Diana-Elena Gratie

Refinement of Biomodels Using Petri Nets

Turku Centre *for* Computer Science

TUCS Dissertations
No 216, October 2016

Refinement of biomodels using Petri nets

Diana-Elena Gratie

To be presented, with the permission of the Faculty of Science and Engineering of Åbo Akademi University, for public criticism in Auditorium XX on October 20, 2016, at 12 noon.

Åbo Akademi University
Faculty of Science and Engineering
Vattenborgsvägen 3
20500 Åbo

2016

Supervisor

Professor Ion Petre
Faculty of Science and Engineering
Åbo Akademi University
Vattenborgsvägen 3, 20500 Turku
Finland

Reviewers

Associate Professor Jetty Kleijn
Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1, 2333 CA Leiden
The Netherlands

Associate Professor Daniela Besozzi
Department of Computer Science
University of Milano-Bicocca
Piazza dell'Ateneo Nuovo, 1 – 20126 Milano
Italy

Opponent

Associate Professor Jetty Kleijn
Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1, 2333 CA Leiden
The Netherlands

ISBN 978-952-12-3438-5
ISSN 1239-1883

Abstract

Systems Biology is the multidisciplinary field concerned with the research of large, complex biological systems from a holistic perspective. The end goal is to understand how such systems function as a whole rather than as the sum of their composing parts. Tools and methods from disciplines such as Biology, Biochemistry, Molecular Biology, Bioinformatics, Mathematics, Physics, Systems Theory and Computer Science are used to this end. Modern Biology generates huge amounts of data, requiring computational tools and computational power to store, manage and analyze the generated data. Moreover, isolated models are not enough anymore in the quest for building comprehensive models of cells, tissues, organs or organisms. In turn, biomodeling efforts are beginning to focus on integrating existing models into larger systems. As such, efforts to reuse ready-fit models have intensified. The modeling focus is shifting from building models from scratch to refining existing models, and integrating models for the construction of large-scale models that comprise several interacting subunits that function at different resolutions.

The work presented in this thesis deals with an important part of the effort of using existing models, namely model refinement. This is the process of adding details to a model in a systematic way such that the new model is more specialized and preserves the properties of the initial model. The stepwise construction (from lower to higher levels of detail) of models is a good strategy for building large models. Moreover, as a by-product it generates several models at different levels of resolution, which could consequently be organized in a comprehensive multilevel model of a system. Current modeling efforts deal with the seamless transition between different levels of detail of a model.

We are concerned in this thesis with modeling methodologies for the refinement of reaction-based biomodels. As a foundation, we consider biological systems that can be modeled as sets of reactions. Namely, a number of entities (species) interact with each other and these interactions describe the behavior of the system. Reaction-based systems can be modeled and simulated using many different frameworks and techniques. We present

briefly several such frameworks. We then focus on the notion of model refinement, both from a qualitative and from a quantitative point of view. We present our idea that refinements of species can propagate to refinements of their interactions when the internal composition of the species is known. We exemplify the refinement concepts on a case study of the eukaryotic heat shock response.

The core of the thesis details our original contributions to refining biomodels in the framework of Petri nets. We compare two existing classes of Petri nets (standard Petri nets and colored Petri nets) with respect to their capabilities of refining models in a compact way. We show that colored Petri nets allow for compact representations of refined models, and that refinements can be modeled using different strategies. Moreover, we prove that a full structural refinement of a model can be implemented via a type refinement of a colored Petri net representation of that model. Finally, we propose a new class of Petri nets for model refinement (composition colored Petri nets), with great potential for automatizing the refinement process. The construction builds on the assumption that species are either atomic or complex (composed from several atomic species), and the internal composition of the complex species is known. This internal structure of species is explicitly modeled in the network, which makes later refinements of atomic species automatically reflect in the complex species. We conclude with a discussion on possible extensions of our formalism and an overview of the current challenges of Systems Biology.

Sammanfattning

Systembiologi är ett multidisciplinellt område som avser forskning om stora, komplexa biologiska system from ett holistiskt perspektiv. Målet är att förstå hur sådana system fungerar som helheter istället för som summorna av deras delar. För detta används verktyg och metoder från områden såsom biologi, biokemi, molekylärbiologi, bioinformatik, matematik, fysik, systemteori och datavetenskap. Modern biologi genererar enorma mängder data, och det krävs beräkningsverktyg och beräkningskapacitet för att lagra, hantera och analysera genererad data. Utöver detta är isolerade modeller inte längre tillräckligt i jakten på att kunna bygga omfattande modeller av celler, vävnader, organ eller organismer. Inom biomodellering har man i sin tur börjat fokusera på att integrera existerande modeller till större system. Därav har ansträngningen för att kunna återanvända passande modeller intensifierats. Fokuset för modelleringen håller på att skifta från att bygga modeller från grunden till att precisera existerande modeller, och till att integrera modeller för att konstruera storskaliga modeller som inbegriper flertalet interagerande delenheter som fungerar på olika nivåer.

Det arbete som presenteras i denna avhandling handlar om en viktig del av användandet av existerande modeller, nämligen precisering av modeller. Det är processen där man lägger till detaljer till en modell på ett systematiskt sätt, så att den nya modellen är mer specialiserad och bevarar egenskaperna hos den ursprungliga modellen. Den stegvisa konstruktionen av modeller (från lägre till högre nivå av detaljer) är en bra strategi för att bygga stora modeller. Dessutom skapar det som biprodukt flertalet modeller på olika nivåer av upplösning, vilka följaktligen kan organiseras till en omfattande flernivåmodell av ett system. Nuvarande modelleringsansträngningar hanterar sömlös övergång från olika detaljnivåer inom en modell.

I denna avhandling intresserar vi oss för modelleringsmetodologier för preciseringen av reaktionsbaserade biomodeller. Som en grund tar vi biologiska system som kan modelleras som mängder av reaktioner. Detta innebär att ett antal entiteter (arter) interagerar med varandra och dessa interaktioner beskriver hur systemet uppför sig. Många olika ramverk och tekniker kan användas för att modellera och simulera reaktionsbaserade sys-

tem. Vi presenterar kort ett flertal sådana ramverk. Efter det fokuserar vi på konceptet modellprecisering, både från en kvalitativ och en kvantitativ synvinkel. Vi presenterar vår idé att precisering av arter kan propagera till precisering av deras interaktioner när arternas interna uppbyggnad är känd. Vi exemplifierar precisionskoncepten med hjälp av en fallstudie av den eukaryotiska värmechockresponsen.

Kärnan i avhandlingen redogör i detalj för våra bidrag till precisering av biomodeller inom Petrinät-ramverket. Vi jämför två existerande klasser av Petrinät (standard-Petrinät och färglagda Petrinät) med avseende på deras kapacitet att precisera modeller på ett kompakt sätt. Vi visar att färglagda Petrinät möjliggör kompakta representationer av preciserade modeller, och att preciseringar kan modelleras med olika strategier. Utöver detta bevisar vi att en full strukturell precisering av en modell kan implementeras via en typprecisering av en färglagd Petrinät-representation av modellen. Slutligen föreslår vi en ny klass av Petrinät för modellprecisering (föreningsfärglagda Petrinät), som har stor potential när det gäller automatisering av preciseringsprocessen. Konstruktionen bygger på antagandet att arterna är antingen atomära eller komplexa (uppbyggda av flera atomära arter), och den interna strukturen hos de komplexa arterna är känd. Denna interna struktur hos arterna är uttryckligen modellerad in nätverket, vilket gör att senare preciseringar av atomära arter automatiskt avspeglas i de komplexa arterna. Vi avslutar med en diskussion om möjliga utökningar av vår formalism och en översikt av aktuella utmaningar inom systembiologi.

Acknowledgements

The journey toward becoming a PhD is one that is never taken alone. Therefore, I thank those who have traveled with me, those who have supported me through this journey from a distance or rooted for me on the sideways, and those who have inspired me to embark on this journey in the first place.

First and foremost, I would like to express my sincere thanks to my supervisor Prof. Ion Petre for traveling with me all the way, whether leading the way or encouraging me to explore unmarked paths. I thank him for all the fruitful and inspiring discussions we have had, for his advice and support, for his patience, for believing in me even when I did not, for granting me independence and for always being willing to help. I am deeply grateful for the chance of doing my doctoral studies under his guidance.

I am grateful to my reviewers Dr. Daniela Besozzi and Dr. Jetty Kleijn for carefully reading an earlier version of this thesis. Their critical review, comments and suggestions contributed to improving the quality of this thesis. I especially thank Dr. Jetty Kleijn for agreeing to act as opponent at the public defense of my doctoral thesis. I also thank my colleagues who have read and commented on an earlier version of this thesis.

I thank my co-authors Sepinoud Azimi, Eugen Czeizler, Cristian Gratie, Bogdan Iancu, Nebiat Ibssa, Ion Petre, Vladimir Rogojin, Tolou Shadbahr and Fatemeh Shokri for collaborating with me. I enjoyed working together and learned a lot in the process. I thank the current and former members and also visitors of the Computational Biomodeling Laboratory (Combio) for their help with work-related issues, for engaging discussions on both science and life in general, for great lab outings and for making Combio such a nice place to be in. I especially thank Bogdan Iancu, Sepinoud Azimi, Charmi Panchal and Vladimir Rogojin.

I gratefully thank Turku Centre for Computer Science (TUCS), the Doctoral Network in Information Technologies and Mathematics (DITM), the former Department of Information Technologies at Åbo Akademi and the Rector of Åbo Akademi for the financial support they offered during my studies. I thank TUCS, Stiftelsen för Åbo Akademi, Akademi of Finland, the EU COGANGS grant and the 7^{th} q-bio Summer School for funding my

travels to conferences, workshops and summer schools.

A special thank you goes to Prof. Cristian Giumale for being a model for me, for opening the door towards biomodeling during my Bachelor's studies, and for all the valuable advice he has given me over the years. A special thank you also to Prof. Adina Florea for encouraging me to come to Finland, for believing in me and for offering greatly appreciated support and advice. I thank both of them for inspiring me to embark on the PhD journey.

I thank Åbo Akademi University and TUCS for the great infrastructure and working environment they provided during my studies. I also thank the administrative personnel at the department and at TUCS for their help with administrative matters, especially Christel Engblom, Nina Hutholm, Tove Österroos, Susanne Ramstedt, Kalle Rönnholm and Tomi Suovuo.

I am lucky to have had wonderful colleagues at the department to discuss with, to share ideas with, to laugh with, to practice foreign languages with and to learn from. I thank all of them for the great moments we have shared at department activities, during breaks, or over lunches in Swedish. I thank the amazing people with whom I have spent time outside of work for wonderful and happy moments: my friends, both old and new, my colleagues and team mates in the OSSCB 2012 Summer School and the 7^{th} q-bio Summer School, the members of the folk dancing association Folkdanslaget Otakt vid Åbo Akademi r.f. and the Young Women's Saint Alexandra Club, as well as my colleagues in the LLEES Master's program. I especially thank my dear friends Bogdan Iancu, Oana Macovei, Alia Joko, Mikhail Barash, Simona Căbuz, Ambrozie Gavrilă, Charmi Panchal and Suwisa Kaewphan. I deeply thank the extended Sandvik family for showing me how beautiful Finland is and for making me feel welcome and more at home here.

I am grateful to my family for their unconditional love, support and encouragements. I would not have made it so far if it weren't for them. My dear parents, thank you for all the sacrifices you have made for me over the years. My dearest brother, thank you for *always* supporting me, for understanding me and for being with me through both tough and happy times. I feel blessed to have had the chance to spend so much time with you. I also thank my dear late grandma Nina for her great contribution to who I am today.

Finally, a very special thank you to Petter Sandvik for everything. And for translating the abstract.

List of Original Publications

1. Diana-Elena Gratie, Bogdan Iancu, and Ion Petre. ODE analysis of biological systems. In: Marco Bernardo, Erik de Vink, Alessandra Di Pierro, and Herbert Wiklicky (Eds.), *Formal Methods for Dynamical Systems*, volume 7938 of Lecture Notes in Computer Science, pp. 29 – 62. Springer Berlin Heidelberg, 2013.

2. Bogdan Iancu, Diana-Elena Gratie, Sepinoud Azimi, and Ion Petre. On the implementation of quantitative model refinement. In: Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe (Eds.), *Algorithms for Computational Biology*, volume 8542 of Lecture Notes in Computer Science, pp. 95 – 106. Springer International Publishing, 2014.

3. Diana-Elena Gratie and Ion Petre. Hiding the combinatorial state space explosion of biomodels through colored Petri nets. *Annals of University of Bucharest*, LXI:23 – 41, Editura Universității din București, 2014.

4. Diana-Elena Gratie and Ion Petre. Full structural model refinement as type refinement of colored Petri nets. In: Monika Heiner and Annegret K. Wagler (Eds.), *Proceedings of the 6th International Workshop on Biological Processes and Petri Nets*, volume 1373 of Ceur Workshop Proceedings, pp. 70 – 84. CEUR Workshop Proceedings, 2015.

5. Diana-Elena Gratie and Cristian Gratie. Composition colored Petri nets for the refinement of reaction-based models. *Electronic Notes in Theoretical Computer Science*, Vol. 326C, pp. 51 – 72, in press.

Contents

Part I

Research Summary

1 | Introduction

Modeling is a way of representing an abstraction of reality with the purpose of capturing the main features of interest to try and understand the mechanisms governing the studied systems or phenomena and make predictions about their behavior in different conditions. Models and modeling have drastically changed since computers started being used for modeling purposes. Visual models have evolved from schematic drawings to 3D complex models with zoom-in capabilities and transitions between different levels of detail [10, 71, 100, 8, 95], setting the ground for new ways of thinking about, understanding and advancing sciences. Computer simulations and mathematical modeling have reshaped sciences, and will continue shaping advancements in, e.g., new energy sources [113, 123, 48], drug discovery [34, 97, 79], personalized medicine [135, 55, 54].

