact and recovery rates. Robustness was positive in green points and negative in orange points. Darker colour represents greater absolute value of robustness.

lowing way:

$$
\begin{aligned}
X &\xrightarrow{\nu} 2X \\
X + Y &\xrightarrow{\alpha} Y \\
Y &\xrightarrow{\gamma}
\end{aligned}
$$

The semantics of the respective cCRNM is represented by the following ordinary differential equations:

$$\frac{dX}{dt} = \nu X - \alpha XY \qquad\qquad \frac{dY}{dt} = \alpha XY - \gamma Y$$

The model simulates a situation where a prey species X is hunted by a predator species Y with the simplifying assumption that predator birth rate and prey death rate are equal and proportional to the probability of prey and predator contact, and thus to the product of both species populations. We use the following coefficients: prey natality (ν), predator mortality (γ) and predation rate (α). Typical behaviour of this models constitutes periodic oscillations (see Figure 7.17b).

We consider perturbation of two aforementioned coefficients, ν and α, and compute robustness with respect to two properties specified by the following formulae:

$$\begin{aligned}\psi_1 &= \mathbf{G}_{[0,300]} * \mathbf{F}_{[0,100]}\,(X \geq Y^*)\\ \psi_2 &= \mathbf{G}_{[0,300]}((X \geq 1 \wedge Y \geq 1 \wedge \mathbf{F}_{[0,50]} *(\mathbf{F}_{[0,75]}(X^* - X \geq 25)\\ &\quad \wedge \mathbf{F}_{[0,75]}(X - X^* \geq 25))))\end{aligned}$$

The property ψ_1 requires that for each time point $t \in [0, 300]$, there is a subsequent time point $t' \in [t, t + 100]$ such that population of prey in t' is greater than population of predators in t. According to Definition 6.1 its corresponding robustness can be expressed as follows:

$$\rho(\varphi, \mathfrak{x}) = \min_{t \in [0,300]} \max_{t' \in [t,t+100]} \frac{X[t'] - Y[t]}{2}$$

where $X[t']$ and $Y[t]$ denote values of s associated with given species at given time. The robustness value is maximized with respect to t' and minimized with respect to t, therefore, it uses maximal values of both X and Y. Consequently, this property can be interpreted as maximum population of prey being greater then maximum population of predators (restricted to given intervals).

Formula ψ_2 is based on the similar principle. While rejecting aberrant behaviour where population of one of the species drops below one individual, intuitively, it requires that there always is time in the future when population of prey can increase or decrease by 25 individuals, which is stated by the following subformula:

$$\mathbf{F}_{[0,50]} * \left(\mathbf{F}_{[0,75]}\,(X^* - X \geq 25) \wedge \mathbf{F}_{[0,75]}\,(X - X^* \geq 25)\right).$$

Therefore, ψ is satisfied when the difference between maximal and minimal prey population is greater than 50 and the associated robustness is proportional to this difference. Again, we have avoided use of the extreme property, which would adversely affect robustness value.

Results of this analysis are presented in Figure 7.19. Here, we should point out that small prey natality produced behaviour where predator population approached zero and period of oscillations was greatly increased. For such behaviour, intervals used in ψ_1 and ψ_2 were shorter than one period.

Apparently, satisfaction of ψ_1 is not affected by predation rate. More interestingly, when prey natality increases, predator population exceeds that of prey (see Figure 7.19 (top)). Figure 7.19 (bottom) shows that amplitude of prey population oscillation is affected by both prey natality and predation rate.

7.6.3 Performance

Performance of robustness analysis is summarized in Table 7.3. All results have been obtained by executing the algorithm implementation on a 4 core

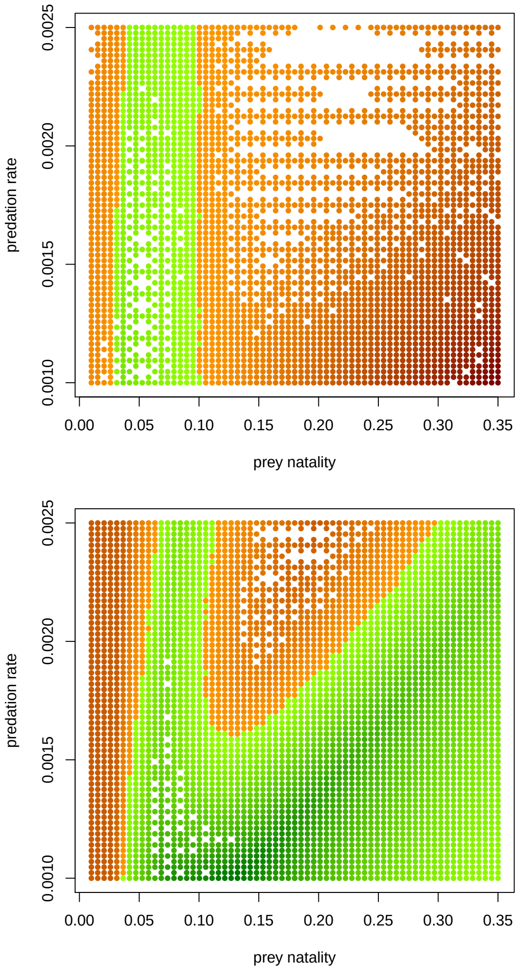


Figure 7.19: Robustness of predator-prey model with respect to ψ_1 (up) and ψ_2 (bottom) for variable prey natality and predation rate. Robustness was positive in green points and negative in orange points. Darker colour represents greater absolute value of robustness.

2 GHz CPU with 4 GB RAM. Each computation has been arranged into 8 threads. For each analysis we have set an optimal resolution of the trajectories (number of simulated points). The number of simulated trajectories has been bounded by the number of refinement iterations in the Parasim parameter space sampling procedure.

It is worth noting that all analysed properties consist only of $*\mathbf{F}$ and $*\mathbf{G}$ operators for which the procedure is optimized by employing Lemire queues in the same way as proposed in [149]. This is based on an optimal streaming algorithm for computing maxima (resp. minima) of a numerical sequence and allows to reduce the quadratic complexity wrt formula size to linear.

Property (model)	formula size	# signals	# points per a signal	time
φ_1 (SIR)	2	250	500	8.6 s
φ_2 (SIR)	6	1365	1000	15.2 s
ψ_1 (Predator-Prey)	4	831	400	85.4 s
ψ_2 (Predator-Prey)	12	1293	423	309.4 s

Table 7.3: Performance of the robustness computation measured on the prototype implementation.

The increase in computation time in the case of ψ_1 is caused by longer time intervals quantifying the temporal operators. Computation of the property ψ_2 has been slowed down due to insufficient memory.

7.7 Publications Summary

The case study presented in Section 7.2 has been published in [139]. The work has been initiated by Jiří Damborský (Faculty of Science, MU) and it makes a part of a larger synthetic biology project on which I have worked with a doctoral student Martin Demko. My contribution to the presented part of the work is in setting up the problem, formalising the workflow employing the parameter synthesis technology, adapting the techniques to the concrete problem, and assisting in individual steps of the underlying *in silico* experiments.

The problem studied in Section 7.3 has provided a continuous case study we have used for calibration of our techniques. The CTL-based results have been published in [82] (independent parameters) and [56] (dependent parameters). The CSL-based analysis has been published in [89]. My contribution is in setting up the problem, formalising the workflow employing the parameter synthesis/exploration technology, adapting the concrete problem to the technical framework, and guiding individual tasks of the conducted experiments.

The study presented in Section 7.4 has been published in [210]. The work makes a part of a larger systems biology project lead by Pavel Krejčí (Faculty of Medicine, MU) on which I have worked with a doctoral student Matej Hajnal. My contribution to the presented part of the work is in setting up the computational modelling problem, adapting the existing workflow to the settings of UCTL, and assisting in individual steps of the underlying *in silico* experiments.

The robustness analysis case study presented in Section 7.5 has been motivated by discussions with Ralf Steuer (Humboldt University Berlin) during the joint work on the CyanoTeam project. The work on the individual analysis tasks and results visualisation has been realised in collaboration with Milan Češka and Sven Dražan. My contribution is in setting up the problem, adapting the models to the settings of the used techniques, guiding individual steps of the underlying experiments, and interpreting the achieved results.

The analysis of population models described in Section 7.6 has been published in [90]. The work has been realised in collaboration with my master-level student Tomáš Vejpustek. My contribution is in setting up the problem and guiding individual steps of the underlying workflow.

7.8 Discussion

Most of the case studies are focused to application of a single selected technique targeting the nature of the model employed. However, the case study of G_1/S cell cycle transition (Section 7.3) shows a combination of techniques. This can be done due to the fact that the model structure is formulated by means of a CRN and therefore the two different semantics provided by a cCRNM and a sCRNM can be both used. It can be seen that the stochastic model analysis gives refined insight into the dynamics that capture the modelled decision at the level of an individual cell. This feature is not captured in the deterministic continuous model. However, the resulting behaviour of the transient distribution is in agreement with the deterministic settings. Additionally, it can be seen that the computational demands are much more on the side of the parameterised uniformisation in spite of the manual reductions performed on the model.

In general, the parameterised uniformisation method is much more demanding than the coloured model checking procedure. This is due to the fact the problem is enormously more complex in the amount of quantitative information that needs to be processed while meeting a required level of precision.

Both methods – the coloured model checking as well as parameterised uniformisation – require a non-trivial amount of work provided by the user manually. This is easily seen namely in the case studies provided on larger models where several model reduction steps had to be done.

Additionally, the CMC employed for the metabolic network in Section 7.2 has shown a disadvantage of not supporting observable external variables, the so-called assignments, that aggregate the values captured in a current state. This feature would allow direct modelling of (and reasoning about) the metabolic burden. In the case of parameterised uniformisation the power of bounded-time fragment CSL has been extended with

Case Study	Model	Logic	Problem	# param	Time
Lotka-Volterra	cCRNM	STL*	robustness	2	8-13 s (m-c)
SIR	cCRNM	STL*	robustness	2	2-5 min (m-c)
Sus/Trans Sig.	cINM	UCTL	synthesis	2	$<$ 5 min (m-c)
G_1/S	cRINM	CTL	synthesis	2	$<$ 8 min (m-c)
TCP-biodeg.	cRINM	ACTL	synthesis	3	$<$ 2 hrs (m-c)
G_1/S	sCRNM	CSL	exploration	2	4-8 hrs
elem. sig.	sCRNM	CSL+pp	robustness	2	$<$ 10 hrs

Table 7.4: Performance summary of the individual case studies. The times denoted ′m-c′ were achieved with implementations utilising multi-core computing. They reflect optimal settings of the parallelisation. CSL+pp denotes usage of CSL with post-processing functions. The numbers of parameters show the maximal numbers of unknown parameters considered in a single experiment.

the utility of post-processing functions. They seem to be very important for such kind of analysis since they support the possibility of aggregating and observing the information captured in states that was missing in the metabolic pathway case study.

The robustness analysis of population models described in Section 7.6 has the purpose of demonstrating the features of the method rather than inferring new insights at the level of population modelling. It can be seen that a region separating the satisfying parameter valuations from unsatisfying ones has been identified in both population models. This method did not require any significant intervention from the side of the user. However, the models are small in size and they do not incorporate significantly different time-scales for which numerical simulation might become challenging.

Performance summary where the individual case studies are ordered by the computation times is shown in Table 7.4. Time intervals cover the minimal and maximal times achieved during particular experiments with a given model (e.g., varying the analysed formula).

Finally, it is important to mention that the presented experiments do not cover usage of LTL-based methods. Several examples using the algorithms presented in Section 4.2 are described in [244] (for dINMs) and in [33] (for cRINMs and dINMs). The models considered there are comparable in complexity and size to models presented in this chapter. In particular, in [33] we have analysed the bistability of G_1/S model by using LTL formulae. It is apparent that the analysis of bistability with CTL as presented in Section 7.3 gives more precise results due to the fact that branching-time operators and state-based approach can capture the bistability phenomena quite well. This cannot be achieved with the path-centric view of LTL. The computation times achieved with LTL are comparable to the case of CTL.

Conclusion

A concrete summary and a discussion of possible future work have been included into every chapter of the previous text. In this section, however, we briefly summarise the contents of the work once again and we also sum up and generalise the possible directions for future research.

7.9 Summary

In this thesis, we have addressed the problems of parameter synthesis and robustness analysis from the perspective of methods based on temporal logics. The models that have been considered throughout the thesis make a selection that reflects well the current standards used in systems biology.

First, we have considered several variants of continuous-time deterministic models used to model biological phenomena at the macroscopic level where the average behaviour of the population of species (concentration of molecules) is considered as representative. Regarding the parameter synthesis, the first group of techniques we have provided for these models works at the level of a finite discrete abstraction of the continuous dynamics. The quality of the results obtained thus depends on the quality of the abstraction. At this place, the particular shape of the model can also affect the extent of spurious results caused by the abstraction. However, regardless the quality of abstraction, the method is suitable for global parameter synthesis as it allows to capture compactly the continuum of parameter values and to provide guaranteed results for synthesised sets at the level of piece-wise multi-affine representations of the continuous models. However, due to the highly over-approximating characteristics of the abstraction we have not found the abstraction practicable to be used as a based for robustness analysis.

To address the problem of robustness for cBNMs, we have exploited the method of analytical evaluation of the robustness measure on finitely sampled bounded signals obtained for example by numerical simulation. Our method employs the value-freezing extension of signal temporal logic thus lifting the utility of robustness analysis to complex phenomena that cannot be formulated in the plain version of the logic. An important example of such properties are various forms of oscillation that are very relevant in the settings of biological models. These results have a significant impact in the general field of cyber-physical systems and run-time verification.

Second, we have addressed continuous-time stochastic models that provide a detailed (mesoscopic) view of the molecular dynamics. In fact, the models at this level of abstraction fit well into the general framework of time-homogeneous continuous-time Markov processes. Being motivated by the inverse problem in biology in the context of these models, we have introduced a novel general method that is applicable to continuous-time Markov chains with parameterised transition rates. The method works with a limited fragment of the expressive continuous stochastic logic that suffices for typical cases of studying a given mechanism in such a detailed level. The quantitative nature of stochastic models providing probability as a well-defined measure has lead us to extend the method to robustness analysis of stochastic systems. Our implementation of robustness uses probability as the measure of systems performance quantifying a given analysed functionality.

Third, our method of parameter synthesis works well with discrete models of influence networks. Although in this thesis the main focus has been put to quantitative models, our work has shown that the technique of coloured model checking is highly-relevant for Boolean networks and other models fitting the framework of dINMs. These models are typically employed to reveal properties of interwoven non-linear interactions at a large-scale level. The challenge is thus to develop methods scalable for large models consisting of hundreds or thousands of species (e.g., *E. coli* consists of approximately 4500 different genes).

7.10 Future Work

Optimisation of Existing Techniques

An important task for future work is to improve the methods from two main perspectives. The first aspect is the functionality that has to give precise results or approximate results supplied with a numerically characterised error. The second aspect is the need to address the challenge of applicability of the methods in settings of large-scale models.

The first issue targets especially the framework of cBNMs where the quality of the obtained results strongly depends on settings of the approximation/abstraction method. The methods work quite well in cases of untimed analysis of restricted classes of systems that are very close to piece-wise multi-affine dynamics. However, models with complicated non-linear dynamics with multidimensional regulation functions can be highly distorted by using the concept of PWMA approximation. Moreover, the abstraction method applied to the PWMA does not work well for liveness properties such as oscillations where the abstracted (and highly-overapproximated) timing aspects play a crucial role.

The issue of applicability to large-scale problems addresses the entire spectrum of considered model classes. In the case of cBNMs, the need for explicit representation of the rectangular abstraction limits the applicability to larger models. To that end, techniques combining global abstraction methods with (relatively scalable) methods over-approximating the local behaviour (flow-pipes [109, 25]) can be potentially investigated for that purpose. Moreover, the methods of δ-decision algorithms [187, 157] that have been significantly improved very recently, have also a good potential to be combined with global model abstraction techniques.

The method of parameterised uniformisation follows the typical problems of formal analysis of continuous-time stochastic models regarding the scalability. This is a serious issue even in the non-parametric case. Although several improvements to the parameter uniformisation method have been introduced in the follow-up work [103] including its parallelisation [234], the method can be practically applied only for CTMCs with relatively low-dimensional spaces of unknown parameters with limited numbers of states. Combining the method with hybridisation and moment-closure methods addressed in Section 5.7 makes a research direction that has yet to be investigated.

A promising method to address the large-scale challenge for dINMs is static analysis. An important feature of static analysis is the ability to characterise behaviour of a complex network model just by looking into the network structure. We have already exploited several preliminary scenarios of static analysis that work well for reachability analysis of parameterised dINMs [248]. However, the challenge is to combine static analysis techniques with (coloured) model checking in order to bring better scalability to parameter synthesis methods by using the knowledge obtained from network structure.

Filling Existing Gaps

Several gaps remain regarding parameter synthesis and robustness analysis of models fitting in the modelling framework considered in this thesis. First, we have not exploited yet the LTL coloured model checking method with the SMT-encoded parameter sets. In spite of the fact that the method scales significantly better with interval-based encoding (as shown in Section 7.3.3), it is a valuable exercise to see how the explicit-state/symbolic-parameters representation method works with LTL. Second, the method of robustness analysis has not yet been considered in the settings of discrete abstraction of cBNMs. However, assuming there exists a quantitative measure of the error introduced during the approximation step, it would be interesting to incorporate such quantity into the systems states and paths and allow quantitative interpretation of temporal logics on discrete abstractions up-to the given error. Third, in spite of the powerful utility of symbolic model checking, it seems valuable to bring the CTL-based coloured model checking technique to dINMs. Especially this is promising in combination with the static analysis methods discussed above.

Another aspect to be considered for future work is an extension of the entire parameter synthesis framework to the settings of hybrid logics and algorithms utilising analysis of attractors in terms of strongly connected components. These techniques seem to be highly relevant for improving the methods of digital bifurcation analysis already targeted from the perspective of CTL-based model checking and discrete abstractions of cBNMs in [58]. These methods need to be brought into the settings of dINMs and even cRNMs where they are not yet established at all.

Software Development and Methods Applications

Finally, there is a lot of improvement to be done on the side of prototype implementations of the discussed techniques. A challenge is to create a unified framework for all the model classes discussed in this thesis. The ability to automatically transfer from high-level models such as dINMs to the quantitative setting of cINMs is an example of model refinement that is highly relevant in synthetic biology and molecular programming. Methods allowing that are extensively studied in other research.

A valuable feedback for software development comes from applications of the techniques in systems biology workflows. A grand challenge is to deploy the developed software to the online comprehensive modelling platform currently developed for cyanobacteria models [245, 353].

From the case studies presented in Chapter 7 it becomes evident that in most cases the methods and tools could not be performed fully automatically without any intervention from the user. For example, the approximation/abstraction procedures of cBNMs have to be fine-tuned to allow good and correct approximation of the original model. Another example can be seen in Sections 7.3.3 and 7.5 where manual reductions and simplifications of the models were needed in order to allow tractability of the algorithms. Development of methods that in some cases will allow to (semi-)automatise such model reduction tasks will contribute to getting back to the basic philosophy behind model checking – a push-button technology working without the need of manual tasks requiring detailed knowledge of the underlying technology.

Bibliography

[1] H. Abbas, A. Rodionova, E. Bartocci, S. A. Smolka, and R. Grosu. Quantitative regular expressions for arrhythmia detection algorithms. In *Computational Methods in Systems Biology (CMSB 2017)*, volume 10545 of *Lecture Notes in Computer Science*, pages 23–39. Springer, 2017.

[2] Z. Ahmed, D. Benque, S. Berezin, A. C. E. Dahl, J. Fisher, B. A. Hall, S. Ishtiaq, J. Nanavati, N. Piterman, M. Riechert, and N. Skoblov. Bringing ltl model checking to biologists. In *Verification, Model Checking, and Abstract Interpretation*, pages 1–13. Springer, 2017.

[3] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450, 2007.

[4] U. Alon, M. Surette, N. Barkai, and S. Leibler. Robustness in bacterial chemotaxis. *Nature*, 6715(397):168–171, 1999.

[5] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Hybrid Systems: Computation and Control (HSCC 2013)*, pages 173–182. ACM, 2013.

[6] M. Althoff and J. M. Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

[7] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense realtime. *Information and Computation*, 104:2–34, 1993.

[8] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

[9] R. Alur and T. A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–203, 1994.

[10] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[11] R. Alur, K. Mamouras, and D. Ulus. Derivatives of quantitative regular expressions. In *Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, volume 10460, pages 75–95. Springer, 2017.

[12] A. Andreychenko, L. Mikeev, D. Spieler, and V. Wolf. Parameter Identification for Markov Models of Biochemical Reactions. In *Computer Aided Verification (CAV 2011)*, Lecture Notes in Computer Science, pages 83–98. Springer, 2011.

[13] A. Andreychenko, L. Mikeev, and V. Wolf. Model reconstruction for moment-based stochastic chemical kinetics. *ACM Trans. Model. Comput. Simul.*, 25(2):12:1–12:19, 2015.

[14] Y. S. R. Annapureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan. S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 6605 of *Lecture Notes in Computer Science*, pages 254–257. Springer, 2011.

[15] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. *Cell Biochemistry and Biophysics*, 38:271–286, 2003.

[16] C. Areces and B. ten Cate. Hybrid logics. In *Handbook of Modal Logic*. Elsevier, 2007.

[17] G. Arellano, J. Argil, E. Azpeitia, M. Benítez, M. Carrillo, P. Góngora, D. A. Rosenblueth, and E. R. Alvarez-Buylla. "Antelope": a hybrid-logic model checker for branching-time Boolean GRN analysis. *BMC Bioinformatics*, 12(1):1–15, 2011.

[18] E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, 43:451–476, 2007.

[19] E. Asarin, A. Donzé, O. Maler, and D. Nickovic. Parametric identification of temporal properties. In *Runtime Verification*, pages 147–160. Springer, 2012.