Computation has become an integral part (often with a central role) of a large range of scientific areas, with new directions emerging: computational biosciences, computational mathematics, computational linguistics, computational sociology, etc. The common denominator lies in the computational models that this plethora of sciences uses in order to analyze, understand and make predictions about the subject-specific phenomena that are being studied. Computational modeling and simulations are nowadays vital to the advancement of science.

In the quest for understanding how biological systems work, biologists need to complement observations and experiments with computational tools in order to model the studied system, validate it against available experimental data, and make predictions about the system's behavior when subjected to conditions that in practice would be difficult to obtain, measure or would take too long to complete. Many of the frameworks that were used in different areas have been tailored to fit the needs of biologists for theoretical, computational frameworks necessary for the implementation, testing and validation of biological models. One of these frameworks is the formalism of Petri nets that we have mainly focused on in our work.

The role of computing and modeling in advancing scientific knowledge nowadays has been acknowledged at the highest level when the 2013 Nobel

Prize in Chemistry was awarded for "the development of multiscale models for complex chemical systems" [120]. Simulations allow joining different formalisms rather than having to choose one, and "the computer is just as important a tool [...] as the test tube" [120].

Advances in molecular biology have brought about a change in focus from trying to understand how molecules work to trying to understand how the interplay of molecular interactions gives rise to the observed behavior. This led to the emergence of Systems Biology as a research field bringing together at least Biology, Mathematics, Computer Science and Physics, but also Systems Theory and other fields. Systems Biology has greatly benefited from computational models and from computational techniques and analyses that were previously used in Computer Science, e.g., verification and refinement. The two fields are so intertwined nowadays that building a biological model is much like developing a computer program, as stated in [17].

Systems Biology as a field has at the very beginning benefited from methods and simulations in Computer Science and Mathematics. Nowadays the direction is shifting, with the need for new tools and precision methods pushing for new advancements in computing and Mathematics. This can be seen in standardizations (whether certified or de facto) of, e.g., markup languages for Biology (SBML [70]), gene ontologies (GO [9]) and graphical notations for Biology (SBGN [90]), see [23, 126] for a more detailed list; in the quest for handling Big Data [96, 35]; in the development of programming languages for Biology and biological toolboxes (pySB [93], MatlabSB [122], etc.); in attempts of file sharing systems for scientific data (BioTorrents [89]), but also the need to share both code and data for improving research practices [111].

We are now witnessing a shift in Systems Biology from modeling to model integration in the quest to build models that include interactions on different scales. Models of parts of a system that are verified and fit to experimental data are put together to create systems models that include all the relevant submodules, thus with a grainier level of detail than existing high-level coarse abstractions. The problem with integration is that some models are qualitative, some quantitative, and the modeling frameworks differ. The solution is to establish communication between the different parts of the system. A good example of such a model integration effort for a whole cell model is [80].

The doctoral research included in this thesis is concerned with issues of computational modeling of biological systems. We examine biological models represented as sets of reactions, their entities and quantitative aspects and present our case study in Chapter 2. We continue in Chapter 3 with the notion of refinement of reaction biomodels that we have used in the research included in this thesis, while in Chapters 4 and 5 we focus on the formalism that we have used, Petri nets, the problem of refining Petri net

models, and our proposed approach. We list the contribution of each paper included in this thesis in Chapter 6, and conclude with a discussion about future directions and challenges of modeling in Chapter 7.

2 | Modeling Biological Systems

We present in this chapter a model representation of biological systems as sets of reactions, and the semantics associated with such models. We exemplify this model representation on a case study of the eukaryotic heat shock response, explaining how the biological processes are mapped to reactions.

2.1 What to Include in a Model

Our work is situated in the field of Computational Systems Biology, at the interplay between computational tools and Biology, where the need for biological simulations drives the development of new computational tools, and in turn the results of such simulations help advance the field of Biology [2]. The use of modeling in Biology is important due to the complex character of biological systems, and of living organisms and processes sustaining life in general. Characterizing such complex systems is a great challenge and a topical research subject, with research teams trying to bridge the gap between models at different resolution, using integration techniques with the goal of characterizing a complete organism. Some of the challenges of Systems Biology modeling are: the availability of data; the quality of data in biological databases; biological processes and interactions that are not known or fully understood.

Modeling is omnipresent in the field of Biology, and has been so from the very beginning. Whether we talk about diagrammatic models summarizing a set of observations, results from a breeding experiment, biochemical networks, a signaling pathway, the sequence of nucleotides in a gene, the 3D structure of a protein or a large, integrated, executable whole cell model [80], these are all models with different levels of detail for Biology. With the advancement of computational techniques, the methods of discovery in Biology have shifted from experiment-driven discovery to discoveries based on simulations and validation of hypotheses via computational models [50], and even more recently to computer-based hypothesis generation [136].

With any system, be it biological, industrial, or of a different nature, the first task in model specification is to identify the parts (actors, components, entities) of the system, and the states of these parts that are relevant to include in a model of the system. Depending on the desired level of detail, the model can include a subset of such states, all of the possible states, or a generic state that represents all possible states. For example, a protein with a residue that can be phosphorylated has two possible states governed by its residue: phosphorylated and not phosphorylated. If the phosphorylation status of the protein is not relevant for the desired level of detail, the protein may be considered to have one state. But if the phosphorylation status plays a significant role, e.g., in determining possible interactions, then the protein's two states should be considered separately. The next task in modeling a system is to identify and model the interactions between the selected components of the system. The interactions affect the involved entities, altering their state, transforming them, introducing new actors in the system or removing existing ones.

Upon deciding on the entities and interactions of the system that are to be included in a model, the next step is to choose an appropriate modeling framework. There exist many formalisms and tools that address the issue of modeling dynamical systems, see [27, 112, 47, 125, 17]. Some offer a formal, most often mathematical representation [137, 83, 85], while others focus on the graphical representation, offering a visual intuition of the system's actors and events while leaving out the mathematics of the system dynamics [66, 71, 82]. Finally, some combine these two aspects, offering a visual, often executable formalization of the system, see [7, 17, 103]. The formalism that we have used in our modeling and extended based on our understanding of the requirements for modeling complex systems with combinatorial size explosion, Petri nets, belong to this third class of modeling formalisms.

Finally, the modeling process is cyclic and iterative. After a prototype model is built, there may be a need to add new entities or to change some of the interactions, or even to increase or decrease the level of detail, and then the process of taking modeling decisions repeats.

2.2 Modeling Biological Systems as Reactions

A biological system comprises a set of *species*, or *entities* that denote the molecules, organisms, etc. that are present in a system, and the interactions between these species, called *reactions*, which give the dynamics of the system. Quantitatively, species have an associated *concentration* or number of particles to denote how many copies of the same species are present in the system. Reactions happen with a certain speed, which can be described through a *reaction rate* and depends on the concentration of reactants, tem-

perature, presence or absence of some species, etc. Reactions have a set of *reactants*, or the left hand side of the reaction, which are the elements that are consumed by the reaction, and a set of *products*, on the right hand side of the reaction, denoting the elements that are produced by the reaction. The left and right hand sides of a reaction can be seen as multisets (called *complexes*) of species, and the reaction can be thought of as a rewriting rule.

In biochemical systems, the actors are structured entities, as compounds composed of several atomic elements. Interactions between compounds are based on these atomic elements, and thus representing them individually in the system is really important.

Definition 1. *[36, 57] A* reaction network *is a tuple* $N = (\mathscr{S}, \mathscr{C}, \mathscr{R}, k)$ *where* $\mathscr{S}$ *denotes the finite set of* species*,* $\mathscr{C} \subseteq \mathbb{N}^{\mathscr{S}}$ *denotes the finite set of* complexes*,* $\mathscr{R} \subseteq \mathscr{C} \times \mathscr{C}$ *denotes the finite set of* reactions*, and* $k : \mathscr{R} \to \mathbb{R}_{\geq 0}$ *is a function that assigns to each reaction a* kinetic rate constant*.*

Reactions can either be written as pairs of complexes, or in the form they have in chemistry, i.e.,

$$r_j : \; c_1 S_1 + c_2 S_2 + \cdots + c_n S_n \xrightarrow{k_j} c'_1 S_1 + c'_2 S_2 + \cdots + c'_n S_n, \tag{2.1}$$

where $c_i \geq 0$ denote the stoichiometric coefficients of each species on the left hand side of the reaction, $c'_i \geq 0$ denote the stoichiometric coefficients of the products of the reaction, and k_j denotes the kinetic rate constant of reaction r_j. In practice, whenever a stoichiometric coefficient is 0, that term can be omitted from the reaction.

Compactly, a reaction can either be written as a pair of complexes $(c, d) \in \mathscr{C}$ with the meaning that c denotes the left hand side of the reaction and d denotes its right hand side, or as a compacted reaction $c \to d$, where c and d are complexes as previously defined.

We present in the following section the eukaryotic molecular response to heat stress, and its representation as a set of molecular reactions.

2.3 Running Example: The Heat Shock Response in Eukaryotes

We have modeled and used in our papers the eukaryotic heat shock response as a case study. In this thesis introduction, we will use the same case study or small parts of it to support and exemplify the discussion. We present here the biological details of the process, and their translation to a reaction-based model as reported in [107].

2.3.1 The Biological Process

The eukaryotic cellular response to heat stress, also called *heat shock response*, is a cellular defense mechanism that is well-conserved among species. Its role is vital, that of sustaining cell functionality under stress conditions (e.g., elevated temperatures, toxines, viral infection, etc.). Under irrecoverable conditions (too high intensity or prolonged exposure to stress), the heat shock response may fail to preserve cell functionality, and the cell undergoes apoptosis (controlled cell death) or necrosis (uncontrolled cell death).

When cells are subject to environmental stressors, proteins unfold and start binding to each other (in a try to protect their hydrophobic residues from exposure to water), forming large aggregates that impair the proper functioning of the cell and may ultimately cause cell death. To counter this destructive effect, cells produce *heat shock proteins (HSPs)*, a highly conserved set of proteins among eukaryotes, and a response mechanism is triggered.

HSPs have several functions, as chaperones that assist unfolded or misfolded proteins in their correct refolding, but also in inter-cellular communication, immune response and modulation of apoptosis. HSPs are highly conserved among all eukaryotes, with several families of such proteins and slight adaptations to different organisms, see [91, 52]. HSPs are present in the cell at all times, in low concentrations, some being bound to other proteins, called *heat shock factors (HSFs)*. HSFs enhance the production of HSPs by binding (in a trimeric state) to the HSP-encoding gene and facilitating its expression. When stress is applied, the HSP-HSF complexes break, which allows HSFs to form trimers and bind to the HSP-encoding gene, promoting its expression. The production of HSPs increases so that misfolded proteins can be chaperoned to refold correctly. When enough many HSPs are present in the cell, i.e., very few misfolded proteins remain, HSPs downregulate their expression, by causing HSFs to detach from the gene and thus stopping the synthesis of new HSP proteins.

2.3.2 The HSR Model as a Set of Reactions

We consider as running example the continuous model of the eukaryotic heat shock response proposed in [107], or smaller parts of it. As discussed in the previous section, some proteins and encoding genes are present in low numbers in a cell. This would point to stochastic modeling of the process. However, as discussed in [99], for the particular case of this heat shock response model the dynamics displayed by the stochastic model (1000 runs were analyzed) agree well with the deterministic model. This motivates choosing a deterministic approach in our models.

The set of reactions in the considered HSR model are listed in Table 2.1.

Other computational models of the heat shock response can be found, e.g., in [119, 128, 38].

Table 2.1: The reaction-based model of the eukaryotic heat shock response proposed in [107].

No.	Reaction	No.	Reaction
(1)	$2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$	(7)	$\mathsf{hsp} + \mathsf{hsf}_3 \rightarrow \mathsf{hsp{:}hsf} + 2\,\mathsf{hsf}$
(2)	$\mathsf{hsf} + \mathsf{hsf}_2 \rightleftarrows \mathsf{hsf}_3$	(8)	$\mathsf{hsp} + \mathsf{hsf}_3{:}\mathsf{hse} \rightarrow \mathsf{hsp{:}hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$
(3)	$\mathsf{hsf}_3 + \mathsf{hse} \rightleftarrows \mathsf{hsf}_3{:}\mathsf{hse}$	(9)	$\mathsf{hsp} \rightarrow \emptyset$
(4)	$\mathsf{hsf}_3{:}\mathsf{hse} \rightarrow \mathsf{hsf}_3{:}\mathsf{hse} + \mathsf{hsp}$	(10)	$\mathsf{prot} \rightarrow \mathsf{mfp}$
(5)	$\mathsf{hsp} + \mathsf{hsf} \rightleftarrows \mathsf{hsp{:}hsf}$	(11)	$\mathsf{hsp} + \mathsf{mfp} \rightleftarrows \mathsf{hsp{:}mfp}$
(6)	$\mathsf{hsp} + \mathsf{hsf}_2 \rightarrow \mathsf{hsp{:}hsf} + \mathsf{hsf}$	(12)	$\mathsf{hsp{:}mfp} \rightarrow \mathsf{hsp} + \mathsf{prot}$

Reactions (1)-(2) in Table 2.1 account for the reversible trimerization of HSF molecules, where hsf denotes the monomeric molecule, hsf_2 denotes the dimeric molecule, and hsf_3 denotes the trimeric molecule. Reaction (3) models the DNA binding (hse denotes the HSP-encoding gene binding site) of HSF trimers, while the transcription and translation of a new HSP is captured in reaction (4). The self-regulation of HSP synthesis is modeled in reactions (5)-(8), where HSP binds to HSF and breaks down HSF complexes, while reaction (9) depicts the natural degradation of HSPs. Reactions (10)-(12) model the misfolding of proteins and their HSP-mediated refolding.

2.3.3 A Refined Model of the Heat Shock Response

The model can be refined by adding further details about its species. For example, there exist several families of HSFs (HSF1 through HSF4) and HSPs (HSP27, HSP60, HSP70, HSP90), but in this model all families of HSFs and HSPs have been represented via one species. Further, post-translational modifications of the HSF trimers may affect their binding activity and thus make a good candidate for refinement. Acetylation of HSFs has been found to decrease their binding activity, thus reducing the production of HSPs, see [134].

A model that takes into account the acetylation details of HSFs can be found in [37]; this model has been used in our papers as a benchmark for some of the computational models we have developed. The proposed refinement is to differentiate between acetylated and non-acetylated hsf molecules. Namely, hsf is replaced in the new model by either $\mathsf{rhsf}^{(0)}$, its non-acetylated variant, or by $\mathsf{rhsf}^{(1)}$, its acetylated variant. This introduces many changes in the model, as compounds that contain hsf need to be refined as well. Table 2.2 contains the list of species in the initial model that are subject to refinement and their respective counterparts in the refined model. The species that do not contain hsf molecules will be named as in the initial

model but with a leading "r" to denote that they are part of the refined model, e.g., hse is renamed to rhse in the refined model. The list of refined reactions considered in the model of [37] is presented in Table 2.3

Table 2.2: The refinement of species induced by the identification of the acetylation status of hsf.

Species name	**Refined species name**
hsf	$\mathsf{rhsf}^{(0)}, \mathsf{rhsf}^{(1)}$
$\mathsf{hsf_2}$	$\mathsf{rhsf_2}^{(0)}, \mathsf{rhsf_2}^{(1)}, \mathsf{rhsf_2}^{(2)}$
$\mathsf{hsf_3}$	$\mathsf{rhsf_3}^{(0)}, \mathsf{rhsf_3}^{(1)}, \mathsf{rhsf_3}^{(2)}, \mathsf{rhsf_3}^{(3)}$
$\mathsf{hsf_3{:}hse}$	$\mathsf{rhsf_3}^{(0)}\mathsf{{:}rhse}, \mathsf{rhsf_3}^{(1)}\mathsf{{:}rhse}, \mathsf{rhsf_3}^{(2)}\mathsf{{:}rhse}, \mathsf{rhsf_3}^{(3)}\mathsf{{:}rhse}$
$\mathsf{hsp{:}hsf}$	$\mathsf{rhsp{:}rhsf}^{(0)}, \mathsf{rhsp{:}rhsf}^{(1)}$

Table 2.3: The list of reactions for the refined HSR model that includes the acetylation status of hsf.

Refined reactions
$2\,\mathsf{rhsf}^{(0)} \rightleftarrows \mathsf{rhsf_2}^{(0)}$ $\mathsf{rhsf}^{(0)} + \mathsf{rhsf}^{(1)} \rightleftarrows \mathsf{rhsf_2}^{(1)}$ $2\,\mathsf{rhsf}^{(1)} \rightleftarrows \mathsf{rhsf_2}^{(2)}$
$\mathsf{rhsf}^{(0)} + \mathsf{rhsf_2}^{(0)} \rightleftarrows \mathsf{rhsf_3}^{(0)}$ $\mathsf{rhsf}^{(1)} + \mathsf{rhsf_2}^{(0)} \rightleftarrows \mathsf{rhsf_3}^{(1)}$ $\mathsf{rhsf}^{(0)} + \mathsf{rhsf_2}^{(1)} \rightleftarrows \mathsf{rhsf_3}^{(1)}$ $\mathsf{rhsf}^{(1)} + \mathsf{rhsf_2}^{(1)} \rightleftarrows \mathsf{rhsf_3}^{(2)}$ $\mathsf{rhsf}^{(0)} + \mathsf{rhsf_2}^{(2)} \rightleftarrows \mathsf{rhsf_3}^{(2)}$ $\mathsf{rhsf}^{(1)} + \mathsf{rhsf_2}^{(2)} \rightleftarrows \mathsf{rhsf_3}^{(3)}$
$\mathsf{rhsf_3}^{(0)} + \mathsf{rhse} \rightleftarrows \mathsf{rhsf_3}^{(0)}\mathsf{{:}rhse}$ $\mathsf{rhsf_3}^{(1)} + \mathsf{rhse} \rightleftarrows \mathsf{rhsf_3}^{(1)}\mathsf{{:}rhse}$ $\mathsf{rhsf_3}^{(2)} + \mathsf{rhse} \rightleftarrows \mathsf{rhsf_3}^{(2)}\mathsf{{:}rhse}$ $\mathsf{rhsf_3}^{(3)} + \mathsf{rhse} \rightleftarrows \mathsf{rhsf_3}^{(3)}\mathsf{{:}rhse}$
$\mathsf{rhsf_3}^{(0)}\mathsf{{:}rhse} \rightarrow \mathsf{rhsf_3}^{(0)}\mathsf{{:}rhse} + \mathsf{rhsp}$ $\mathsf{rhsf_3}^{(1)}\mathsf{{:}rhse} \rightarrow \mathsf{rhsf_3}^{(1)}\mathsf{{:}rhse} + \mathsf{rhsp}$ $\mathsf{rhsf_3}^{(2)}\mathsf{{:}rhse} \rightarrow \mathsf{rhsf_3}^{(2)}\mathsf{{:}rhse} + \mathsf{rhsp}$ $\mathsf{rhsf_3}^{(3)}\mathsf{{:}rhse} \rightarrow \mathsf{rhsf_3}^{(3)}\mathsf{{:}rhse} + \mathsf{rhsp}$
$\mathsf{rhsp} + \mathsf{rhsf}^{(0)} \rightleftarrows \mathsf{rhsp{:}rhsf}^{(0)}$ $\mathsf{rhsp} + \mathsf{rhsf}^{(1)} \rightleftarrows \mathsf{rhsp{:}\,rhsf}^{(1)}$
$\mathsf{rhsp} + \mathsf{rhsf_2}^{(0)} \rightarrow \mathsf{rhsp{:}rhsf}^{(0)} + \mathsf{rhsf}^{(0)}$ $\mathsf{rhsp} + \mathsf{rhsf_2}^{(1)} \rightarrow \mathsf{rhsp{:}rhsf}^{(0)} + \mathsf{rhsf}^{(1)}$

Table 2.3: The list of reactions for the refined HSR model that includes the acetylation status of hsf. - Continued

$\mathsf{rhsp} + \mathsf{rhsf}_2^{(1)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(0)}$
$\mathsf{rhsp} + \mathsf{rhsf}_2^{(2)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(1)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(0)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(0)} + 2\,\mathsf{rhsf}^{(0)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(1)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(0)} + \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(0)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(1)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + 2\,\mathsf{rhsf}^{(0)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(2)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(0)} + 2\,\mathsf{rhsf}^{(1)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(2)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(0)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(3)} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + 2\,\mathsf{rhsf}^{(1)}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(0)}\colon \mathsf{rhse} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(0)} + 2\,\mathsf{rhsf}^{(0)} + \mathsf{rhse}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(1)}\colon \mathsf{rhse} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + 2\,\mathsf{rhsf}^{(0)} + \mathsf{rhse}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(1)}\colon \mathsf{rhse} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(0)} + \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(0)} + \mathsf{rhse}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(2)}\colon \mathsf{rhse} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(1)} + \mathsf{rhsf}^{(0)} + \mathsf{rhse}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(2)}\colon \mathsf{rhse} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(0)} + 2\,\mathsf{rhsf}^{(1)} + \mathsf{rhse}$
$\mathsf{rhsp} + \mathsf{rhsf}_3^{(3)}\colon \mathsf{rhse} \rightarrow \mathsf{rhsp}\colon \mathsf{rhsf}^{(1)} + 2\,\mathsf{rhsf}^{(1)} + \mathsf{rhse}$
$\mathsf{rhsp} \rightarrow \emptyset$
$\mathsf{rprot} \rightarrow \mathsf{rmfp}$
$\mathsf{rhsp} + \mathsf{rmfp} \rightleftarrows \mathsf{rhsp}\colon \mathsf{rmfp}$
$\mathsf{rhsp}\colon \mathsf{rmfp} \rightarrow \mathsf{rhsp} + \mathsf{rprot}$

2.4 From Reactions to Computational Models

We have mentioned in Section 2.1 that every reaction has a reaction rate associated to it, to describe how fast the reaction fires. The firing of a reaction modifies the context of the system, by decreasing the concentration levels of the reactants of the reaction and increasing the concentration of the products. Depending on the system being modeled, there are several possibilities of representing it. First, the modeler needs to choose between a *qualitative* and a *quantitative* modeling approach. Qualitative models offer some insights into the main properties of a system and its behavior, but they abstract from the quantitative aspects. In turn, quantitative models comprise details about the quantities of elements present in the system, and how those quantities vary. We briefly present here some widely used modeling frameworks.

2.4.1 ODE-based Models

Ordinary differential equations (ODEs) are widely used for representing the dynamics of deterministic systems as the evolution over time of the concen-

trations of species. Reaction networks can be associated a reaction rate, according to some *kinetic function*. One of the most widely used function is *mass action* kinetics, see [64, 132]. According to this dynamics, the change in concentration of a species depends on the kinetic constants of the reactions it participates in (with a positive change if the species is produced, and a negative change if it is consumed in the reaction), the concentrations of all reactants in the reaction, and their stoichiometries. The general expression of the reaction rate for a mass action reaction of the general form 2.1 is given in Equation 2.2, with v_j denoting the reaction rate, and k_j the kinetic rate constant of the reaction. The resulting set of ODEs is given in Equation 2.3. We use the notation $[S]$ to denote the concentration of a species S. The time course for each species' concentration is obtained via numerical integration of the system of ODEs.

$$v_j = k_j \prod_{i=1}^{n} [S_i]^{c_i}. \tag{2.2}$$

$$\frac{d[S_i]}{dt} = (c_i' - c_i) k_j \prod_{l=1}^{n} [S_l]^{c_l},\ 1 \leq i \leq n. \tag{2.3}$$

In our paper [61] we present some of the basic concepts for modeling biological systems using ODEs, and exemplify them on a model of the heat shock response. We cover the mass action rate law together with several enzymatic reactions rate laws, and present the problem of model fitting, i.e., estimating the parameters of a model such that its predictions fit the available experimental data. We briefly discuss several computational analysis techniques, using as case study the heat shock response.

Quantitative modeling of a system using ODEs means assigning a deterministic behavior to the system. However, difficulties may arise when small populations are involved due to stochastic events. In such cases, a stochastic or a hybrid modeling approach could be preferred. The deterministic and stochastic modeling approaches both assume that the system is well-stirred and at thermodynamical equilibrium. But while ODEs model the average behavior of the system, a stochastic model gives individual runs of the system. Two consecutive runs can be very different, as a run is a random walk through the possible stated of a system. Stochastic models work with particle numbers, i.e., each entity is modeled individually and the underlying principle is that of molecular collisions. A reaction can happen only if there exist enough molecules to collide to produce the output of the reaction. In contrast, ODEs work with concentrations of species and model a continuous evolution of the system, having as underlying concept diffusion-like reactions. Numerical simulations are fast for ODEs, while simulating Gillespie's

algorithm is slow, and many runs are required. It is good to have these differences in mind when making a modeling decision.

Apart from the widely-used ODE modeling formalism, other computational modeling formalisms emerged for computational biomodels, or were adapted/extended to the particularities of the field. We very briefly present here rule-based modeling, CTMCs, guarded command languages, process algebras and Petri nets. For more detailed descriptions and a few other modeling formalisms we refer to [94, 17].

2.4.2 Rule-based Modeling

The rule-based modeling formalism was introduced for biological modeling with the aim of representing molecules and their interactions in a compact and comprehensive way, see [39, 40, 41, 49, 31]. The power of this modeling framework lies in the way it represents the states of molecules and their sites. The formalism supports the definition of rules that are applicable to more than one configuration of a molecule by specifying only the characteristics of interest and allowing the others to have any value. Thus, one single rule may encode several reactions. The formalism is also suitable for data refinement in the sense that we present in Chapter 3, as we have studied in our papers [72, 60]. For large models, a modular extension of the Kappa language has been recently proposed, see [105].

2.4.3 Continuous Time Markov Chains (CTMCs)

Continuous time Markov chains are a mathematical formalism that can be used to describe the transformation of a system. For a CTMC representing a biological system, the states are the number of molecules for each species, and the transitions are reactions (where for every reaction, the next state of the CTMC model is determined by subtracting from each variable encoding a species present in the reactants of the reaction the quantities indicated in the reaction, and adding to each variable encoding a species present in the products of the reaction the quantities indicated in the reaction). Each transition has a probabilistic rate of firing, and the built computational model can be verified, see [25].

2.4.4 Guarded Command Languages

Guarded command languages are another formalism that can be used for modeling and checking properties of biological systems. As an example PRISM [86] can be used for the probabilistic model checking of biomodels. Internally, models are represented as continuous or discrete Markov chains while the desired properties are designed using temporal logic. The PRISM framework generates all possible behaviors of the system in the form of a

state-transition system, with states representing the possible configurations of the system and transitions representing the transition from one state to another as the result of executing an action. Transitions are assigned rates to account for the probability of the transition being fired within time t. For models represented as CTMCs, due to the quantitative nature of the model, its properties are expressed in probabilistic temporal logic, with the result being the probability that the checked property holds. We refer to [72] for a PRISM representation of the heat shock response model.