[20] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Verifying continuous time Markov chains. In *Computer Aided Verification (CAV 1996)*, volume 1102 of *Lecture Notes in Computer Science*, pages 269–276. Springer, 1996.

[21] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli, and G. Franceschinis. The greatspn tool: recent enhancements. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):4–9, 2009.

[22] M. Backenköhler, L. Bortolussi, and V. Wolf. Moment-based parameter estimation for stochastic reaction networks in equilibrium. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(4):1180–1192, 2018.

[23] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model Checking Continuous-Time Markov Chains by Transient Analysis. In *Computer Aided Verification (CAV 2000)*, volume 1855 of *Lecture Notes in Computer Science*, pages 358–372. Springer, 2000.

[24] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[25] S. Bak and M. Caccamo. Computing reachability for nonlinear systems with hycreate, 2013. HSCC 2013. Poster Presentation.

[26] A. Bakhirkin, T. Ferrère, O. Maler, and D. Ulus. On the quantitative semantics of regular expressions over real-valued signals. In *Formal Modeling and Analysis of Timed Systems*, pages 189–206. Springer, 2017.

[27] P. Ballarini, M. Forlin, T. Mazza, and D. Prandi. Efficient parallel statistical model checking of biochemical networks. In *Parallel and Distributed Methods in verifiCation (PDMC)*, volume 14 of *Electronic Proceedings in Theoretical Computer Science*, pages 47–61, 2009.

[28] B. Barbot, M. Z. Kwiatkowska, A. Mereacre, and N. Paoletti. Estimation and verification of hybrid heart models for personalised medical and wearable devices. In *Computational Methods in Systems Biology (CMSB 2015)*, volume 9308 of *Lecture Notes in Computer Science*, pages 3–7. Springer, 2015.

[29] R. Barbuti, G. Caravagna, A. Maggiolo-Schettini, P. Milazzo, and S. Tini. Foundational aspects of multiscale modeling of biological systems with process algebras. *Theor. Comput. Sci.*, 431:96–116, 2012.

[30] N. Barkai and S. Leibler. Robustness in simple biochemical networks. *Nature*, (387):913–917, 1997.

[31] J. Barnat, N. Beneš, L. Brim, M. Demko, M. Hajnal, S. Pastva, and D. Šafránek. Detecting attractors in biological models with uncertain parameters. In *Computational Methods in Systems Biology (CMSB 2017)*, volume 10545 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2017.

[32] J. Barnat, L. Brim, and J. Chaloupka. Parallel breadth-first search ltl model-checking. In *Automated Software Engineering (ASE 2003)*, pages 106–115, 2003.

[33] J. Barnat, L. Brim, A. Krejčí, A. Streck, D. Šafránek, M. Vejnár, and T. Vejpustek. On Parameter Synthesis by Parallel Model Checking. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(3):693–705, 2012.

[34] J. Barnat, L. Brim, and P. Ročkai. Divine multi-core — a parallel ltl model-checker. In *Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis*, ATVA '08, pages 234–239. Springer, 2008.

[35] J. Barnat, L. Brim, I. Černá, S. Dražan, J. Fabriková, J. Láník, D. Šafránek, and H. Ma. BioDiVinE: A Framework for Parallel Analysis of Biological Models. In *Computational Models for Cell Processes (COMPMOD)*, volume 6 of *Electronic Proceedings in Theoretical Computer Science*, pages 31–45, 2009.

[36] J. Barnat, L. Brim, I. Černá, P. Moravec, P. Ročkai, and P. Šimeček. DiVinE – A Tool for Distributed Verification. In *Computer Aided Verification (CAV 2006)*, volume 4144 of *Lecture Notes in Computer Science*, pages 278–281. Springer, 2006.

[37] J. Barnat, L. Brim, and D. Šafránek. High-Performance Analysis of Biological Systems Dynamics with the DiVinE Model Checker. *Briefings in Bioinformatics*, 11(3):301–312, 2010.

[38] J. Barnat, L. Brim, D. Šafránek, and M. Vejnar. Parameter scanning by parallel model checking with applications in systems biology. In *Parallel and Distributed Methods in Verification, 2010 Ninth International Workshop on, and High Performance Computational Systems Biology, Second International Workshop on*, pages 95–104. IEEE, 2010.

[39] C. Barrett, P. Fontaine, and C. Tinelli. The SMT-LIB Standard: Version 2.5. Technical report, Department of Computer Science, The University of Iowa, 2015.

[40] E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti. On the Robustness of Temporal Properties for Stochastic Models. *ArXiv e-prints*, 2013.

[41] E. Bartocci, L. Bortolussi, and G. Sanguinetti. Data-driven statistical learning of temporal logic properties. In *Formal Modeling and Analysis of Timed Systems*, pages 23–37. Springer, 2014.

[42] E. Bartocci, F. Corradini, M. R. Di Berardini, E. Merelli, and L. Tesei. Shape calculus. a spatial mobile calculus for 3d shapes. *Scientific Annals of Computer Science*, 20:1, 2010.

[43] E. Bartocci, F. Corradini, E. Merelli, and L. Tesei. Detecting synchronisation of biological oscillators by model checking. *Theoretical Computer Science*, 411(20):1999 – 2018, 2010.

[44] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan. *Specification-Based Monitoring of Cyber-Physical Systems: A Survey on Theory, Tools and Applications*, pages 135–175. Springer, 2018.

[45] E. Bartocci and P. Lio. Computational modeling, formal analysis, and tools for systems biology. *PLOS Computational Biology*, 12(1):1–22, 2016.

[46] E. Bartocci, P. Liò, E. Merelli, and N. Paoletti. Multiple verification in complex biological systems: The bone remodelling case study. In *Transactions on Computational Systems Biology XIV*, pages 53–76. Springer, 2012.

[47] E. Batchelor and M. Goulian. Robustness and the cycle of phosphorylation and dephosphorylation in a two-component regulatory system. *Proceedings of the National Academy of Sciences*, 100(2):691–696, 2003.

[48] D. G. Bates and C. Cosentino. Validation and invalidation of systems biology models using robustness analysis. *IET Systems Biology*, 5(4):229–244, 2011.

[49] G. Batt, C. Belta, and R. Weiss. Model checking genetic regulatory networks with parameter uncertainty. In *Hybrid Systems: Computation and Control (HSCC)*, volume 4416 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2007.

[50] G. Batt, D. Bergamini, H. de Jong, H. Garavel, and R. Mateescu. Model checking genetic regulatory networks using gna and cadp. In *Model Checking Software*, pages 158–163. Springer, 2004.

[51] G. Batt, M. Page, I. Cantone, G. Gössler, P. Monteiro, and H. de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):603–610, 2010.

[52] 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. *Bioinformatics*, 21(suppl_1):i19–i28, 2005.

[53] G. Batt, R. B. Salah, and O. Maler. On timed models of gene networks. In *Formal Modeling and Analysis of Timed Systems (FORMATS)*, Lecture Notes in Computer Science, pages 38–52. Springer, 2007.

[54] G. Batt, B. Yordanov, R. Weiss, and C. Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–2422, 2007.

[55] C. Belta and L. Habets. Controlling a class of nonlinear systems on rectangles. *IEEE Transactions on Automatic Control*, 51(11):1749–1759, 2006.

[56] N. Beneš, L. Brim, M. Demko, S. Pastva, and D. Šafránek. Parallel SMT-based parameter synthesis with application to piecewise multi-affine systems. In *ATVA*, volume 9936 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2016.

[57] N. Beneš, L. Brim, M. Demko, S. Pastva, and D. Šafránek. Pithya: A parallel tool for parameter synthesis of piecewise multi-affine dynamical systems. In *Computer Aided Verification (CAV 2017)*, pages 591–598. Springer, 2017.

[58] N. Beneš, L. Brim, M. Demko, S. Pastva, and D. Šafránek. A model checking approach to discrete bifurcation analysis. In *FM 2016*, volume 9995 of *Lecture Notes in Computer Science*, pages 85–101. Springer, 2016.

[59] D. Benque, S. Bourton, C. Cockerton, B. Cook, J. Fisher, S. Ishtiaq, N. Piterman, A. Taylor, and M. Y. Vardi. Bma: Visual tool for modeling and analyzing biological networks. In *Computer Aided Verification (CAV 2012)*, pages 686–692. Springer, 2012.

[60] S. Berman, A. Halász, and V. Kumar. MARCO: a reachability algorithm for multi-affine systems with applications to biological systems. In *Hybrid Systems: Computation and Control (HSCC 2007)*, Lecture Notes in Computer Science, pages 76–89. Springer, 2007.

[61] F. Bernardini, C. Biggs, J. Derrick, M. Gheorghe, M. Niranjan, and G. Sanguinetti. Parameter Estimation and Model Checking in a Model of Prokaryotic Autoregulation. Technical report, University of Sheffield, 2007.

[62] 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.

[63] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004.

[64] R. Blossey, L. Cardelli, and A. Phillips. A compositional approach to the stochastic dynamics of gene networks. In *Transactions on Computational Systems Biology IV*, pages 99–122. Springer, 2006.

[65] E. Boczko, W. D. Kalies, and K. Mischaikow. Polygonal approximation of flows. *Topology and its Applications*, 154(13):2501 – 2520, 2007.

[66] S. Bogomolov, M. Giacobbe, T. A. Henzinger, and H. Kong. Conic abstractions for hybrid systems. In *FORMATS 2017*, pages 116–132. Springer, 2017.

[67] S. Bogomolov, C. Schilling, E. Bartocci, G. Batt, H. Kong, and R. Grosu. Abstraction-based parameter synthesis for multiaffine systems. In *Hardware and Software: Verification and Testing*, volume 9434 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2015.

[68] G. Bombara, C.-I. Vasile, F. Penedo, H. Yasuoka, and C. Belta. A decision tree approach to data classification using signal temporal logic. In *Hybrid Systems: Computation and Control (HSCC 2016)*, HSCC 2016, pages 1–10. ACM, 2016.

[69] B. Bonet, P. Haslum, S. Hickmott, and S. Thiébaux. *Directed Unfolding of Petri Nets*, pages 172–198. Springer, 2008.

[70] N. Bonzanni, E. Krepska, K. A. Feenstra, W. Fokkink, T. Kielmann, H. E. Bal, and J. Heringa. Executing multicellular differentiation: quantitative predictive modelling of *C.elegans* vulval development. *Bioinformatics*, 25(16):2049–2056, 2009.

[71] L. Bortolussi, L. Cardelli, M. Kwiatkowska, and L. Laurenti. Approximation of probabilistic reachability for chemical reaction networks using the linear noise approximation. In *Quantitative Evaluation of Systems (QEST 2016)*, volume 9826, pages 72–88. Springer, 2016.

[72] L. Bortolussi, D. Milios, and G. Sanguinetti. U-check: Model checking and parameter synthesis under uncertainty. In *Quantitative Evaluation of Systems (QEST 2015)*, pages 89–104. Springer, 2015.

[73] L. Bortolussi, D. Milios, and G. Sanguinetti. Smoothed model checking for uncertain continuous-time markov chains. *Information and Computation*, 247:235 – 253, 2016.

[74] L. Bortolussi and R. Paškauskas. Mean-field approximation and quasi-equilibrium reduction of markov population models. In *Quantitative Evaluation of Systems (QEST 2014)*, volume 8657 of *Lecture Notes in Computer Science*, pages 106–121. Springer, 2014.

[75] L. Bortolussi and A. Policriti. Hybrid systems and biology. In *Formal Methods for Computational Systems Biology, 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008*, Lecture Notes in Computer Science, pages 424–448. Springer, 2008.

[76] L. Bortolussi and A. Policriti. Modeling biological systems in stochastic concurrent constraint programming. *Constraints*, 13(1-2):66–90, 2008.

[77] D. Bosnacki, H. M. M. ten Eikelder, M. N. Steijaert, and E. P. de Vink. Stochastic analysis of amino acid substitution in protein synthesis. In *Computational Methods in Systems Biology (CMSB 2008)*, pages 367–386, 2008.

[78] P. Boutillier, M. Maasha, X. Li, H. F. Medina-Abarca, J. Krivine, J. Feret, I. Cristescu, A. G. Forbes, and W. Fontana. The kappa platform for rule-based modeling. *Bioinformatics*, 34(13):i583–i592, 2018.

[79] L. Brim, N. Beneš, D. J., S. Pastva, and D. Šafránek. Facetal abstraction of non-linear dynamical systems, 2018. Submitted.

[80] L. Brim, I. Černá, P. Krčál, and R. Pelánek. Distributed ltl model checking based on negative cycle detection. In *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science*, pages 96–107. Springer, 2001.

[81] L. Brim, I. Černá, P. Moravec, and J. Šimša. Accepting predecessors are better than back edges in distributed LTL model-checking. In *Formal Methods in Computer Aided Design (FMCAD)*, volume 4144 of *Lecture Notes in Computer Science*, pages 352–366. Springer, 2004.

[82] L. Brim, M. Češka, M. Demko, S. Pastva, and D. Šafránek. Parameter synthesis by parallel coloured CTL model checking. In *CMSB*, volume 9308 of *Lecture Notes in Computer Science*, pages 251–263. Springer, 2015.

[83] L. Brim, M. Češka, S. Dražan, and D. Šafránek. On Robustness Analysis of Stochastic Biochemical Systems by Probabilistic Model Checking. *ArXiv e-prints*, 2013.

[84] L. Brim, M. Češka, and D. Šafránek. Model checking of biological systems. In *Formal Methods for Dynamical Systems*, pages 63–112. Springer, 2013.

[85] L. Brim, M. Demko, S. Pastva, and D. Šafránek. High-performance discrete bifurcation analysis for piecewise-affine dynamical systems. In *Hybrid Systems Biology*, pages 58–74. Springer, 2015.

[86] L. Brim, P. Dluhoš, D. Šafránek, and T. Vejpustek. STL*: Extending signal temporal logic with signal-value freezing operator. *Information and Computation*, 236:52–67, 2014.

[87] L. Brim, J. Fabriková, S. Drazan, and D. Safránek. On approximative reachability analysis of biochemical dynamical systems. *Trans. Computational Systems Biology*, 7625(14):77–101, 2012.

[88] L. Brim, J. Nižnan, and D. Šafránek. Compact representation of photosynthesis dynamics by rule-based models. *Electronic Notes in Theoretical Computer Science*, 316:17–27, 2015.

[89] L. Brim, M. Češka, S. Dražan, and D. Šafránek. Exploring parameter space of stochastic biochemical systems using quantitative model checking. In *Computer Aided Verification (CAV 2013)*, volume 8044 of *Lecture Notes in Computer Science*, pages 107–123. Springer, 2013.

[90] L. Brim, T. Vejpustek, D. Šafránek, and J. Fabriková. Robustness analysis for value-freezing signal temporal logic. *arXiv preprint arXiv:1309.0867*, 2013.

[91] L. Brim, K. Yorav, and J. Žídková. Assumption-based distribution of ctl model checking. *International Journal on Software Tools for Technology Transfer*, 7(1):61–73, 2005.

[92] M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using continuous time markov chains. In *Transactions on Computational Systems Biology VI*, volume 4220 of *Lecture Notes in Computer Science*, pages 44–67. Springer, 2006.

[93] L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman. Machine learning biochemical networks from temporal logic properties. In *Transactions on Computational Systems Biology VI*, Lecture Notes in Computer Science, pages 68–94. Springer, 2006.

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

[95] D. Campagna and C. Piazza. Hybrid automata in systems biology: How far can we go? In *From Biology to Concurrency and Back (FBTC)*, volume 229 of *Electronic Notes in Theoretical Computer Science*, pages 93 – 108, 2009.

[96] F. Camporesi, J. Feret, and K. Q. Lý. Kade: A tool to compile kappa rules into (reduced) ODE models. In *Computational Methods in Systems Biology (CMSB 2017)*, volume 10545 of *Lecture Notes in Computer Science*, pages 291–299. Springer, 2017.

[97] G. Caravagna and J. Hillston. Modeling biological systems with delays in Bio-PEPA. In *Proceedings Fourth Workshop on Membrane Computing and Biologically Inspired Process Calculi 2010*, volume 40 of *Electronic Proceedings in Theoretical Computer Science*, pages 85–101, 2010.

[98] L. Cardelli. Brane calculi. In *Computational Methods in Systems Biology (CMSB 2005)*, pages 257–278. Springer, 2005.

[99] L. Cardelli. Morphisms of reaction networks that couple structure to function. *BMC Systems Biology*, 8(1):84, 2014.

[100] L. Cardelli, M. Češka, M. Fränzle, M. Kwiatkowska, L. Laurenti, N. Paoletti, and M. Whitby. *Syntax-Guided Optimal Synthesis for Chemical Reaction Networks*, pages 375–395. Springer, 2017.

[101] R. Carter and E. M. Navarro-López. Dynamically-driven timed automaton abstractions for proving liveness of continuous systems. In *FORMATS 2012*, volume 7595 of *Lecture Notes in Computer Science*, pages 59–74. Springer, 2012.

[102] A. Casagrande and C. Piazza. Unwinding biological systems. *Theoretical Computer Science*, 587:26 – 48, 2015.

[103] M. Češka, P. Pilař, N. Paoletti, L. Brim, and M. Kwiatkowska. Prism-psy: Precise gpu-accelerated parameter synthesis for stochastic systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 367–384. Springer, 2016.

[104] A. Champneys and K. Tsaneva-Atanasova. *Dynamical Systems Theory, Bifurcation Analysis*, pages 632–637. Springer New York, 2013.

[105] C. Chaouiya. Petri net modelling of biological networks. *Briefings in Bioinformatics*, 8(4):210–219, 2007.

[106] C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs: A computational tool based on a discrete formal framework. In *Positive Systems*, volume 294 of *LNCIS*, pages 830–832. Springer, 2003.

[107] T. Chatain, S. Haar, L. Jezequel, L. Paulevé, and S. Schwoon. Characterization of reachable attractors using petri net unfoldings. In *Computational Methods in Systems Biology (CMSB 2014)*, volume 8859 of *Lecture Notes in Computer Science*, pages 129–142. Springer, 2014.

[108] T. Chatain, S. Haar, and L. Paulevé. Boolean networks: Beyond generalized asynchronicity. In *Cellular Automata and Discrete Complex Systems*, pages 29–42. Springer, 2018.

[109] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Computer Aided Verification (CAV 2013)*, volume 8044 of *Lecture Notes in Computer Science*, pages 258–263. Springer, 2013.

[110] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 2003.

[111] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV 2: An Open-Source Tool for Symbolic Model Checking. In *Computer Aided Verification (CAV 2002)*, volume 2404 of *Lecture Notes in Computer Science*, pages 359–364. Springer, 2002.

[112] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. NuSMV: a new symbolic model checker. *J. Softw. Tools Technol. Transf.*, 2(410–425), 2000.

[113] F. Ciocchetta, S. Gilmore, M. L. Guerriero, and J. Hillston. Integrated simulation and model-checking for the analysis of biochemical systems. *Electronic Notes in Theoretical Computer Science*, 232:17–38, 2009.

[114] F. Ciocchetta and J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theor. Comput. Sci.*, 410(33-34):3065–3084, 2009.

[115] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2000.

[116] E. Clarke and P. Zuliani. Statistical Model Checking for Cyber-Physical Systems. In *Automated Technology for Verification and Analysis (ATVA)*, volume 6996 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2011.

[117] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, 1986.

[118] F. H. Clarke. *Optimization and Nonsmooth Analysis*. S.I.A.M., 1983.

[119] P. Collins and A. Goldsztejn. The reach-and-evolve algorithm for reachability analysis of nonlinear dynamical systems. *Electronic Notes in Theoretical Computer Science*, 223:87 – 102, 2008.

[120] P. Collins, L. C. Habets, J. H. van Schuppen, I. Černá, J. Fabriková, and D. Šafránek. Abstraction of biochemical reaction systems on polytopes. In *Proceedings of the 18th IFAC World Congress*, volume 18, pages 14869–14875, 2011.

[121] G. Craciun, Y. Tang, and M. Feinberg. Understanding bistability in complex enzyme-driven reaction networks. *Proceedings of the National Academy of Sciences*, 103(23):8697–8702, 2006.

[122] A. Crudu, A. Debussche, and O. Radulescu. Hybrid stochastic simplifications for multiscale gene networks. *BMC Systems Biology*, 3(1):89, 2009.

[123] P. Cull. Difference equations as biological models. *Sci. Math. Jpn.*, (2):217–233, 2006.

[124] B. Daigle, M. Roh, L. Petzold, and J. Niemi. Accelerated Maximum Likelihood Parameter Estimation for Stochastic Biochemical Systems. *BMC Bioinformatics*, 13(1):68–71, 2012.

[125] A. E. Dalsgaard, A. Laarman, K. G. Larsen, M. C. Olesen, and J. van de Pol. Multi-core reachability for timed automata. In *Formal Modeling and Analysis of Timed Systems*, pages 91–106. Springer, 2012.

[126] T. Dang, A. Donzé, O. Maler, and N. Shalev. Sensitive state-space exploration. In *IEEE Conference on Decision and Control*, pages 4049–4054, 2008.

[127] T. Dang, T. Dreossi, and C. Piazza. Parameter synthesis through temporal logic specifications. In *FM 2015: Formal Methods - 20th International Symposium, Oslo, Norway, June 24-26, 2015, Proceedings*, volume 9109 of *Lecture Notes in Computer Science*, pages 213–230. Springer, 2015.

[128] T. Dang, C. L. Guernic, and O. Maler. Computing reachable states for nonlinear biological models. *Theor. Comput. Sci.*, 412(21):2095–2107, 2011.

[129] T. Dang, C. Le Guernic, and O. Maler. Computing reachable states for nonlinear biological models. In *CMSB 2009*, volume 5688 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2009.

[130] V. Danos and C. Laneve. Formal molecular biology. *Theor. Comput. Sci.*, 325(1):69–110, 2004.

[131] R. Darling and J. Norris. Differential equation approximations for markov chains. *Probab. Surveys*, 5:37–79, 2008.