2.4.5 Process Algebras

The formalism of process algebras [19, 51] was originally introduced in Computer Science to study the behavior of systems using algebraic methods and emerged based on communicating sequential processes [24] and the calculus of communicating systems [98]. A process algebra can be used to both specify and verify a system (i.e., verify that the system satisfies some given property), and allows for different kinds of composition of systems. Several extensions of process algebras have addressed issues like time, parametrization of processes and probabilistic modeling, see [15]. Among the extensions originating from biomodeling, we list classification of components, introduction of reaction rates, etc. Some of the process algebra tools developed for biomodeling are SPiM [110], Beta Workbench [45] and Bio-PEPA [32]. We refer to [33] for a more detailed list.

2.4.6 Petri Nets

Petri nets are a formalism initially introduced in [108] for the modeling of concurrent systems. Due to the concurrent nature of biological processes, this formalism is suitable as a modeling framework for biological systems. In Chapter 4 we discuss in more detail modeling biosystems using Petri nets, and mention several of the existing extensions of the formalism, some of which were specifically introduced for biomodeling purposes.

The presented formalisms are closely connected with each other, and transformations from one formalism to another are possible. For example, a PRISM model can be constructed based on a Petri net model, ODEs can be automatically derived from a process algebra model, and all models can be exported to the standard format SBML [70] that facilitates import into different formalisms.

3 | Refinement of Reaction Network Models

Model building is a time-consuming process that is constantly subject to change, as new information about the modeled system needs to be added to the model. It is a good practice thus to start with a coarse model that incorporates the main features of the system being modeled, and check the main properties of the model. Subsequent steps can add further details to the model, while making sure that the checked properties are still satisfied. In many cases, it is easier to check the properties on the coarse model and on the module being added, and infer theoretically that the properties of interest are still satisfied by the refined model. This allows for a systematic, modular approach to modeling, where an entity or action is replaced by a detailed module. Refinement in this sense has been applied in different domains, e.g., software development [12, 26, 18], theorem provers [56], safety critical systems [22, 20], formal modeling [124, 1], etc. The formal specifications and details vary of course, but the underlying principle is the same.

For domains where the built systems are both dynamic and quantitative, but the system can also be analyzed qualitatively, the refinement can be thought of as two-step, with the *structural* refinement step being separated from and preceding the *quantitative* refinement step. The structural refinement step is concerned with adding details regarding the entities in the system (e.g., introducing new entities in the system, or detailing the internal structure of entities already present in the system) and the processes including them. The subsequent quantitative refinement step concerns the quantitative part of the system, e.g., setting new parameters, reaction rates or initial quantities (concentrations) for the newly introduced entities.

3.1 Structural Refinement

Considering the definition of a biomodel presented in Section 2.2, namely Definition 1, we show in this section how a structural refinement of such

models can be made. The refinement is concerned with introducing new (more detailed or specialized) species in the system, and more precisely replacing existing species with a set of new species. These new species can be thought of as differentiated subspecies of the species that they are replacing. For example, in a coarse model one could represent a group of proteins with the same function as one generic protein. In a more detailed model the small differences between these proteins may become significant, and thus replacing the generic protein with its variants is a necessary refinement.

3.1.1 Refinement of Species

In [57] all species are considered to be refined, and the refinements of reactions may include all possible combinations of subspecies that the corresponding reactants and products refine to. More precisely, for species that do not undergo refinement, the refinement means replacing a species with a singleton set, so basically a renaming of the species. For species that are subject to refinement, they will be replaced with the set of subspecies they refine to. This is formally captured in Definition 2.

Definition 2. *[57] Let $\mathscr{S}$ and $\mathscr{S}'$ be two sets of species, and $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ a relation. ρ is said to be a* species refinement relation *iff the following conditions hold:*

1. *for each species $A \in \mathscr{S}$ there exists $A' \in \mathscr{S}'$ such that $(A, A') \in \rho$;*
2. *for each species $A' \in \mathscr{S}'$ there exists exactly one $A \in \mathscr{S}$ such that $(A, A') \in \rho$.*

In other words, a refinement relation ρ says for each species in the initial set to which species or set of species it refines. Every species must refine to at least one new species, possibly more than one. For species that refine to one species, the refinement can be thought of as a renaming of the species, thus in order to have a refinement that implies more than just a renaming of species, there should be at least one species that refines to at least two new (sub)species, namely that the new set of species $\mathscr{S}'$ has cardinality strictly greater than that of $\mathscr{S}$.

3.1.2 Refinement of Reactions

Next we discuss about the refinement of reactions that is induced by the refinement relation ρ. For every reaction, every species on its left hand side and every species on its right hand side will be replaced by a species that it is in relation with according to ρ. Consider a species whose set of refined subspecies contains at least two elements. If this species appears in a reaction with stoichiometry greater than one, then its instances in that reaction

need not refine to the same subspecies. All combinations of substitutions on the left- and right- hand side of the reaction are considered. This is formally captured in Definition 3.

Definition 3. *[57] Let $\mathscr{S}$ and $\mathscr{S}'$ be two sets of species, and $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ a refinement relation. Given two complexes c over $\mathscr{S}$ and c' over $\mathscr{S}'$, c' is said to be a ρ-refinement of c if, for every species $S \in \mathscr{S}$, the stoichiometric coefficients of its subspecies (as dictated by ρ) in c' add up to its stoichiometric coefficient in c. The set of all possible ρ-refinements of c is denoted by $\Delta_\rho(c)$.*

A reaction $c \to d$ over $\mathscr{S}$ is said to be refined to a reaction $c' \to d'$ over $\mathscr{S}'$ if c' and d' are respectively ρ-refinements of c and d. The set of all possible refinements of a reaction $c \to d$ can be written as $\Delta_\rho(c) \times \Delta_\rho(d)$.

Given two models expressed as reaction networks, one is said to be the refinement of the other one if there exists a refinement relation over their respective sets of species such that all reactions in the refined model are refinements of reactions in the initial model, and each reaction in the initial model has at least one refinement in the refined model. This is formally captured in Definition 4.

Definition 4. *[57, 60] Given two reaction networks $N = (\mathscr{S}, \mathscr{C}, \mathscr{R}, k)$ and $N' = (\mathscr{S}', \mathscr{C}', \mathscr{R}', k')$ and a refinement relation $\rho \subseteq \mathscr{S} \times \mathscr{S}'$, N' is said to be a* structural refinement *of N if the following conditions hold:*

1. $\mathscr{C}' = \bigcup_{c \in \mathscr{C}} \Delta_\rho(c)$;
2. $\mathscr{R}' \subseteq \bigcup_{c \to d \in \mathscr{R}} \Delta_\rho(c) \times \Delta_\rho(d)$ *and* $\rho(r) \cap \mathscr{R}' \neq \varnothing \ \forall r \in \mathscr{R}$.

In case of equality in the second condition, N' is said to be the full structural refinement *of N.*

The full structural refinement of a model is considering all possible refined complexes on either side of the reaction, and all possible reactions among them, thus is a complete representation of any possible refined reaction. In some cases, considering the full structural refinement of a model may be needed. However, in practice the refinement of a model may be restricted due to additional constraints. One of the most common such constraints is imposing a correspondence between the left- and right- hand sides of a reaction. Example 1 illustrates such a constraint.

Example 1. *Let* P *be a protein that binds to a receptor* R *in a reaction* $\mathsf{P} + \mathsf{R} \to \mathsf{P}:\mathsf{R}$. *Let* P_1 *and* P_2 *be refinements of* P, R' *be a refinement of* R *and* $\mathsf{P}_1:\mathsf{R}', \mathsf{P}_2:\mathsf{R}'$ *be refinements of* $\mathsf{P}:\mathsf{R}$. *The full structural refinement of*

the considered reaction is given by the following set of reactions:

$$(1)\ \mathsf{P}_1 + \mathsf{R}' \to \mathsf{P}_1 : \mathsf{R}'$$
$$(2)\ \mathsf{P}_1 + \mathsf{R}' \to \mathsf{P}_2 : \mathsf{R}'$$
$$(3)\ \mathsf{P}_2 + \mathsf{R}' \to \mathsf{P}_1 : \mathsf{R}'$$
$$(4)\ \mathsf{P}_2 + \mathsf{R}' \to \mathsf{P}_2 : \mathsf{R}'.$$

If an additional constraint requires that the protein in the bound state on the right hand side of the reaction is precisely the protein on the left hand side of the reaction, then only reactions (1) *and* (4) *will be included in the refined model.*

Structural refinement is only concerned with the structural part of the model, namely which reactions are included in the refined model. For the case where the model has an established quantitative part (namely the kinetic rate constants are set), an equivalence condition between the two quantitative models can be established, as we will present in Section 3.2. However, in the remainder of this thesis we will only be concerned with structural refinement and the challenges that it imposes in our chosen modeling framework, Petri nets.

3.1.3 Propagating Refinement

We have considered in our work that species are of two types: atomic and complex, and that refinement concerns only the atomic species. Complex species can be seen as multisets over the set of atomic species, and thus their refinement can be induced by the refinement of their constituents. For compounds that are defined as multisets of atomic species, we use the notation $i`A$ to denote i copies of species A, to preserve the notation used in colored Petri nets.

Example 2. *Assume a model M comprises the set of species $\mathscr{S} = \{A, B, C\}$. Consider further that A and B are atomic species, while C is a complex species that contains one A and two Bs: $C = \{1`A, 2`B\}$. Let $\rho \subseteq \{A, B\} \times \{X, Y, Z\}$ be the following refinement relation:*

$$\rho = \{(A, X), (B, Y), (B, Z)\}.$$

One could extend the refinement relation ρ based on the composition of C and the refinements of A and B with the following pairs, to get a refinement of both the atomic and the complex species of the model:

$$(\{1`A, 2`B\}, \{1`X, 2`Y\});$$
$$(\{1`A, 2`B\}, \{1`X, 1`Y, 1`Z\});$$
$$(\{1`A, 2`B\}, \{1`X, 2`Z\}).$$

In Example 2 the refinements of C can be inferred based on the refinements of the species composing C: each entity in C is refined according to the refinement relation ρ. We think of this as the *propagation* of the refinement from atomic species to complex species. The advantage of propagation of refinement is that only the atomic species need to be refined, while the complex species are refined automatically, in contrast to refining every species individually.

3.2 Quantitative Refinement

Although we have focused primarily on structural refinements of biomodels, an important aspect of biomodels is their quantitative part, namely the concentration (or number of particles) of each species and its evolution over time, the speed of reactions and the kind of kinetic laws they obey (mass action, Michaelis-Menten, other types of enzymatic reactions, special kinetics depending on external factors like temperature, etc.). Work has been done that focuses on the quantitative refinement of biomodels, offering algorithmic solutions to setting the new parameters of a refined model based on the parameters of the initial model, the refinement that is to be implemented, and the kinetic laws of each reaction to be refined, see [39, 37, 57, 11].

For a biomodel whose reactions follow the mass action kinetic law, a sufficient condition for it to be quantitatively equivalent with a model representing a refinement of it is given in [57]. We summarize here briefly the result.

Theorem 1. *[57] Let $N = (\mathscr{S}, \mathscr{C}, \mathscr{R}, k)$ be a reaction network and $N' = (\mathscr{S}', \mathscr{C}', \mathscr{R}', k')$ a full structural $\rho-$refinement of N with $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ the refinement relation. If for every $\boldsymbol{c} \to \boldsymbol{d} \in \mathscr{R}$ and for every $\boldsymbol{c'} \in \Delta_\rho(\boldsymbol{c})$ it holds that*

$$\sum_{\boldsymbol{d'} \in \Delta_\rho(\boldsymbol{d})} k'_{\boldsymbol{c'} \to \boldsymbol{d'}} = \binom{\boldsymbol{c}}{\boldsymbol{c'}} k_{\boldsymbol{c} \to \boldsymbol{d}}, \text{ where } \binom{\boldsymbol{c}}{\boldsymbol{c'}} = \frac{\prod_{i=1}^{|\mathscr{S}|} c_i!}{\prod_{j=1}^{|\mathscr{S}'|} c'_i!},$$

then N' is a fit-preserving refinement of N.

In other words, in a refinement that preserves the fit of the initial model the rate constants of the refined reactions with the same left hand side depend on the rate of the parent reaction and on the stoichiometric coefficients on the left hand sides in a linear manner. The fit-preserving condition presented in Theorem 1 holds for partial refinements as well. In a matrix representation, the condition is that the sum of the kinetic constants of all reactions whose left-hand side is a refinement of a reaction in the initial model equals the kinetic constant of the initial reaction factored by a dependency of the stoichiometries on the left hand side of both reactions. There

exist of course an infinity of values for the kinetic constants that satisfy the fit preserving condition. For a refined model where some of the kinetic constants of refined reactions are set (experimentally), the fit preservation can be attained in the refined model by setting the kinetic constants to satisfy the fit condition, in the case that the already set constants do not violate the condition to begin with. Extra biological information (e.g., all refinements of a reaction fire with the same rate) could also help compute the values of the kinetic constants.

We exemplify the setting of kinetic constants for our example partial refinement of the dimerization reaction in the heat shock response case study (namely the forward direction of reaction (1) in Table 2.1) in Table 3.1, and the backward direction of the trimerization reaction in Table 3.2.

The fit-preserving conditions for the dimerization read:

$$k'_1 = k_1; \quad 2k'_2 = k_1; \quad k'_3 = k_1.$$

The conditions give directly the kinetic constants, since there is only one refinement for each left hand side of a refined reaction.

The fit-preserving conditions for the refined reactions in Table 3.2 read:

$$k'_1 = k_1; \quad k'_2 + k'_3 = k_1; \quad k'_4 + k'_5 = k_1; \quad k'_6 = k_1.$$

In this case the constants k'_2, k'_3, k'_4 and k'_5 were chosen to have equal values, just like in [37].

Table 3.1: The initial dimerization reaction and its included refinements.