[132] A. Darlington, J. Kim, and D. Bates. Robustness analysis of a synthetic translational resource allocation controller. *IEEE Control Systems Letters*, August 2018.

[133] A. David, D. Du, K. G. Larsen, A. Legay, M. Mikucionis, D. B. Poulsen, and S. Sedwards. Statistical model checking for stochastic hybrid systems. In *Hybrid Systems and Biology (HSB 2012)*, volume 92 of *Electronic Proceedings in Theoretical Computer Science*, pages 122–136, 2012.

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

[135] H. de Jong, J. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulations of genetic regulatory networks using piecewise linear models. *Bull. Math. Biol.*, 66:301–340, 2004.

[136] T. Děd, D. Šafránek, M. Troják, M. Klement, J. Šalagovič, and L. Brim. Formal biochemical space with semantics in kappa and bngl. *Electronic Notes in Theoretical Computer Science*, 326:27–49, 2016.

[137] C. Dehnert, S. Junges, J.-P. Katoen, and M. Volk. A storm is coming: A modern probabilistic model checker. In *Computer Aided Verification (CAV 2017)*, pages 592–600. Springer, 2017.

[138] L. Dematté, C. Priami, and A. Romanel. The beta workbench: a computational tool to study the dynamics of biological systems. *Briefings in Bioinformatics*, 9(5):437–449, 2008.

[139] M. Demko, N. Beneš, L. Brim, S. Pastva, and D. Šafránek. High-performance symbolic parameter synthesis of biological models: A case study. In *CMSB 2016*, volume 9859 of *Lecture Notes in Computer Science*, pages 82–97. Springer, 2016.

[140] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia. Robust online monitoring of signal temporal logic. In *Runtime Verification*, pages 55–70. Springer, 2015.

[141] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia. Robust online monitoring of signal temporal logic. *Formal Methods in System Design*, 51(1):5–30, 2017.

[142] F. Didier, T. A. Henzinger, M. Mateescu, and V. Wolf. Fast Adaptive Uniformization for the Chemical Master Equation. In *Parallel and Distributed Methods in Verification and High Performance Computational Systems Biology (HiBi/PDMC 2009)*, pages 118–127. IEEE Computer Society, 2009.

[143] P. Dluhoš, L. Brim, and D. Šafránek. On expressing and monitoring oscillatory dynamics. In *Hybrid Systems and Biology (HSB 2012)*, volume 92 of *Electronic Proceedings in Theoretical Computer Science*, pages 73–87, 2012.

[144] A. Doi, S. Fujita, H. Matsuno, M. Nagasaki, and S. Miyano. Constructing Biological Pathway Models with Hybrid Functional Petri Nets. *In Silico Biology*, 4(3):271–291, 2004.

[145] A. Donzé. Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems. In *Computer Aided Verification (CAV 2010)*, volume 6174 of *Lecture Notes in Computer Science*, pages 167–170. Springer, 2010.

[146] A. Donzé, G. Clermont, and C. Langmead. Parameter synthesis in nonlinear dynamical systems: Application to systems biology. *Journal of Computational Biology*, 17(3):325–336, 2010.

[147] A. Donzé, G. Clermont, A. Legay, and C. J. Langmead. Parameter synthesis in nonlinear dynamical systems: Application to systems biology. In *Research in Computational Molecular Biology*, pages 155–169. Springer, 2009.

[148] A. Donzé, E. Fanchon, L. M. Gattepaille, O. Maler, and P. Tracqui. Robustness analysis and behavior discrimination in enzymatic reaction networks. *PLoS ONE*, 6(9):e24246, 2011.

[149] A. Donzé, T. Ferrère, and O. Maler. Efficient robust monitoring for stl. In *Computer Aided Verification (CAV 2013)*, volume 8044 of *Lecture Notes in Computer Science*, pages 264–279. Springer, 2013.

[150] A. Donzé and O. Maler. Systematic simulation using sensitivity analysis. In *Hybrid Systems: Computation and Control*, pages 174–189. Springer, 2007.

[151] A. Donzé and O. Maler. Robust satisfaction of temporal logic over real-valued signals. In *Formal Modeling and Analysis of Timed Systems*, volume 6246 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2010.

[152] T. Dreossi. Sapo: Reachability computation and parameter synthesis of polynomial dynamical systems. In *Hybrid Systems: Computation and Control (HSCC 2017)*, pages 29–34. ACM, 2017.

[153] T. Dreossi, C. Piazza, and T. Dang. *Reachability Computation and Parameter Synthesis for Polynomial Dynamical Systems*. PhD thesis, PhD thesis, 2016.

[154] D. Drusinsky. Monitoring temporal rules combined with time series. In *Computer Aided Verification (CAV 2003)*, pages 114–117. Springer, 2003.

[155] P. Dvořák. *Engineering of the synthetic metabolic pathway for biodegradation of environmental pollutant*. PhD thesis, Masaryk University, 2014.

[156] A. Eggers, M. Fränzle, and C. Herde. SAT Modulo ODE: A Direct SAT Approach to Hybrid Systems. In *ATVA 2008*, pages 171–185. Springer, 2008.

[157] A. Eggers, N. Ramdani, N. S. Nedialkov, and M. Fränzle. Improving the SAT modulo ODE approach to hybrid systems analysis by combining different enclosure methods. *Software and System Modeling*, 14(1):121–148, 2015.

[158] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway logic: Symbolic analysis of biological signaling. In *Pacific Symposium on Biocomputing*, pages 400–412, 2002.

[159] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie. Stochastic Modelling of Gene Regulatory Networks. *Int. J. of Robust and Nonlinear Control*, 15(15):691–711, 2005.

[160] H. W. Engl, C. Flamm, P. Kügler, J. Lu, S. Müller, and P. Schuster. Inverse problems in systems biology. *Inverse Problems*, 25(12):123014, 2009.

[161] J. Faeder, M. Blinov, and W. Hlavacek. Rule-based modeling of biochemical systems with bionetgen. *Methods Mol Biol.*, 500:113–167, 2009.

[162] F. Fages and A. Rizk. On the analysis of numerical data time series in temporal logic. In *Computational Methods in Systems Biology (CMSB 2007)*, pages 48–63. Springer, 2007.

[163] F. Fages and A. Rizk. On temporal logic constraint solving for analyzing numerical data time series. *Theor. Comput. Sci.*, 408(1):55–65, 2008.

[164] F. Fages and A. Rizk. From model-checking to temporal logic constraint solving. In *Principles and Practice of Constraint Programming - CP 2009*, pages 319–334. Springer, 2009.

[165] F. Fages and S. Soliman. Formal cell biology in Biocham. In *8th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM08*, volume 5016, pages 54–80, 2008.

[166] F. Fages and S. Soliman. On robustness computation and optimization in biocham-4. In *Computational Methods in Systems Biology (CMSB 2018)*, pages 292–299. Springer, 2018.

[167] F. Fages, S. Soliman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine biocham. *Journal of Biological Physics and Chemistry*, 4(2):64–73, 2004.

[168] G. Fainekos and G. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[169] M. Feinberg and F. J. Horn. Dynamics of open chemical systems and the algebraic structure of the underlying reaction network. *Chemical Engineering Science*, 29(3):775 – 787, 1974.

[170] J. Feret and K. Q. Ly. Local traces: An over-approximation of the behavior of the proteins in rule-based models. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 15(4):1124–1137, 2018.

[171] J. Feret and K. Q. Lý. Reachability analysis via orthogonal sets of patterns. *Electr. Notes Theor. Comput. Sci.*, 335:27–48, 2018.

[172] J. Fisher, D. Harel, and T. A. Henzinger. Biology as reactivity. *Communications of the ACM*, 54(10):72–82, 2011.

[173] J. Fisher and T. Henzinger. Executable cell biology. *Nature biotechnology*, 25(11):1239–1249, 2007.

[174] J. Fisher and N. Piterman. *Model Checking in Biology*, pages 255–279. Springer, 2014.

[175] L. F. Fitime, O. Roux, C. Guziolowski, and L. Paulevé. Identification of bifurcation transitions in biological regulatory networks using answer-set programming. *Algorithms Mol Biol.*, 12(19), 2017.

[176] B. L. Fox and P. W. Glynn. Computing Poisson Probabilities. *CACM*, 31(4):440–445, 1988.

[177] G. Frehse, S. Jha, and B. Krogh. A counterexample-guided approach to parameter synthesis for linear hybrid automata. In *Hybrid Systems: Computation and Control (HSCC 2008)*, volume 4981 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2008.

[178] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable Verification of Hybrid Systems. In *Computer Aided Verification (CAV 2011)*, pages 379–395. Springer, 2011.

[179] F. Fröhlich, F. Theis, and J. Hasenauer. Uncertainty analysis for non-identifiable dynamical systems: Profile likelihoods, bootstrapping and more. In *CMBS*, volume 8859 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 2014.

[180] F. Frölich, P. Thomas, A. Kazeroonian, F. J. Theis, R. Grima, and J. Hasenauer. Inference for stochastic chemical kinetics using moment equations and system size expansion. *PLoS Computational Biology*, 12:1–28, 2016.

[181] J. Fromentin, D. Eveillard, and O. Roux. Hybrid modeling of biological networks: mixing temporal and qualitative biological properties. *BMC Systems Biology*, 4(1):79, 2010.

[182] A. Funahashi, M. Morohashi, H. Kitano, and N. Tanimura. Celldesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1(5):159 – 162, 2003.

[183] A. Gábor and J. R. Banga. Improved parameter estimation in kinetic models: Selection and tuning of regularization methods. In *CMSB 2014*, volume 8859 of *Lecture Notes in Computer Science*, pages 45–60. Springer, 2014.

[184] E. Gallet, M. Manceny, P. Le Gall, and P. Ballarini. *Formal Methods and Software Engineering: 16th International Conference on Formal Engineering Methods, ICFEM 2014, Luxembourg, Luxembourg, November 3-5, 2014. Proceedings*, chapter An LTL Model Checking Approach for Biological Parameter Inference, pages 155–170. Springer, 2014.

[185] V. Galpin, J. Hillston, and L. Bortolussi. HYPE Applied to the Modelling of Hybrid Biological Systems. *Electronic Notes in Theoretical Computer Science*, 218:33–51, 2008.

[186] S. Gao, J. Avigad, and E. M. Clarke. δ-complete decision procedures for satisfiabilityoverthe reals. In *Automated Reasoning*, pages 286–300. Springer, 2012.

[187] S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *CADE-24*, volume 7898 of *Lecture Notes in Computer Science*, pages 208–214. Springer, 2013.

[188] H. Garavel, F. Lang, and R. Mateescu. CADP 2006: A toolbox for the construction and analysis of distributed processes. In *Computer Aided Verification (CAV 2007)*, volume 4590 of *Lecture Notes in Computer Science*, pages 153–163. Springer, 2007.

[189] H. Garnier and L. Wang. *Identification of Continuous-time Models from Sampled Data*. Springer, 1st edition, 2008.

[190] S. Gay, F. Fages, T. Martinez, S. Soliman, and C. Solnon. On the subgraph epimorphism problem. *Discrete Applied Mathematics*, 162:214–228, 2014.