Initial reaction	**Refined reactions**
$2\,\mathsf{hsf} \xrightarrow{k_1} \mathsf{hsf}_2$	$2\,\mathsf{rhsf}^{(0)} \xrightarrow{k'_1} \mathsf{rhsf}_2{}^{(0)}$ $\mathsf{rhsf}^{(0)} + \mathsf{rhsf}^{(1)} \xrightarrow{k'_2} \mathsf{rhsf}_2{}^{(1)}$ $2\,\mathsf{rhsf}^{(1)} \xrightarrow{k'_3} \mathsf{rhsf}_2{}^{(2)}$

Table 3.2: The initial de-trimerization reaction and its included refinements.

Initial reaction	**Refined reactions**
$\mathsf{hsf}_3 \xrightarrow{k_1} \mathsf{hsf}_2 + \mathsf{hsf}$	$\mathsf{rhsf}_3{}^{(0)} \xrightarrow{k'_1} \mathsf{rhsf}^{(0)} + \mathsf{rhsf}_2{}^{(0)}$ $\mathsf{rhsf}_3{}^{(1)} \xrightarrow{k'_2} \mathsf{rhsf}^{(1)} + \mathsf{rhsf}_2{}^{(0)}$ $\mathsf{rhsf}_3{}^{(1)} \xrightarrow{k'_3} \mathsf{rhsf}^{(0)} + \mathsf{rhsf}_2{}^{(1)}$ $\mathsf{rhsf}_3{}^{(2)} \xrightarrow{k'_4} \mathsf{rhsf}^{(1)} + \mathsf{rhsf}_2{}^{(1)}$ $\mathsf{rhsf}_3{}^{(2)} \xrightarrow{k'_5} \mathsf{rhsf}^{(0)} + \mathsf{rhsf}_2{}^{(2)}$ $\mathsf{rhsf}_3{}^{(3)} \xrightarrow{k'_6} \mathsf{rhsf}^{(1)} + \mathsf{rhsf}_2{}^{(2)}$

4 | Model Refinement in the Petri Net Framework

In this chapter we recall the notions of Petri nets, colored Petri nets, and present several existing refinements of Petri nets. Out of the refinements, type refinement of colored Petri nets will form the basis for the next chapter. We present the required steps for translating a reaction-based model into a Petri net representation, and exemplify it on our case study.

4.1 Introduction to (Colored) Petri Nets

The Petri net formalism has been introduced by Carl Adam Petri in 1962, in his thesis "Kommunikation mit Automaten", see [108]. The formalism was developed for representing asynchronous, concurrent systems with resource sharing, and has been constantly developed and extended to fit the particularities of other fields ever since. There exist nowadays many extensions that allow for the modeling, analysis and simulation of a multitude of types of systems: extended Petri nets [4, 106], self-modifying nets [130], predicate/transition nets [53], reconfigurable Petri nets [14, 13], timed Petri nets [114, 133], colored Petri nets [73, 74, 75], stochastic Petri nets [101, 5], continuous Petri nets [42], hybrid Petri nets [6, 44], fuzzy Petri nets [92, 29], object Petri nets [87], etc. Some of the extensions have a direct mapping to the particularities of biological systems (e.g., inhibitory reactions can be represented with extended Petri nets containing inhibitor arcs [102, 116]). Both structural and dynamic analysis techniques can be used for discrete, stochastic, continuous or hybrid models. For details on Petri nets definitions and properties we refer to [106, 117, 102, 43, 46, 118], and for extensions of Petri nets and their applications we refer to [3, 78, 43, 44, 16].

Petri nets are being used as a modeling framework for many fields, ranging from Computer Science to Control Engineering, Manufacturing Systems (production chains), Information Science, Systems Biology, etc.

4.1.1 Definition

The formalism of Petri nets evolved a lot from its original definition, and due to the many developed extensions there exist several slightly different definitions (with respect to notations and whether the structure is separated from the semantics or not), see [106, 117, 43, 65, 67, 118]. We will consider in this thesis the definition of Petri nets as in [117, 102], and for colored Petri nets we consider the definition in [75]. Although definitions of Petri nets tailored for biology, where different types of arcs and transition functions are considered exist, see [65, 67], our main focus is on the structural aspects of modeling with (colored) Petri nets and implementing refinements. Thus, we opted for a general definition in order to keep the results and techniques open for other application domains.

Definition 5. *[102] A Petri net is a 5-tuple* $N = (P, T, A, f, M_0)$ *such that:*

- P *is the finite set of* places*;*
- T *is the finite set of* transitions*, such that* $P \cap T = \varnothing$*;*
- $A \subseteq (P \times T) \cup (T \times P)$ *is a finite set of arcs;*
- $f : A \to \mathbb{N}^*$ *is a weight function;*
- $M_0 : P \to \mathbb{N}$ *is an initialization function.*

Apart from the formal definition, Petri nets offer as well an intuitive graphical representation, which is a great asset for the formalism to be adopted in many domains. Places are depicted as circles or ellipses, transitions are pictured as bars, squares or rectangles, and arcs are represented as arrows indicating the direction of an interaction: from a place to a transition, or from a transition to a place.

4.1.2 Elements of a Petri Net

Places can be thought of as *passive* elements of the network, and they in a biological context encode the molecules or entities of a system, or a signal. Transitions are in turn the *active* elements of the network, and they encode the *actions* in a system, i.e., the biological interactions and molecular transformations. Both places and transitions are frequently named according to the entities and events that they represent. Places host quantitative information about how many entities represented by the place are present in the system, as its *marking*. For low numbers (usually up to 3), this information can be depicted through *tokens*: black dots inside a place, as many as the place's marking. The initialization function assigns for each place its initial marking.

Arcs are used to depict which actors are involved in which interactions, and the weight function gives the cost of an interaction, i.e., how many elements are being consumed (for a place-to-transition arc) or produced (for a transition-to-place arc). The fact that arcs only connect a place with a transition or a transition with a place yields the bipartite structure of Petri nets.

For a given transition $t \in T$, the set of places $p \in P$ such that $(p, t) \in A$ is called the set of *pre-places* of t, denoted ${}^\bullet t$, and the set of places such that $(t, p) \in A$ is called the set of *post-places* of t, denoted $t^\bullet$.

4.1.3 Semantics

We have mentioned that transitions are the active elements of a Petri net. They may fire when certain preconditions are met, removing tokens from their pre-places and adding tokens to their post-places, thus changing the marking of the net. This gives the behavior of a Petri net. A transition t is said to be *enabled* in a marking M if all its pre-places are sufficiently marked, i.e., $M(p) \geq f((p, t)), \forall p \in {}^\bullet t$. An enabled transition may *fire*, changing the markings of all the places connected to it, as explained next.

For a Petri net $N = (P, T, A, f, M_0)$ with $n = |P|$ places and $m = |T|$ transitions, one can assume an ordering of the places and an ordering of the transitions (e.g., in lexicographic order of their name). For a transition t, the effect of its firing on the place marking can be represented as a column vector $\underline{t} \in \mathbb{N}^n$, whose entries are:

$$\underline{t_i} = \begin{cases} f(t, p_i), \text{ if } (t, p_i) \in A \text{ and } (p_i, t) \notin A; \\ f(p_i, t), \text{ if } (p_i, t) \in A \text{ and } (t, p_i) \notin A; \\ f(t, p_i) - f(p_i, t), \text{ if } (t, p_i) \in A \text{ and } (p_i, t) \in A; \\ 0, \text{ otherwise.} \end{cases}$$

All the transition vectors form the matrix of the Petri net, $\underline{N} \in \mathbb{N}^{n \times m}$, such that $\underline{N} = (\underline{t_1}, \ldots, \underline{t_m})$. This matrix is also called the incidence matrix, and is sometimes denoted by C in the literature.

The firing of a transition may change the number of tokens in the places connected to it, possibly enabling new transitions or disabling others. The repeated firing of transitions gives the behavior of the Petri net. This can be represented as the *reachability graph* of the network, which can be finite or infinite. This graph has as nodes markings that are reachable from the initial marking, while its arcs represent transition firing events. Transitions may fire following different strategies. In some semantics, only one transition is fired at a time, while in other semantics parallel firings are allowed. Transitions that are concurrently enabled and are not in conflict may fire concurrently. In the case of a conflict, where two or more transitions are

competing for some token(s), a nondeterministic choice is made for which transition to fire.

In the case of standard Petri nets, only one transition is fired at a time and the firing changes the marking of the network thus impacting the transitions that are enabled at the next step. The whole behavior of the network is given by all possible partially or totally ordered firing sequences. However, kinetic information can be added to standard Petri nets, resulting in different behaviors.

In stochastic Petri nets, every transition is assigned a firing rate which indicates the time that needs to elapse from when the transition becomes enabled to when it can be fired. These are randomly distributed variables according to some probability distribution function. Every transition is also assigned a stochastic hazard function that is dependent on the marking of its pre-places. The behavior of stochastic Petri nets is described by continuous time Markov chains labeled with the transition rates (or sum of rates for parallel transitions).

In continuous Petri nets the place markings and arc weights are non-negative reals. Every transition is assigned a firing rate function, and the transition fires continuously provided that all of its pre-places are marked with a value greater than zero, with the strength of the flow determined by the rate function and the markings of its pre-places. The semantics of continuous Petri nets is described by ODEs.

Hybrid Petri nets have both continuous and stochastic transitions, with the stochastic and continuous parts influencing each other with the restriction that discrete places are not connected to continuous transitions through standard arcs. Stochastic transitions follow the stochastic semantics, and continuous transitions follow the continuous semantics.

We refer to [21] for a very recent overview of modeling with different kinds of Petri nets for Biology including examples, properties and analysis techniques.

4.1.4 Properties

Some of the behavioral properties of a Petri net that are of most interest in a biological model are *liveness*, *boundedness* and *reversibility*, while the structural properties of interest are *P-invariants* and *T-invariants*. We explain briefly next the definition of these properties and their biological interpretation. For formal definitions and more properties we refer to [117, 102].

A Petri net is said to be *live* if from any marking reachable from the initial marking it is possible to fire any transition through some firing sequence. In a biological setup, this property ensures that every reaction can be executed at some point. It also means that the system will never reach a state where no reactions can be executed.

A place is said to be *k-bounded* if k is an upper bound on the number of tokens that it will ever host in any reachable marking. A Petri net is *k-bounded* if all its places are k-bounded. This property ensures that there exists no infinite accumulation of some element in a system.

A Petri net is *reversible* if the initial marking can be reached again from any reachable marking. In a biological system, it may happen that some reactions are irreversible, and the reversibility property of the network shows that there exists an alternative pathway to produce the species that are consumed by some of the reactions in the model.

A *place invariant (P-invariant)* is a vector of places $x : P \to \mathbb{N}^n$, where $n = |P|$, such that $\underline{N}^T \cdot x = 0$. In a biological setting, P-invariants encode mass conservation relations, and may account for different forms of some species. For example, in the HSR model there is an invariant that accounts for the total number of hsf molecules in the system being constant regardless of the form of the molecule (monomeric or in a complex), i.e., $\mathsf{HSF} + 2 \cdot \mathsf{HSF_2} + 3 \cdot \mathsf{HSF_3} + \mathsf{HSP{:}HSF} + 3 \cdot \mathsf{HSF_3{:}HSE}$ is constant (we used here with a slight abuse of notation the place names in Table 4.1 to denote the number of tokens in the corresponding place). Any place involved in a P-invariant is bounded.

A *transition invariant (T-invariant)* is a vector of transitions $y : T \to \mathbb{N}^m$, where $m = |T|$, such that $\underline{N} \cdot y = 0$. The effect of firing all the transitions in a T-invariant on the marking of the network is null. A Petri net is said to be covered by T-invariants if every transition of the network is part of a T-invariant. A network that is covered by T-invariants has no accumulations of species, because of the null effect of firing all transitions in a T-invariant.

Apart from the many available analysis techniques, one of the key advantages of Petri nets over other modeling formalisms for biological systems is their intuitive graphical representation. Another advantage for modellers is the availability of many software tools for the construction, simulation and analysis of Petri nets models. For a comprehensive list of Petri net software tools we refer to [109].

4.1.5 Colored Petri Nets

Colored Petri nets (also called CP-nets) [76, 77] add a layer of programmability to the formalism of Petri nets, by introducing data types for the places in the form of color sets, and supplementary conditions for the firing of a transition, via transition guards. Places no longer host identical elements (tokens), but are seen instead as bags containing possibly different elements of the same type (colors of a color set). Arc expressions may use typed variables from a set of variables V, with $\mathsf{Type}[x]$ denoting the type (color set) of a variable or expression x, and $EXPR_V$ denoting the set of expressions using variables from the set V. S_{MS} is used to denote the set of all multisets

over S, see [77].

Definition 6. *[77] A* colored Petri Net (CP-net) *is a tuple* $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ *satisfying the requirements below:*

1. P, T, A *are defined as for a standard Petri net.*
2. Σ *is a finite set of non-empty types, called* color sets*.*
3. V *is a finite set of* typed variables*, where* $\mathsf{Type}[v] \in \Sigma$*, for all* v *in* V*.*
4. $C : P \to \Sigma$ *is a* color set *function that assigns to every place a color set.*
5. $G : T \to EXPR_V$ *is a* guard *function that defines conditions for transitions.*
6. $E : A \to EXPR_V$ *is an* arc expression *function with* $Type(E(a)) = C(p(a))_{MS}$*, for all arcs* a *in* A*, where* $p(a)$ *is the place corresponding to arc* a*.*
7. I *is an* initialization *function assigning to each place* p *an initialization expression such that* $Type(I(p)) = C(p)_{MS}$*.*

In order to evaluate the value of arc expressions and transition guards, all variables in an expression (transition guard) must be assigned a value from the corresponding data type. This process is called *binding*: a variable is bound to a certain value. A binding of all variables that are associated to a transition (i.e., all variables of the arc expressions on in- and out-arcs connected to the transition and all the variables in the guard of that transition) forms a *transition instance*. If the guard of a transition instance evaluates to true and there are enough many tokens in the pre-places of the transition as dictated by the bindings of the arc expressions on incoming arcs, then the transition instance is enabled and may fire. The firing of a transition instance will modify the marking of the net by removing colored tokens from the pre-places of the transition and adding colored tokens to its post-places, where the colors of the tokens are determined by the arc expressions and variable bindings.