[191] S. Gay, S. Soliman, and F. Fages. A graphical method for reducing and relating models in systems biology. *Bioinformatics*, 26(18):i575–i581, 2010.

[192] M. Giacobbe, C. C. Guet, A. Gupta, T. A. Henzinger, T. Paixão, and T. Petrov. Model checking the evolution of gene regulatory networks. *Acta Inf.*, 54(8):765–787, 2017.

[193] D. Gilbert, R. Breitling, M. Heiner, and R. Donaldson. An introduction to biomodel engineering, illustrated for signal transduction pathways. In *Membrane Computing*, volume 5391 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2009.

[194] D. Gilbert, M. Heiner, F. Liu, and N. Saunders. Colouring space - a coloured framework for spatial modelling in systems biology. In *Application and Theory of Petri Nets and Concurrency*, pages 230–249. Springer, 2013.

[195] D. T. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. *Journal of Physical Chemistry*, 81(25):2340–2381, 1977.

[196] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(13):404 – 425, 1992.

[197] D. T. Gillespie. Stochastic Simulation of Chemical Kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, 2007.

[198] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control (HSCC 2006)*, volume 3927 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2006.

[199] S. V. Goethem, J.-M. Jacquet, L. Brim, and D. Šafránek. Timed modelling of gene networks with arbitrary expression level discretization. In *Interactions between Computer Science and Biology*, Electronic Notes in Theoretical Computer Science. Elsevier, 2013. in press.

[200] A. Golightly and D. J. Wilkinson. Bayesian Parameter Inference for Stochastic Biochemical Network Models Using Particle Markov Chain Monte Carlo. *Interface Focus*, 1(6):807–820, 2011.

[201] W. Grassmann. Transient Solutions in Markovian Queueing Systems. *Computers & Operations Research*, 4(1):47 – 53, 1977.

[202] D. Gratie and I. Petre. Full structural model refinement as type refinement of colored petri nets. In *In Biological Processes & Petri Nets (BPPN 2015)*, volume 1373 of *CEUR Workshop Proceedings*, pages 70–84. CEUR-WS.org, 2015.

[203] L. Grieco, L. Calzone, I. Bernard-Pierrot, F. Radvanyi, B. Kahn-Peries, and D. Thieffry. Integrative modelling of the influence of mapk network on cancer cell fate decision. *PLoS Computational Biology*, 9(10):1–15, 2013.

[204] G. Großmann, C. Kyriakopoulos, L. Bortolussi, and V. Wolf. Lumping the approximate master equation for multistate processes on complex networks. In *Quantitative Evaluation of Systems (QEST 2018)*, pages 157–172. Springer, 2018.

[205] R. Grosu, G. Batt, F. H. Fenton, J. Glimm, C. Le Guernic, S. A. Smolka, and E. Bartocci. From cardiac cells to genetic regulatory networks. In *Computer Aided Verification (CAV 2011)*, volume 6806 of *Lecture Notes in Computer Science*, pages 396–411. Springer, 2011.

[206] C. M. Guldberg and P. Waage. Studies concerning affinity. *Forhandelinger: Videnskabs-Selskabet i Christiana*, 1864.

[207] R. Guzzi, T. Colombo, and P. Paci. *Inverse Problems in Systems Biology: A Critical Review*, pages 69–94. Springer, 2018.

[208] J. E. Haag, A. Vande Wouwer, and M. Remy. A general model of reaction kinetics in biological systems. *Bioprocess and Biosystems Engineering*, 27(5):303–309, 2005.

[209] L. Habets and J. H. van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40(1):21 – 35, 2004.

[210] M. Hajnal, D. Šafránek, M. Demko, S. Pastva, P. Krejčí, and L. Brim. Toward modelling and analysis of transient and sustained behaviour

of signalling pathways. In *Hybrid Systems Biology (HSB 2016)*, pages 57–66. Springer, 2016.

[211] K. Hamaguchi, H. Hiraishi, and S. Yajima. Formal verification of speed-dependent asynchronous circuits using symbolic model checking of branching time regular temporal logic. *Lecture Notes in Computer Science*, 575:410–421, 1991.

[212] T. Han, J. Katoen, and A. Mereacre. Approximate parameter synthesis for probabilistic time-bounded reachability. In *2008 Real-Time Systems Symposium*, pages 173–182, 2008.

[213] K. Havelund and G. Rou. Monitoring java programs with java pathexplorer. *Electronic Notes in Theoretical Computer Science*, 55(2):200 – 217, 2001.

[214] B. R. Haverkort, A. Bell, and H. C. Bohnenkamp. On the efficient sequential and distributed generation of very large Markov chains from stochastic Petri nets. In *Petri Net and Performance Models (PNPM)*, pages 12–21. IEEE Computer Society Press, 1999.

[215] M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In *Formal methods for the design of computer, communication, and software systems 8th international conference on Formal methods for computational systems biology (SFM)*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer, 2008.

[216] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick. Snoopy – a unifying petri net tool. In *Application and Theory of Petri Nets*, pages 398–407. Springer, 2012.

[217] T. Henzinger. The theory of hybrid automata. In *Logic in Computer Science (LICS)*, pages 278 –292. IEEE Computer Society, 1996.

[218] T. Henzinger and H. Wong-Toi. Using hytech to synthesize control parameters for a steam boiler. In *In Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control, Lecture Notes in Computer Science 1165*, pages 265–282. Springer, 1996.

[219] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 373–382. ACM, 1995.

[220] T. A. Henzinger, M. Mateescu, and V. Wolf. Sliding Window Abstraction for Infinite Markov Chains. In *Computer Aided Verification (CAV 2009)*, volume 5643 of *Lecture Notes in Computer Science*, pages 337–352. Springer, 2009.

[221] H.-M. Ho, J. Ouaknine, and J. Worrell. Online monitoring of metric temporal logic. In *Runtime Verification*, pages 178–192. Springer, 2014.

[222] G. J. Holzmann. The model checker spin. *IEEE Trans. Softw. Eng.*, 23(5):279–295, 1997.

[223] R. Honorato-Zimmer et al. Chromar, a Language of Parametrised Objects. *Theoretical Computer Science*, 2017.

[224] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. Copasi – a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[225] F. Horn and R. Jackson. General mass action kinetics. *Archive for Rational Mechanics and Analysis*, 47:81–116, 1972.

[226] B. Hoxha, A. Dokhanchi, and G. Fainekos. Mining parametric temporal logic properties in model-based design for cyber-physical systems. *International Journal on Software Tools for Technology Transfer*, 20(1):79–93, 2018.

[227] B. Hoxha, N. Mavridis, and G. Fainekos. Vispec: A graphical tool for elicitation of mtl requirements. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3486–3492, 2015.

[228] Hucka, M. et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[229] C. A. Hunt, G. E. Ropella, S. Park, and J. Engelberg. Dichotomies between computational and mathematical models. *Nature biotechnology*, 26(7):737, 2008.

[230] M. A. Islam, G. Byrne, S. Kong, E. Clarke, R. Cleaveland, F. Fenton, R. Grosu, P. Jones, and S. Smolka. Bifurcation analysis of cardiac alternans using δ-decidability. In *CMSB*, volume 9859 of *Lecture Notes in Computer Science*, pages 132–146, 2016.

[231] C. Jarzynski. Stochastic and Macroscopic Thermodynamics of Strongly Coupled Systems. *Physical Review X*, 7(1):011008, 2017.

[232] S. K. Jha, E. M. Clarke, C. Langmead, A. Legay, A. Platzer, and P. Zuliani. A bayesian approach to model checking biological systems. In *Computational Methods in Systems Biology (CMSB 2009)*, volume 5688 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2009.

[233] B. Joshi and A. Shiu. Atoms of multistationarity in chemical reaction networks. *Journal of Mathematical Chemistry*, 51(1):153–178, 2013.

[234] M. C. Jr., F. Dannenberg, N. Paoletti, M. Kwiatkowska, and L. Brim. Precise parameter synthesis for stochastic biochemical systems. *Acta Inf.*, 54(6):589–623, 2017.

[235] N. Kashtan, A. E. Mayo, T. Kalisky, and U. Alon. An analytically solvable model for rapid evolution of modular structure. *PLoS Computational Biology*, 5(4):1–14, 2009.

[236] W. O. Kermack and A. G. McKendrick. A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society of London. Series A*, 115(772):700–721, 1927.

[237] Z. Khalis, J. Comet, A. Richard, and G. Bernot. The smbionet method for discovering models of gene regulatory networks. *Genes, Genomes and Genomics*, 3(1), 2009.

[238] M. Khammash. An engineering viewpoint on biological robustness. *BMC Biology*, 14(1):22, Mar 2016.

[239] B. N. Kholodenko. Cell-signalling dynamics in time and space. *Nature Molecular Cell Biology*, 7:165–176, 2006.

[240] H. Kitano. Computational systems biology. *Nature*, 420:206–210, 2002.

[241] H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.

[242] H. Kitano. Towards a theory of biological robustness. *Molecular Systems Biology*, 3(137), 2007.

[243] H. Klarner, A. Streck, and H. Siebert. Pyboolnet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics*, 33(5):770–772, 2017.

[244] H. Klarner, A. Streck, D. Šafránek, J. Kolčák, and H. Siebert. Parameter identification and model ranking of thomas networks. In *Computational Methods in Systems Biology (CMSB 2012)*, Lecture Notes in Computer Science, pages 207–226. Springer, 2012.

[245] M. Klement, D. Šafrńek, T. Děd, A. Pejznoch, L. Nedbal, R. Steuer, J. Červený, and S. Mueller. A Comprehensive Web-based Platform For Domain-Specific Biological Models . *Electronic Notes in Theoretical Computer Science*, 299(0):61 – 67, 2013.

[246] C. H. Koh, M. Nagasaki, A. Saito, C. Li, L. Wong, and S. Miyano. MIRACH: efficient model checker for quantitative biological pathway models. *Bioinformatics*, 27(5):734–735, 2011.

[247] C. H. Koh, S. Palaniappan, P. Thiagarajan, and L. Wong. Improved Statistical Model Checking Methods for Pathway Analysis. *BMC Bioinformatics*, 13(Suppl 17):S15, 2012.

[248] J. Kolčák, D. Šafránek, S. Haar, and L. Paulevé. Parameter space abstraction and unfolding semantics of discrete regulatory networks. *Theoretical Computer Science*, 2018.

[249] M. Komorowski, M. J. Costa, D. A. Rand, and M. P. H. Stumpf. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, 108(21):8645–8650, 2011.

[250] H. Kong, E. Bartocci, S. Bogomolov, R. Grosu, T. A. Henzinger, Y. Jiang, and C. Schilling. Discrete abstraction of multiaffine systems. In *Hybrid Systems Biology (HSB 2016)*, volume 9957 of *Lecture Notes in Computer Science*, pages 128–144, 2016.

[251] H. Kong, E. Bartocci, and T. A. Henzinger. Reachable set over-approximation for nonlinear systems using piecewise barrier tubes. In *Computer Aided Verification (CAV 2018)*, pages 449–467. Springer, 2018.