Colored Petri nets can be used to address one of the challenges of modeling large systems with Petri nets, namely that the representation can become too large. CP-nets allow for similar subnets to be folded in one subnet, with the distinction between the individual subnets being made via colors. They also allow foldings of slightly different subnets, of branching and of conflict processes. This may yield a significantly smaller network, that is easier to work with. Furthermore, a CP-net can easily scale up, e.g., by adding colors to the color sets of those subnets that need to be supplemented, or scale down by removing colors when certain subnets can be omitted.

4.2 Biological Models as Petri Nets

The idea of representing metabolic pathways using Petri nets has been introduced in [116], and exemplified by the same authors in [115] on a case study of the combined metabolism of two pathways in an erythrocyte cell. The advantage of using this formalism is in building computational models that offer qualitative analysis methods for the network, based on the structural properties of the network. The authors argue that compounds (e.g., proteins) are represented as places, with a possibility of representing the same protein through several places, if it has several possible states or it is important to locate the compartment it is in. Transitions are used to represent reactions or even depict a set of successive intermediary reactions, when by-products are not of interest. The ideas in [116] have been extended to allow not only a qualitative analysis of a biological system represented as a Petri net, but also discrete and continuous representations of biopathways, see [121, 69, 67]. For details on techniques for modeling different kinds of biological networks (cell signaling pathways, gene regulatory networks, metabolic pathways) using Petri nets we refer to [28, 67, 16, 84].

We will exemplify the procedure of translating a reaction-based model into a Petri net on our heat shock response case study. For this, consider the set of reactions in Table 2.1. Each species is represented as a place in a Petri net, and each reaction is represented as a transition. As a naming convention, each place is named after the species it is representing (as present in the metabolic model in Table 2.1), in capitals, and transitions are numbered such that transition t_i represents reaction no. i in Table 2.1. For the reversible reactions, we differentiate the forward direction as t_i^F and the backward direction as t_i^B. In Tables 4.1 and 4.2 we list the correspondence between places and species, and transitions and reactions, respectively.

Table 4.1: Correspondence between places in the Petri net representation and species in the molecular HSR model.

Place name	**Species**	**Place name**	**Species**
HSF	hsf	HSF_2	hsf_2
HSF_3	hsf_3	HSE	hse
HSF_3:HSE	hsf_3:hse	HSP	hsp
HSP:HSF	hsp:hsf	PROT	prot
MFP	mfp	HSP:MFP	hsp:mfp

Some of the species in our model are involved in more than one reaction. In order to obtain an easily readable Petri net representation of the system, we use *logical places* for some of these species. Logical places are depicted as gray filled circles, and may appear several times in the Petri net, but denote

the same entity. For each reaction we draw arcs from the places representing the substrates of the reaction (species on its left hand side) to the transition representing the reaction, and from the transition representing the reaction to places representing the products of the reaction (species on its right hand side). The resulting Petri net is depicted in Figure 4.2. The weight of each arc is given by the stoichiometry of the place connected to the arc. We only include in the figure the arc weights greater than 1.

Table 4.2: Correspondence between transitions in the Petri net representation and the reactions they are modeling.

Transition	**Reaction**
$\boldsymbol{t_1^F}$	$2\,\mathsf{hsf} \rightarrow \mathsf{hsf}_2$
$\boldsymbol{t_1^B}$	$\mathsf{hsf}_2 \rightarrow 2\,\mathsf{hsf}$
$\boldsymbol{t_2^F}$	$\mathsf{hsf} + \mathsf{hsf}_2 \rightarrow \mathsf{hsf}_3$
$\boldsymbol{t_2^B}$	$\mathsf{hsf}_3 \rightarrow \mathsf{hsf} + \mathsf{hsf}_2$
$\boldsymbol{t_3^F}$	$\mathsf{hsf}_3 + \mathsf{hse} \rightarrow \mathsf{hsf}_3\mathsf{:hse}$
$\boldsymbol{t_3^B}$	$\mathsf{hsf}_3\mathsf{:hse} \rightarrow \mathsf{hsf}_3 + \mathsf{hse}$
$\boldsymbol{t_4}$	$\mathsf{hsf}_3\mathsf{:hse} \rightarrow \mathsf{hsf}_3\mathsf{:hse} + \mathsf{hsp}$
$\boldsymbol{t_5^F}$	$\mathsf{hsp} + \mathsf{hsf} \rightarrow \mathsf{hsp:hsf}$
$\boldsymbol{t_5^B}$	$\mathsf{hsp:hsf} \rightarrow \mathsf{hsp} + \mathsf{hsf}$
$\boldsymbol{t_6}$	$\mathsf{hsp} + \mathsf{hsf}_2 \rightarrow \mathsf{hsp:hsf} + \mathsf{hsf}$
$\boldsymbol{t_7}$	$\mathsf{hsp} + \mathsf{hsf}_3 \rightarrow \mathsf{hsp:hsf} + 2\,\mathsf{hsf}$
$\boldsymbol{t_8}$	$\mathsf{hsp} + \mathsf{hsf}_3\mathsf{:hse} \rightarrow \mathsf{hsp:hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$
$\boldsymbol{t_9}$	$\mathsf{hsp} \rightarrow \emptyset$
$\boldsymbol{t_{10}}$	$\mathsf{prot} \rightarrow \mathsf{mfp}$
$\boldsymbol{t_{11}^F}$	$\mathsf{hsp} + \mathsf{mfp} \rightarrow \mathsf{hsp:mfp}$
$\boldsymbol{t_{11}^B}$	$\mathsf{hsp:mfp} \rightarrow \mathsf{hsp} + \mathsf{mfp}$
$\boldsymbol{t_{12}}$	$\mathsf{hsp:mfp} \rightarrow \mathsf{hsp} + \mathsf{prot}$

4.3 Model Refinement with Petri Nets

The concept of refinements of Petri nets is not new, and in fact there are many known types of refinements for both simple and colored Petri nets. We briefly present here several existing notions of refinement for Petri nets. Some of the refinements are abstractions of the network, i.e., the refined network is smaller, while other refinements are expanding the network.

Refinements can preserve some properties of the network, but they can also introduce new properties for the grainier network. In the case of a behavior-preserving refinement, the behavior of the refined network is determined by the behavior of the initial network, and checked properties are

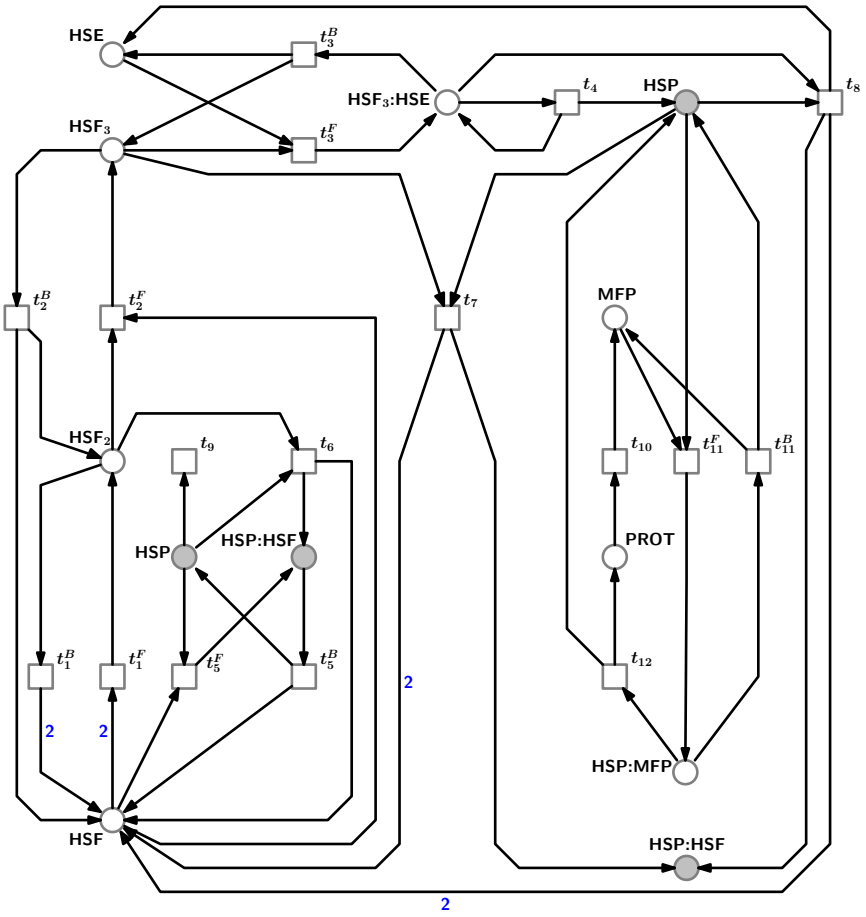


Figure 4.1: Petri net representation of the heat shock response model. Gray places are logical places, i.e., they appear more than once in the network for clarity purposes, but they denote the same entity. Place (transition) names are displayed next to the corresponding place (transition, resp.). Arc expressions greater than 1 are displayed next to the corresponding arc. Figure generated using Snoopy [68].

preserved so that the property checking step is no longer necessary.

4.3.1 Refinements of Petri Nets

The first formalization of refinements of Petri nets dates from 1979, when the notion of stepwise refinement of a Petri net by replacing a transition with a subnetwork was introduced, see [129]. Some transitions can be seen as not instantaneous, meaning that their execution can be detailed by decomposing them into several steps. Thus, a transition could be replaced with a subnet called *block*, under the assumption that the block is well-formed as explained next. A block contains two special transitions: one *initial*, t_{ini}, used to get input from the rest of the network, and one *final*, t_{fin}, used for output to the rest of the network. A Petri net obtained from a block by adding one place (called idle place) that has as only input transition t_{fin} and as only output transition t_{ini} is called the associated Petri net of the block. A block is well-formed if and only if its associated Petri net is live and the initial marking is the only marking where the idle place is not empty and moreover the only enabled transition in the initial marking is t_{ini}. The properties of the initial network (e.g., boundedness, liveness) are preserved if the transition being replaced is not two-enabled (all pre-places of the transition are sufficiently marked so that the transition could fire twice) in any marking and the block it is being replaced with is well-formed. The gain is thus in not having to check the properties of the new, larger Petri net if the properties of the initial network are known and the preservation conditions are fulfilled.

Building on the ideas of [129], a method for expanding or reducing a Petri net via stepwise transformations of its transitions was formalized, see [127]. Namely, a transition could be detailed to a subnet, or a subnet could be abstracted to a single transition, when certain criteria are met. The technique in [129] is generalized from a two-enabledness condition to k-enabledness: a transition that is at most k-enabled can be replaced with a k-well-behaved Petri net to obtain a refinement, as explained next. A transition t is k-enabled if there exists a reachable marking M such that all the pre-places of t are sufficiently marked so that t could be fired k times. A Petri net is said to be k-well-behaved with respect to two transitions t_{ini} and t_{fin} if in the network obtained by adding an idle place with marking k, the initial transition t_{ini} is live, and for any firing sequence the number of firings of t_{ini} is always greater or equal to the number of firings of t_{fin}, and there always exists a firing sequence that can make the number of firings of t_{ini} and t_{fin} equal.

The initial Petri net is seen as an abstraction of the refined one. The procedure is also generalized to refine places: a place p can refine to a subnet consisting of two places and a transition such that one place has as pre-transitions exactly p's pre-transitions and as only post-transition the new

transition, and the other place has the new transition as only pre-transition, and exactly p's post-transitions as post-transitions. The newly introduced transition can at a subsequent step be refined to a subnetwork to further refine the model. In [131] the condition that a subnet has only one transition that affects the input and one that affects the output is relaxed. This allows for the initial network (also called *host network*) and the subnetwork that replaces a transition (also called *daughter network*, or *module*) to be designed independently. The pre-places of the transition to be replaced are input places for the daughter network, while its post-places are output places of the daughter network.

For the formalism of colored Petri nets, refinements of transitions are possible for hierarchical colored Petri nets: a transition can be replaced with a detailed subnetwork (also called page) describing its action. The model is organized as modules communicating through interfaces (subsets of input/output places that are common to two modules, and through which the two modules communicate), see [77]. Models can be represented as hierarchical Petri nets to begin with, but detailing the action of a transition through a submodule, thus refining the network, is also possible.

Some work has been done also on refinements of Petri nets using rules, see [104] for a survey. In this approach, a subnetwork in a Petri net is replaced or extended with a subnetwork in such a way that the properties (liveness or safety properties expressed as logic formulas) of the previous network are preserved. The transformation is done according to rules of the form $L \xleftarrow{k_1} K \xrightarrow{k_2} R$, where L is the left-hand side subnetwork, R is the right-hand side subnetwork, and K is an interface between the two (i.e., a subnetwork of both L and R), with k_1 and k_2 morphisms that satisfy certain assumptions.

Other kinds of refinements of Petri nets have been proposed in the literature, but they are beyond the scope of this work.

4.3.2 Type Refinement of Colored Petri Nets.

We detail here the notion of *type refinement* introduced in [88], since it is the ground for the refinement technique in Chapter 5. The color sets in a colored Petri net represent the data structures or types of the elements contained in its places. Changing the color sets may induce a refinement in the network, if the change adds some new level of detail with respect to the initial color set. This could easily be implemented as suggested in [88] by adding a new field to an existing record color set. The same effect could be obtained by, e.g., replacing an integer color set (or other simple color sets) containing only one value with the same kind of color set that has more values. The structure of the network is preserved, but changes may be needed in the arc expressions and guards.

Every refined color set can be projected onto the initial color set, i.e., by forgetting the details introduced by the refinement, or, in the case of increasing the domain of a color set, the projection of any new color would be the one color in the initial color set.

Example 3. *Consider a model that consists of the reversible dimerization and further reversible trimerization of a molecule* hsf*, namely reactions* (1) *and* (2) *in Table 2.1. The color set for the place representing* hsf *can be a color set with one value, e.g., a record with one field* t *(to denote a token) of the generic type* Dot *containing one element,* dot*:*

$$\begin{aligned} &\mathsf{colorset}\ \mathit{Dot}\ =\ \mathsf{dot};\\ &\mathsf{colorset}\ \mathit{C\_hsf}\ =\ \mathsf{record}\ \mathsf{t} : \mathit{Dot}. \end{aligned}$$

The color sets for the dimer hsf_2 *and trimer* hsf_3 *can be products of the color sets of their substrates. Namely, the color set for the place representing* hsf_2 *is* $\mathsf{colorset}\ \mathit{C\_hsf_2}\ =\ \mathsf{product}\ \mathit{C\_hsf} * \mathit{C\_hsf}$*. The full definition of the model is presented in Figure 4.2.*

Refining the model to include the acetylation of hsf *can be done by extending the domain for the color set associated with* hsf *to include a field* ac *with values* $\{0, 1\}$ *to denote the acetylation status of an* hsf *molecule:*

$$\begin{aligned} &\mathsf{colorset}\ \mathit{Range}\ =\ \mathsf{int\ with}\ 0, 1;\\ &\mathsf{colorset}\ \mathit{C\_hsf}\ =\ \mathsf{record}\ \mathsf{t} : \mathit{Dot} * \mathsf{ac} : \mathit{Range}. \end{aligned}$$

Namely, the color set C_hsf *in the refined definition contains two colors:* $\{\mathsf{t} = \mathsf{dot}, \mathsf{ac} = 0\}$ *and* $\{\mathsf{t} = \mathsf{dot}, \mathsf{ac} = 1\}$*. The color set for* hsf_2 *contains, by definition as a compound color set with two instances of* C_hsf*, four colors while the color set for* hsf_3 *contains eight colors, as listed in Table 4.3.*

The same model could be built by using simple color sets, i.e., integers with domain of one value for the initial model, and integer color sets with two, four and respectively eight values for the color sets of places representing the refined hsf*,* hsf_2 *and* hsf_3*, respectively, as discussed on a different example in [30].*

4.3.3 Refinements of the Heat Shock Response Model as Colored Petri Nets

As discussed in the previous chapters of this work, refining models is an important step of the model building process, and it can be done in a systematic way so that previous data is used (e.g., kinetic parameters of a previously fitted model). Different modeling frameworks can be used for implementing such refinements, and studies have been made on the performance of some of the widely used frameworks with respect to refinements, see [60, 11].

Table 4.3: Type refinement of the color sets. For the refined model, one extra field is added to account for the acetylation status of the molecules. The projection of every refined color onto the initial color set is the color in the initial color set (namely, the field ac is dropped).

Color set	Colors (Initial model)	Colors (Refined model)
C_hsf	$\{t = dot\}$	$\{t = dot, ac = 0\}$ $\{t = dot, ac = 1\}$
C_hsf_2	$(\{t = dot\}, \{t = dot\})$	$(\{t = dot, ac = 0\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 0\}, \{t = dot, ac = 1\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 1\})$
C_hsf_3	$(\{t = dot\}, \{t = dot\}, \{t = dot\})$	$(\{t = dot, ac = 0\}, \{t = dot, ac = 0\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 0\}, \{t = dot, ac = 1\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 0\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 1\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 0\}, \{t = dot, ac = 1\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 0\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 1\}, \{t = dot, ac = 0\})$ $(\{t = dot, ac = 1\}, \{t = dot, ac = 1\}, \{t = dot, ac = 0\})$

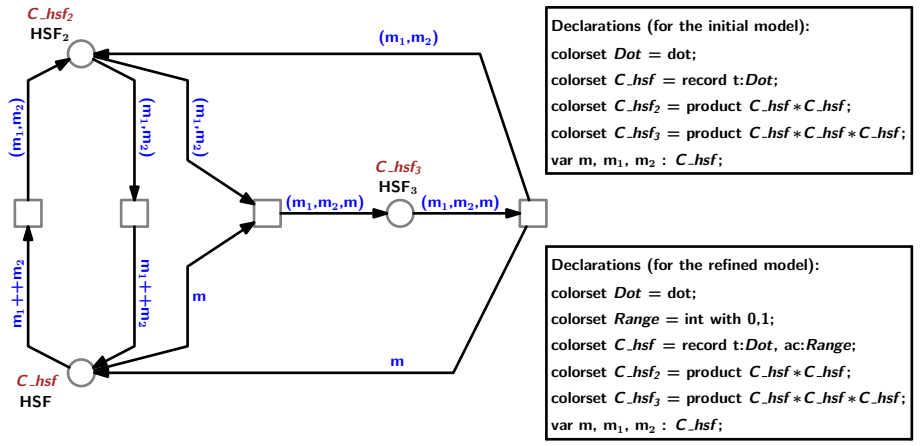


Figure 4.2: Colored Petri net representation of reactions (1)−(2), with insets detailing the color set definition for the initial and the refined models. Place (transition) names are displayed next to the corresponding place (transition, resp.). The color set names are depicted with brown italics, on top of the place name. Arc expressions are displayed in blue. Notation $m_1 + +m_2$ is used to denote a multiset containing variables m_1 and m_2, while (m_1, m_2) denotes a pair from the cartesian product C_hsf_2. Figure generated using Snoopy [68].

Some of the frameworks (e.g., rule-based modeling, colored Petri nets) have built-in mechanisms that allow for a compact representation of the refined models, while others (ODEs, guarded command languages) require a full specification of the expanded model.

Standard Petri nets require a full specification of the refined model, as there is a one-to-one correspondence between species and places, and reactions and transitions, respectively. They are therefore not suitable for refining large models due to the size explosion (in terms of number of places and transitions) that they must explicitly handle. On the contrary, colored Petri nets allow folding places and transitions together respectively, thus reducing the representation of a standard Petri net. Implementing refinements in the formalism of colored Petri nets may resume to choosing suitable color sets for the places that are subject to refinement, and adjusting the transition guards accordingly. For example, in Figure 4.5 all transition guards are set to value true. But if for example there was an extra constraint that a single-acetylated dimer always has the acetylated hsf on the second position, then transition t_1^F should have a guard $[\mathsf{m_1} <= \mathsf{m_2}]$. Alternatively, a modeler may choose to change the structure of the network (by adding new transitions to represent some of the transition refinements) as a trade-off for less complex transition guards and color sets. We have designed two colored Petri net models of the HSR refined model: one that kept the same network structure as in the initial model, and one that used fewer colors and added new transitions in the model to simplify guards, see [72, 59].

The refinement of the HSR model detailed in Chapter 3 can be implemented with colored Petri nets in several ways, depending on the coloring strategy that is chosen, as we first reported in [59] and later detailed in [62, 72, 60]. Depending on the data structures used, it is possible to keep the same topology of the network when implementing the refinement. This was the case for a strategy where the color sets of places representing complex species that were subject to refinement were cartesian products of the color sets of the places representing their composing elements, see Figure 4.5. Another way of preserving the topology is to use multisets as color sets for places representing complex species, see [63].

The second coloring strategy was to use simple color sets for some of the complex species, and introduce some new transitions. The colored Petri net corresponding to this strategy is depicted in Figure 4.4. The color sets *Mono*, *Di* and *Tri* are integer data types with values $\{0, 1\}$, $\{0, 1, 2\}$ and $\{0, 1, 2, 3\}$ respectively. The values are interpreted as the number of acetylated hsf molecules. For example, value 1 in color set *Di* denotes a refined dimer with one acetylated site, namely $\mathsf{rhsf_2}^{(1)}$. Some of the refined reactions in Table 2.3 have the same left hand side but different right hand sides. In such cases, we represented the different reactions with the same left hand side with different transitions. For example, transitions t_6^2 and t_6^3 have the same

guard, $[\mathsf{d} = 1]$ and represent reactions with the same left hand side (namely, $\mathsf{rhsp} + \mathsf{rhsf_2}^{(1)}$). However, the arc expressions of their outgoing arcs are different, as the right hand sides of the represented reactions are different. Transition t_6^2 represents the reaction $\mathsf{rhsp} + \mathsf{rhsf_2}^{(1)} \rightarrow \mathsf{rhsp{:}\,rhsf}^{(1)} + \mathsf{rhsf}^{(0)}$ and transition t_6^3 represents the reaction $\mathsf{rhsp} + \mathsf{rhsf_2}^{(1)} \rightarrow \mathsf{rhsp{:}\,rhsf}^{(0)} + \mathsf{rhsf}^{(1)}$. A similar reasoning has been applied for all pairs of reactions sharing the same left hand side, but the arc expressions for the corresponding transitions have been omitted in Figure 4.4 for clarity.

The simulation runs for the refined models reproduce the simulation run for the initial model. We include in Figure 4.3 the simulation runs for the initial model, the color-focused refinement and the transition-focused refinement. One can notice that the plot of $\mathsf{hsf_3{:}hse}$ in the initial model, (a), is identical with the summated (label HSF3HSE_total) plot of the refinements of $\mathsf{hsf_3{:}hse}$ in both refined models, plots (b) and (c). Moreover,in the transition-focused refinement plot (c) the refined species with the same number of acetylated hsf molecules are plotted with the same color to show that they have precisely the same behavior. This comes from the choice of equal kinetic constants for the corresponding reactions and equal initial markings.

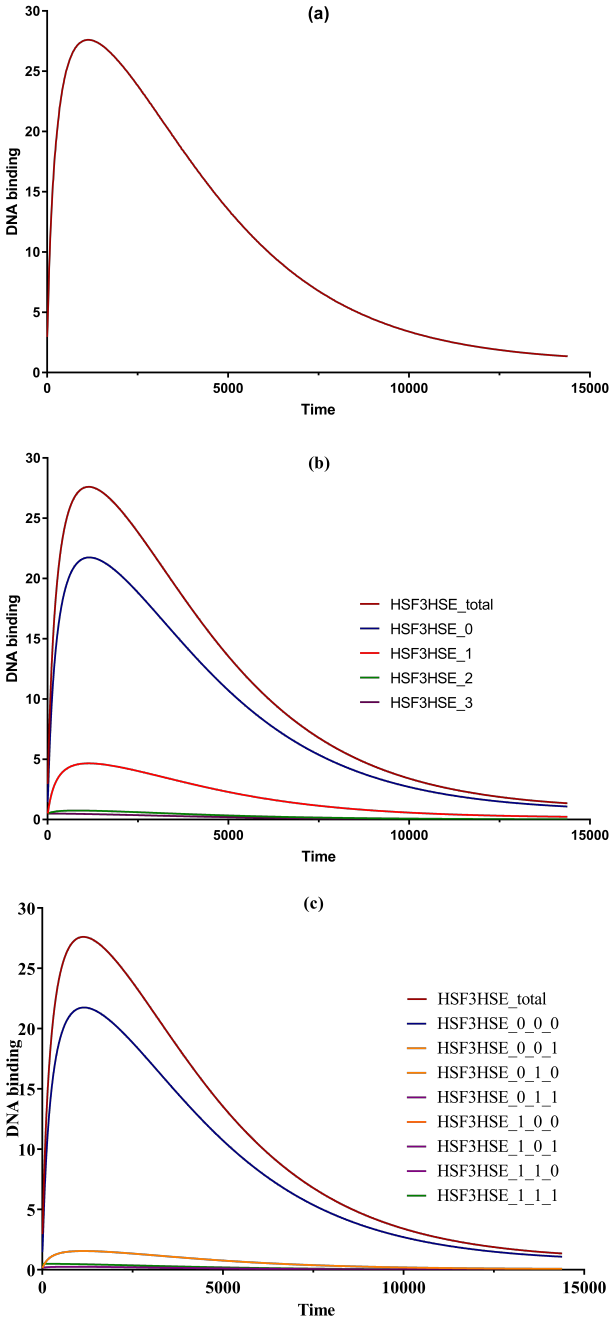


Figure 4.3: Simulation of the initial and refined HSR models. (a) plot of $\mathsf{hsf_3{:}hse}$; (b) plot of the refinements of $\mathsf{hsf_3{:}hse}$ and their sum in the color-based refinement; (c) plot of the refinements of $\mathsf{hsf_3{:}hse}$ and their sum in the transition-based refinement.