[252] S. Kong, S. Gao, W. Chen, and E. Clarke. dReach: δ-Reachability Analysis for Hybrid Systems. In *TACAS 2015*, volume 9035 of *Lecture Notes in Computer Science*, pages 200–205. Springer, 2015.

[253] S. Konur and M. Gheorghe. A property-driven methodology for formal analysis of synthetic biology systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 12(2):360–371, 2015.

[254] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2:255–299, 1990.

[255] J. A. Kromer, S. Märcker, S. Lange, C. Baier, and B. M. Friedrich. Decision making improves sperm chemotaxis in the presence of noise. *PLoS computational biology*, 14(4):e1006109, 2018.

[256] N. P. Kurumbang et al. Computer-assisted engineering of the synthetic pathway for biodegradation of a toxic persistent pollutant. *ACS synthetic biology*, 3(3):172–181, 2013.

[257] M. Kvasnica, P. Grieder, and M. Baotić. *Multi-Parametric Toolbox (MPT)*, 2004.

[258] M. Kwiatkowska and J. Heath. Biological pathways as communicating computer systems. *Journal of Cell Science*, 122(16):2793–2800, 2009.

[259] M. Kwiatkowska, G. Norman, and A. Pacheco. Model Checking Expected Time and Expected Reward Formulae with Random Time Bounds. *Compu. Math. Appl.*, 51(2):305 – 316, 2006.

[260] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *Formal Methods for Performance Evaluation: 7th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM)*, volume 4486 of *Lecture Notes in Computer Science*, pages 220–270. Springer, 2007.

[261] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Computer Aided Verification (CAV 2011)*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.

[262] M. Lapin, L. Mikeev, and V. Wolf. Shave: Stochastic hybrid analysis of markov population models. In *Hybrid Systems: Computation and Control (HSCC 2011)*, HHSCC 2011, pages 311–312. ACM, 2011.

[263] K. G. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *International journal on software tools for technology transfer*, 1(1-2):134–152, 1997.

[264] N. Le Novere, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M. I. Aladjem, S. M. Wimalaratne, et al. The systems biology graphical notation. *Nature biotechnology*, 27(8):735, 2009.

[265] C. Li, M. Donizelli, N. Rodriguez, H. Dharuri, L. Endler, V. Chelliah, L. Li, E. He, A. Henry, M. I. Stefan, J. L. Snoep, M. Hucka, N. Le Novère, and C. Laibe. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology*, 4:92, 2010.

[266] Y. Li, A. Albarghouthi, Z. Kincaid, A. Gurfinkel, and M. Chechik. Symbolic optimization with SMT solvers. In *POPL '14*, pages 607–618. ACM, 2014.

[267] L. Lindemann. *Robust and Abstraction-free Control of Dynamical Systems under Signal Temporal Logic Specifications*. PhD thesis, KTH, 2018.

[268] P. Liò, N. Paoletti, M. A. Moni, K. Atwell, E. Merelli, and M. Viceconti. Modelling osteomyelitis. *BMC bioinformatics*, 13(14):S12, 2012.

[269] B. Liu, S. Kong, S. Gao, P. Zuliani, and E. M. Clarke. Parameter synthesis for cardiac cell hybrid models using δ-decisions. In *Computational Methods in Systems Biology (CMSB 2014)*, pages 99–113. Springer, 2014.

[270] X. Liu, J. Jiang, O. Ajayi, X. Gu, D. Gilbert, and R. Sinnott. Bionessie: a grid enabled biochemical networks simulation environment. *Studies in Health Technology and Informatics*, 138:147–157, 2008.

[271] C. M. Lloyd, M. D. Halstead, and P. F. Nielsen. CellML: its future, present and past. *Progress in biophysics and molecular biology*, 85(2-3):433–450, 2004.

[272] A. J. Lotka. Elements of Physical Biology. *Nature*, 116:461–461, 1925.

[273] A. Lück and V. Wolf. Generalized method of moments for estimating parameters of stochastic reaction networks. *BMC Systems Biology*, 10(1):98, 2016.

[274] C. Madsen, C. Myers, N. Roehner, C. Winstead, and Z. Zhang. Utilizing Stochastic Model Checking to Analyze Genetic Circuits. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 379–386, 2012.

[275] C. Madsen, F. Shmarov, and P. Zuliani. BioPSy: An SMT-based tool for guaranteed parameter set synthesis of biological models. In *Computational Methods in Systems Biology (CMSB 2015)*, volume 9308 of *Lecture Notes in Computer Science*, pages 182–194. Springer, 2015.

[276] O. Maler and G. Batt. Approximating continuous systems by timed automata. In *Formal Methods in Systems Biology (FMSB)*, Lecture Notes in Computer Science, pages 77–89. Springer, 2008.

[277] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant System (FORMATS/FTRTFT)*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004.

[278] O. Maler and D. Ničković. Monitoring properties of analog and mixed-signal circuits. *International Journal on Software Tools for Technology Transfer*, 15(3):247–268, 2013.

[279] E. D. Maria, F. Fages, A. Rizk, and S. Soliman. Design, optimization and predictions of a coupled model of the cell cycle, circadian clock, dna repair system, irinotecan metabolism and exposure control under temporal logic constraints. *Theoretical Computer Science*, 412(21):2108 – 2127, 2011.

[280] R. Mateescu, P. T. Monteiro, E. Dumas, and H. de Jong. CTRL: Extension of CTL with regular expressions and fairness operators to verify genetic regulatory networks. *Theoretical Computer Science*, 412(26):2854–2883, 2011.

[281] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid petri net representation of gene regulatory network. *Pac Symp Biocomput.*, pages 341–352, 2000.

[282] A. Mbodj, E. H. Gustafson, L. Ciglar, G. Junion, A. Gonzalez, C. Girardot, L. Perrin, E. E. M. Furlong, and D. Thieffry. Qualitative dynamical modelling can formally explain mesoderm specification and predict novel developmental phenotypes. *PLoS Computational Biology*, 12(9):1–17, 2016.

[283] T. Melham, J. Bard, E. Werner, and D. Noble. Conceptual foundations of systems biology. *Prog. Biophys. Mol. Biol.*, 2012.

[284] P. Mendes. *Modeling large scale biological systems from functional genomic data: parameter estimation*, pages 163 – 186. MIT Press, Cambridge, MA, 2001.

[285] L. Michaelis and M. Menten. Die kinetik der invertinwirkung. *Biochem Z.*, pages 333–369, 1913.

[286] L. Michaelis, M. Menten, K. Johnson, and R. Goody. The original michaelis constant: translation of the 1913 michaelis-menten paper. *Biochemistry*, 50(39):8264–8269, 2011.

[287] R. Milo, P. Jorgensen, U. Moran, G. Weber, and M. Springer. Bionumbersthe database of key numbers in molecular and cell biology. *Nucleic Acids Research*, 38(suppl_1):D750–D753, 2010.

[288] C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13(11):2467–2474, 2003.

[289] P. Monteiro, W. Abou-Jaoud, D. Thieffry, and C. Chaouiya. Model checking logical regulatory networks. *IFAC Proceedings Volumes*, 47(2):170 – 175, 2014.

[290] P. Monteiro, D. Ropers, R. Mateescu, A. Freitas, and H. de Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233, 2008.

[291] P. T. Monteiro, D. Ropers, R. Mateescu, A. T. Freitas, and H. de Jong. Temporal Logic Patterns for Querying Qualitative Models of Genetic Regulatory Networks. In *ECAI*, volume 178 of *FAIA*, pages 229–233. IOS Press, 2008.

[292] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of chemical physics*, 124:044104, 2006.

[293] C. J. Myers. *Engineering Genetic Circuits*. CRC Press, 2009.

[294] A. Naldi, P. T. Monteiro, C. Mssel, the Consortium for Logical Models, Tools, H. A. Kestler, D. Thieffry, I. Xenarios, J. Saez-Rodriguez, T. Helikar, and C. Chaouiya. Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics*, 31(7):1154–1159, 2015.

[295] L. Nenzi, S. Silvetti, E. Bartocci, and L. Bortolussi. A robust genetic algorithm for learning temporal specifications from data. In *Quantitative Evaluation of Systems (QEST 2018)*, pages 323–338. Springer, 2018.

[296] D. Ničković. Monitoring and measuring hybrid behaviors. In *Runtime Verification*, pages 378–402. Springer, 2015.

[297] D. Ničković, O. Lebeltel, O. Maler, T. Ferrère, and D. Ulus. Amt 2.0: Qualitative and quantitative trace analysis with extended signal temporal logic. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 303–319. Springer, 2018.

[298] D. Nickovic and O. Maler. Amt: A property-based monitoring tool for analog systems. In *Formal Modeling and Analysis of Timed Systems*, pages 304–319. Springer, 2007.

[299] J. R. Norris. *Continuous-time Markov chains I*, page 60107. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[300] M. Ostrowski, L. Paulevé, T. Schaub, A. Siegel, and C. Guziolowski. Boolean network identification from multiplex time series data. In *Computational Methods in Systems Biology (CMSB 2015)*, pages 170–181. Springer, 2015.

[301] B. Pal, A. Banerjee, P. Dasgupta, and P. Chakrabarti. Buspec: A framework for generation of verification aids for standard bus protocol specifications. *Integration*, 40(3):285–304, 2007.

[302] M. Pedersen et al. A High-level Language for Rule-based Modelling. *Plos One*, 10:1–26, 2015.

[303] C. A. Petri and W. Reisig. Petri net. *Scholarpedia*, 3(4):6477, 2008.

[304] A. Phillips and L. Cardelli. Efficient, correct simulation of biological processes in the stochastic π-calculus. In *Computational Methods in Systems Biology (CMSB 2007)*, Lecture Notes in Computer Science, pages 184–199. Springer, 2007.

[305] A. Platzer and J.-D. Quesel. Keymaera: A hybrid theorem prover for hybrid systems (system description). In *Automated Reasoning*, volume 5195 of *Lecture Notes in Computer Science*, pages 171–178. Springer, 2008.

[306] A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13(1):45 – 60, 1981.

[307] L. Popova-Zeugmann, M. Heiner, and I. Koch. Time Petri Nets for Modelling and Analysis of Biochemical Networks. *Fundamenta Informatica*, 67(1-3):149–162, 2005.

[308] S. Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117 – 126, 2006.

[309] C. Priami. Algorithmic systems biology. *Communications of the ACM*, 52(5):80–88, 2009.

[310] C. Priami, A. Regev, E. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80(1):25–31, 2001.

[311] J. P. Queille and J. Sifakis. Specification and verification of concurrent systems in cesar. In *International Symposium on Programming*, pages 337–351. Springer, 1982.

[312] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia. Reactive synthesis from signal temporal logic specifications. In *Hybrid Systems: Computation and Control (HSCC 2015)*, pages 239–248. ACM, 2015.

[313] S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation-based abstraction refinement. *ACM Transactions on Embedded Computing Systems*, 6(1), 2007.

[314] A. Raue, J. Karlsson, M. P. Saccomani, M. Jirstrand, and J. Timmer. Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*, 30(10):1440–1448, 2014.

[315] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, 2004.

[316] A. Regev, W. Silverman, and E. Y. Shapiro. Representation and Simulation of Biochemical Processes Using the π-Calculus Process Algebra. In *Pacific Symposium on Biocomputing*, pages 459–470, 2001.

[317] D. Reguera, J. M. Rubí, and J. M. G. Vilar. The mesoscopic dynamics of thermodynamic systems. *The Journal of Physical Chemistry B*, 109(46):21502–21515, 2005.

[318] S. Reinker, R. Altman, and J. Timmer. Parameter Estimation in Stochastic Biochemical Reactions. *IEEE Proc. Syst. Biol.*, 153(4):168–78, 2006.

[319] A. Rizk, G. Batt, F. Fages, and S. Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25:169–178, 2009.

[320] M. Rybiński, M. Lula, P. Banasik, S. Lasota, and A. Gambin. Tav4sb: integrating tools for analysis of kinetic models of biological systems. *BMC Systems Biology*, 6(1):25, 2012.

[321] A. Sackmann, M. Heiner, and I. Koch. Application of petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics*, 7(1):482, 2006.

[322] K. Sanft, D. Gillespie, and L. Petzold. Legitimacy of the stochastic michaelis-menten approximation. *Systems Biology, IET*, 5(1):58–69, 2011.

[323] S. Sankaranarayanan and A. Tiwari. Relational abstractions for continuous and hybrid systems. In *Computer Aided Verification (CAV 2011)*, volume 6806 of *Lecture Notes in Computer Science*, pages 686–702. Springer, 2011.

[324] S. Sasagawa, Y.-i. Ozaki, K. Fujita, and S. Kuroda. Prediction and validation of the distinct dynamics of transient and sustained erk activation. *Nature cell biology*, 7(4):365–373, 2005.

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

[326] F. Schlögl. Chemical Reaction Models for Non-Equilibrium Phase Transitions. *Zeitschrift fur Physik*, 253:147–161, 1972.

[327] I. Schomburg, L. Jeske, M. Ulbrich, S. Placzek, A. Chang, and D. Schomburg. The brenda enzyme information systemfrom a database to an expert system. *Journal of Biotechnology*, 261:194 – 206, 2017.

[328] M. Schwarick, C. Rohr, and M. Heiner. MARCIE - Model checking And Reachability analysis done effiCIEntly . In *Quantitative Evaluation of SysTems (QEST 2011)*, pages 91–100. IEEE Computer Society, 2011.

[329] K. Sen, M. Viswanathan, and G. Agha. Vesta: A statistical model-checker and analyzer for probabilistic systems. In *Quantitative Evaluation of Systems (QEST 2005)*, pages 251–252, 2005.

[330] G. Shinar, R. Milo, M. R. Martnez, and U. Alon. Inputoutput robustness in simple bacterial signaling systems. *Proceedings of the National Academy of Sciences*, 104(50):19931–19935, 2007.

[331] F. Shmarov and P. Zuliani. Probreach: Verified probabilistic delta-reachability for stochastic hybrid systems. In *Hybrid Systems: Computation and Control (HSCC 2015)*, HSCC 2015, pages 134–139. ACM, 2015.

[332] I. Shmulevich and W. Zhang. Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, 18(4):555–565, 2002.

[333] H. Siebert and A. Bockmayr. Incorporating time delays into the logical analysis of gene regulatory networks. In *Computational Methods in Systems Biology (CMSB 2006)*, volume 4210 of *Lecture Notes in Computer Science*, pages 169–183. Springer, 2006.

[334] A. Singh and R. Grima. The linear-noise approximation and moment-closure approximations for stochastic chemical kinetics. *arXiv preprint arXiv:1711.07383*, 2017.

[335] A. Singh and J. a. P. Hespanha. Stochastic hybrid systems for studying biochemical processes. *Physical and Engineering Sciences*, 368(1930):4995–5011, 2010.

[336] M. W. Sneddon, J. R. Faeder, and T. Emonet. Efficient modeling, simulation and coarse-graining of biological complexity with nfsim. *Nature Methods*, 8(2):177–183, 2011.

[337] M. I. Stefan and N. Le Novre. Cooperative binding. *PLoS Computational Biology*, 9(6):1–6, 2013.

[338] R. Steuer, S. Waldherr, V. Sourjik, and M. Kollmann. Robust signal processing in living cells. *PLoS computational biology*, 7(11):e1002218, 2011.

[339] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1995.

[340] A. Streck. *Toolkit for Reverse Engineering of Molecular Pathways via Parameter Identification*. PhD thesis, Free University of Berlin, Dahlem, Germany, 2016.

[341] A. Streck, J. Kolcák, H. Siebert, and D. Šafránek. Esther: Introducing an online platform for parameter identification of boolean networks. In *CMSB*, pages 257–258. Springer, 2013.

[342] A. Streck, K. Thobe, and H. Siebert. Analysing cell line specific egfr signalling via optimized automata based model checking. In *Computational Methods in Systems Biology (CMSB 2015)*, pages 264–276. Springer, 2015.

[343] S. Streif, K.-K. K. Kim, P. Rumschinski, M. Kishida, D. E. Shen, R. Findeisen, and R. D. Braatz. Robustness analysis, prediction and estimation for uncertain biochemical networks. *IFAC Proceedings Volumes*, 46(32):1 – 20, 2013.

[344] O. Stursberg, S. Kowalewski, I. Hoffmann, and J. Preußig. Comparing timed and hybrid automata as approximations of continuous systems. In *Hybrid Systems IV*, pages 361–377. Springer, 1997.

[345] M. Swat, A. Kel, and H. Herzel. Bifurcation analysis of the regulatory modules of the mammalian G1/S transition. *Bioinformatics*, 20(10):1506–1511, 2004.

[346] M. H. ter Beek, A. Fantechi, S. Gnesi, and F. Mazzanti. A state/event-based model-checking approach for the analysis of abstract system properties. *Sci. Comput. Program.*, 76:119–135, 2011.

[347] D. Thieffry and R. Thomas. Dynamical behaviour of biological regulatory networks—ii. immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57(2):277–297, 1995.

[348] R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563 – 585, 1973.

[349] R. Thomas. Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology*, 153(1):1–23, 1991.

[350] A. Tiwari. Abstractions for hybrid systems. *Formal Methods in System Design*, 32(1):57–83, 2008.

[351] P. Traynard, F. Fages, and S. Soliman. Trace simplifications preserving temporal logic formulae with case study in a coupled model of the cell cycle and the circadian clock. In *Computational Methods in Systems Biology (CMSB 2014)*, pages 114–128. Springer, 2014.

[352] P. Traynard, A. Faur, F. Fages, and D. Thieffry. Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics*, 32(17):i772–i780, 2016.

[353] M. Troják, D. Safránek, J. Hrabec, J. Salagovic, F. Romanovská, and J. Cervený. E-cyanobacterium.org: A web-based platform for systems biology of cyanobacteria. In *Computational Methods in Systems Biology (CMSB 2016)*, volume 9859 of *Lecture Notes in Computer Science*, pages 316–322. Springer, 2016.

[354] J. J. Tyson. Classification of instabilities in chemical reaction systems. *The Journal of Chemical Physics*, 62(3):1010–1015, 1975.

[355] D. Ulus, T. Ferrère, E. Asarin, and O. Maler. Timed pattern matching. In *Formal Modeling and Analysis of Timed Systems*, pages 222–236. Springer, 2014.

[356] T. van Dijk, A. Laarman, and J. van de Pol. Multi-core bdd operations for symbolic reachability. In *11th International Workshop on Parallel and Distributed Methods in verifiCation, PDMC 2012*, Electronic Notes in Theoretical Computer Science, pages 127–143. ELSEVIER, 2012.

[357] A. P. A. van Moorsel and W. H. Sanders. Adaptive uniformization. *ORSA Communications in Statistics: Stochastic Models, vol. 10, no. 3*, pages 619–648, 1994.

[358] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *Proceedings of the Symposium on Logic in Computer Science (LICS '86), Cambridge, Massachusetts, USA, June 16-18, 1986*, pages 332–344. IEEE Computer Society, 1986.

[359] M. Češka, D. Šafránek, S. Dražan, and L. Brim. Robustness analysis of stochastic biochemical systems. *PLoS ONE*, 9(4):1–23, 2014.

[360] V. Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, 1926.

[361] D. Šafránek, J. Červený, M. Klement, J. Pospíšilová, L. Brim, D. Lazár, and L. Nedbal. E-photosynthesis: Web-based platform for modeling of complex photosynthetic processes. *Biosystems*, 103(2):115–124, 2011.

[362] Q. Wang and E. M. Clarke. Formal modeling of biological systems. In *2016 IEEE International High Level Design Validation and Test Workshop (HLDVT)*, pages 178–184, 2016.

[363] L. H. Watanabe and C. J. Myers. Hierarchical stochastic simulation of genetic circuits. In *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, DEVS '14, pages 37:1–37:8. Society for Computer Simulation International, 2014.

[364] R. Wisniewski and C. Sloth. Abstraction of Dynamical Systems by Timed Automata. *Modeling, Identification and Control*, 32(2):79–90, 2011.

[365] S. Yamada, T. Taketomi, and A. Yoshimura. Model analysis of difference between egf pathway and fgf pathway. *Biochemical and biophysical research communications*, 314(4):1113–1120, 2004.

[366] E. Yang, E. van Nimwegen, M. Zavolan, N. Rajewsky, M. Schroeder, M. Magnasco, and J. E. Darnell. Decay Rates of Human mRNAs: Correlation With Functional Characteristics and Sequence Attributes. *Genome Research*, 13(8):1863–1872, 2003.

[367] H.-T. Yang and M. S. H. Ko. Stochastic modeling for the expression of a gene regulated by competing transcription factors. *PLoS ONE*, 7(3):e32376, 2012.

[368] B. Yordanov and C. Belta. Parameter synthesis for piecewise affine systems from temporal logic specifications. In *Hybrid Systems: Computation and Control (HSCC 2008)*, pages 542–555. Springer, 2008.

[369] A. Zaslaver, A. E. Mayo, R. Rosenberg, P. Bashkin, H. Sberro, M. Tsalyuk, M. G. Surette, and U. Alon. Just-in-time transcription program in metabolic pathways. *Nature Genetics*, 36(5):486–491, 2004.

[370] H. Zhang, J. Du, L. Cao, and G. Zhu. A full symbolic reachability analysis algorithm of timed automata based on bdd. In *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems*, pages 301–304, 2015.

[371] J. Zhang, L. T. Watson, and Y. Cao. Adaptive aggregation method for the chemical master equation. *International Journal of Computational Biology and Drug Design*, 2(2):134–148, 2009.

[372] J. Zhou, R. Ramanathan, W.-F. Wong, and P. S. Thiagarajan. Automated property synthesis of odes based bio-pathways models. In *Computational Methods in Systems Biology (CMSB 2017)*, volume 10545 of *Lecture Notes in Computer Science*, pages 265–282. Springer, 2017.

[373] P. Zuliani. Statistical model checking for biological applications. *International Journal on Software Tools for Technology Transfer*, 17(4):527–536, 2015.