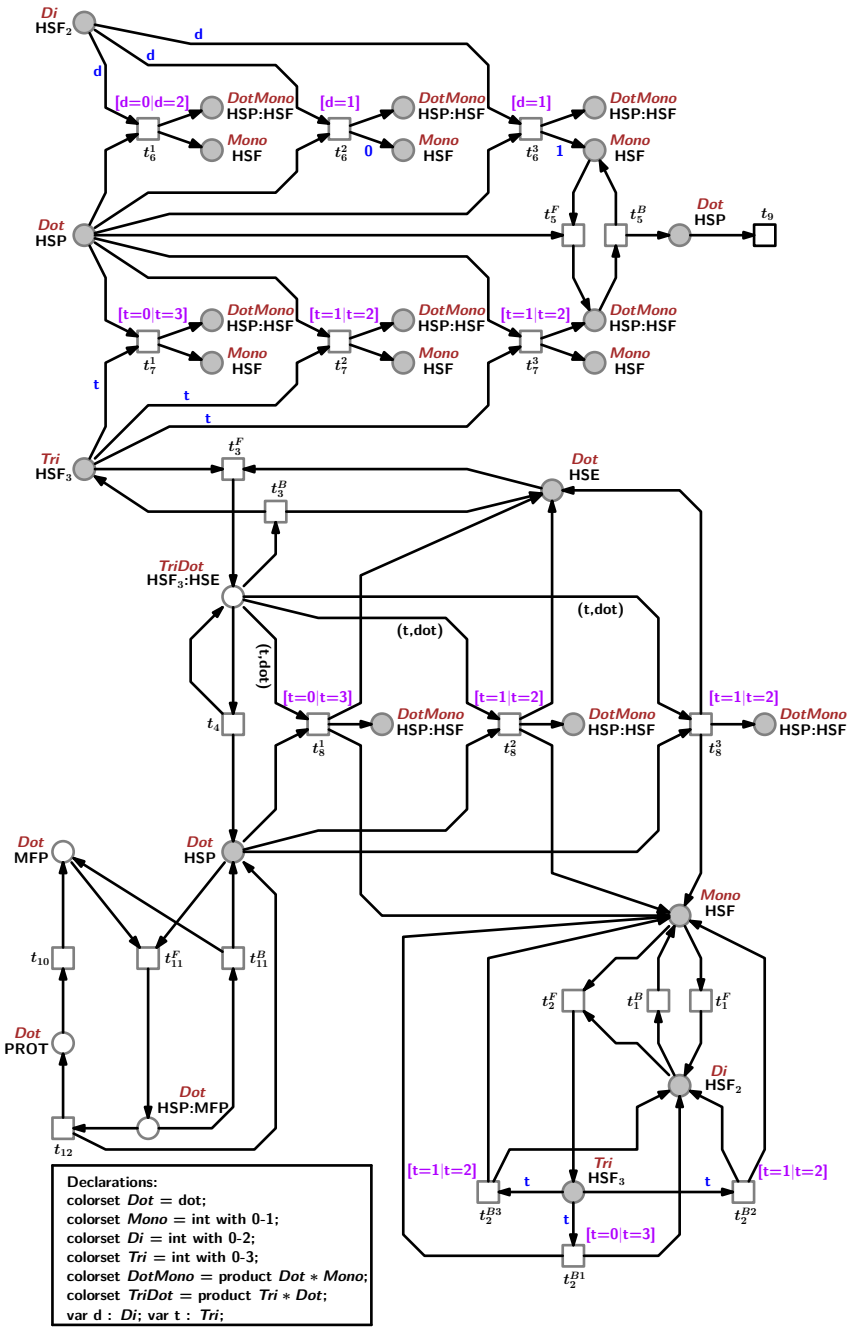


Figure 4.4: Colored Petri net representation of the refined HSR model with different topology from the initial model. Place (transition) names are displayed next to the corresponding place (transition, resp.). The color set names are depicted with brown italics on top of the place name. The inset contains the color set definition. Arc expressions are displayed in blue. Transition guards are displayed with purple. Figure generated using Snoopy [68].

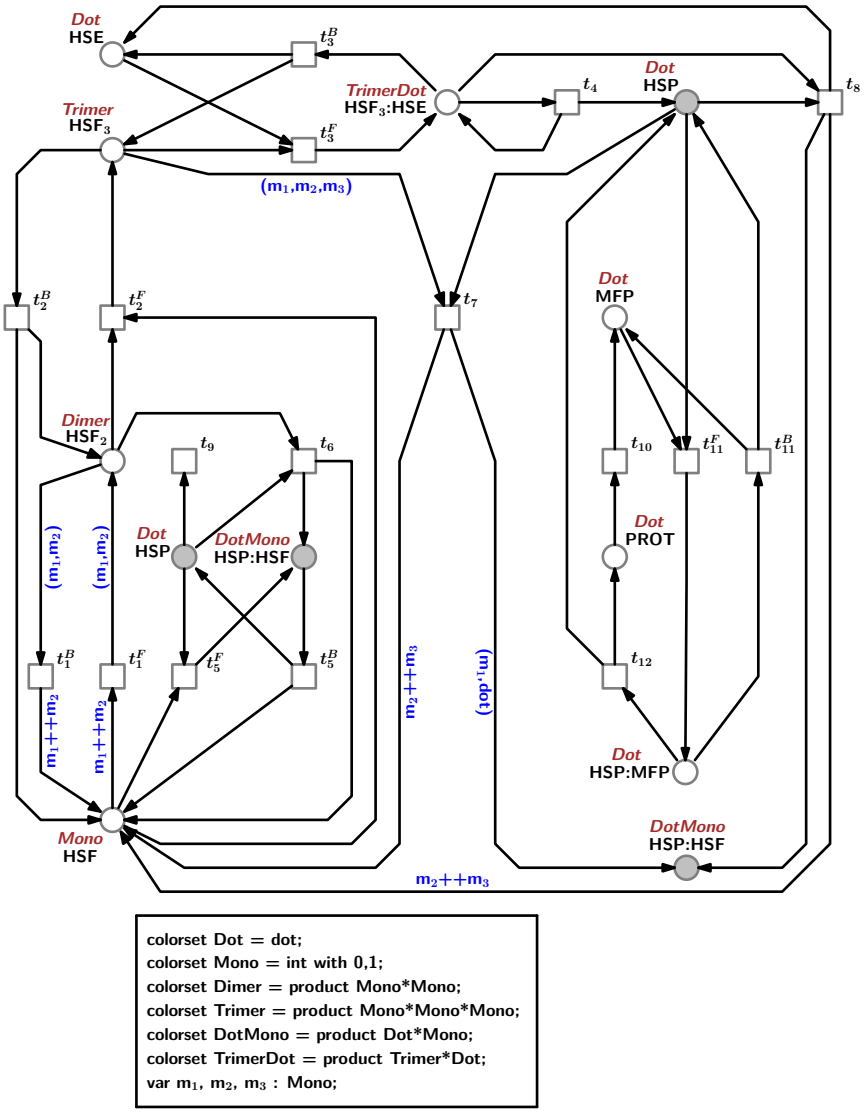


Figure 4.5: Colored Petri net representation of the refined heat shock response model, with the same structure as the initial model. Place (transition) names are displayed next to the corresponding place (transition, resp.). The color set names are depicted with brown italics on top of the place name. The inset contains the color set and variable definition. All places representing compound species have product color sets. Arc expressions are displayed in blue. Most arc expressions have been omitted for readability purposes. Figure generated using Snoopy [68].

5 | Petri Nets with a Compositional Part

We extended Petri nets with a *compositional part* to detail the composition of each place, see [58]. The idea is to capture information about the compositional structure of the entities modeled in a Petri net, both for completeness and for the possibility of automatically assigning color sets in a colored Petri net. We started with the goal of automatizing structural refinement in the Petri net framework, and the formalism allows for automatization of color set definitions for complex places. Having color sets that encode the structure of the elements they are set to represent allows for automatic reflection of changes throughout the network when modifying the color set of an atomic element. We detail in this chapter the construct and its potential for automatization.

5.1 Introduction to Composition (Colored) Petri Nets

As discussed in Chapter 4, colored Petri nets are a good formalism for handling model explosions. However, modelers might find it difficult to choose color sets for the places of the system, and automatizing this procedure would both give a time speedup, and reduce the risk of human errors. We consider that an assignment of color sets based on the composition of places is a good strategy for assigning color sets, because they allow for the identification of constituent elements' colors, and this may prove useful when detailing the reactions, e.g., for a double displacement reaction.

5.1.1 Motivation

We start with an example from biology to unveil the rationale behind the construction. The example shows that the internal structure of the modeled entities plays an important role in the reactions, thus motivating the idea of capturing it in a systematic way.

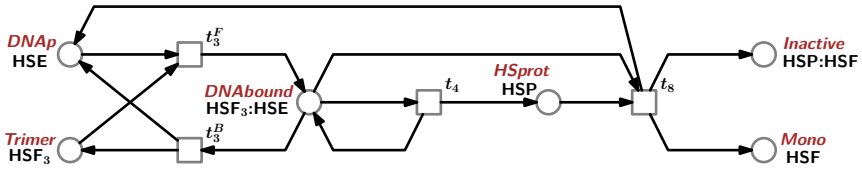
Example 4. *The process by which proteins are produced inside a cell is composed of several steps. The protein is obtained by producing an mRNA molecule based on a template given by the gene encoding the protein, a process called* transcription*, and further production of the protein by ribosomes, process called* translation*. Usually, to enhance the transcription activity, a molecule called a* transcription factor *binds to a special binding region of the gene, and allows for transcription of the intermediary mRNA molecule. Transcription factors may vary largely in form and structure, and in particular they can be complexes formed from several molecules. In our case study of the heat shock response mechanism, the transcription factors for heat shock proteins* (hsf) *must be in a trimeric state* (hsf_3)*. We consider here the reactions capturing the reversible binding of* hsf_3 *to the gene promoting region* hse*, the further production of* hsp*, and also the self-regulating mechanism by which heat shock proteins assist the unbinding of the transcription factor from the* hsp*-encoding gene:*

$$\mathsf{hsf}_3 + \mathsf{hse} \rightleftarrows \mathsf{hsf}_3{:}\mathsf{hse}$$
$$\mathsf{hsf}_3{:}\mathsf{hse} \to \mathsf{hsf}_3{:}\mathsf{hse} + \mathsf{hsp}$$
$$\mathsf{hsp} + \mathsf{hsf}_3{:}\mathsf{hse} \to \mathsf{hsp}{:}\mathsf{hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$$

For the third reaction, it becomes visible that knowing the internal structure (compositional structure) *of each species is important. Namely, the complex species* $\mathsf{hsf}_3{:}\mathsf{hse}$ *is broken down into its constituents. Using color sets that reflect this composition in a colored Petri net representation of this example allows for the identification of the colors for each constituent. We show in Figure 5.1 a colored Petri net model for the three reactions listed here.*

The color sets for places representing complexes were defined based on the color sets of the places representing the elements they are composed of in declaration (a)*, and as simple color sets with one element in declaration* (b)*. Both coloring strategies give a valid model. However, upon refining the color set Mono to contain two colors, the semantics of the model will have to be changed drastically in the* (b) *declaration, since the color sets* Trimer, DNAbound *and* Inactive *will also have to be changed. For the* (a) *declaration, the refinement is reflected in the three aforementioned color sets through their definition.*

As seen in Example 4, the choice of color sets plays a role in the further efforts of refining a model. With our composition Petri nets, we address precisely the issue of assigning color sets when the composition of the modeled entities is known.



Declarations (a):
colorset *DNAp* = enum with hse;
colorset *HSprot* = enum with hsp;
colorset *Mono* = enum with hsf;
colorset *Trimer* = product *Mono*∗*Mono*∗*Mono*;
colorset *DNAbound* = product *Trimer*∗*DNAp*;
colorset *Inactive* = product *HSprot*∗*Mono*;

Declarations (b):
colorset *DNAp* = enum with hse;
colorset *HSprot* = enum with hsp;
colorset *Mono* = enum with hsf;
colorset *Trimer* = enum with trim;
colorset *DNAbound* = enum with bound;
colorset *Inactive* = enum with hsphsf;

Figure 5.1: Colored Petri net representation of a part of the HSR model. Arc expressions are omitted for clarity. Color set declaration (a) uses compound color sets, while color set declaration (b) uses only simple enumeration color sets. Figure generated using Snoopy [68].

5.1.2 Composition Petri Nets (ComP-nets)

We started from the idea that entities in a system can be categorized as *atomic* and *complex*, like detailed in Chapter 3. We include this information in a Petri net by extending the network with a *passive* part, where the transitions (called *composition transitions*) never fire, and their semantics is that of "is-part-of-", namely the pre-places of such a transition represent entities that are part of the entity represented by its post-place. The places of such a Petri net are either *atomic* or *complex*, depending on whether they are the post-place of a composition transition or not.

Definition 7. *[58] A* composition Petri net (ComP-net) *is a tuple* $N = (P, T_c, T, A_c, A, E, I)$ *with the following components:*

1. P, T, A, I *represent respectively the set of places, the set of transitions, the set of arcs and the initialization function.*

2. T_c *is a finite set of* composition transitions *such that* $P \cap T_c = \varnothing$ *and* $T \cap T_c = \varnothing$. *These transitions are used for depicting the composition of places with respect to other places. Composition transitions never fire, irrespective of the marking of the network, and are also called* passive (non-active) transitions. *The regular transitions are, in contrast, called* active.

3. $A_c \subseteq P \times T_c \cup T_c \times P$ *is a set of* composition arcs *such that:*

   - *for any place* $p \in P$, *there is* at most one *incoming composition arc; places with no incoming composition arcs are considered* atomic*;*

- *for every composition transition $t_c \in T_c$ there is* at least one *incoming composition arc and* exactly one *outgoing composition arc;*
- *the graph induced by the composition arcs and the places and transitions they connect is acyclic.*

4. $E : A \cup A_c \to \mathbb{N}_+$ *is an* arc expression *function, such that:*

- *for a composition arc (t_c, p) with $t_c \in T_c$ and $p \in P$, $E(t_c, p) = 1$;*
- *for a composition arc (p, t_c) with $p \in {}^\bullet t_c$, $E(p, t_c)$ can be any positive integer; its interpretation will be that it denotes the number of occurrences of the atomic species represented by p, in the composite species represented by the output place of t_c;*
- *arc expressions of regular arcs have the usual meaning.*

We say that $(P, T_c, A_c, E|_{A_c})$ is the compositional part *of the network, and $(P, T, A, E|_A, I)$ is the* active part *of the network. $E|_S$ denotes the restriction of the arc expression function E to arcs in S. To graphically differentiate between the active and the passive part of a ComP-net, we propose that the composition transitions and the composition arcs are drawn with dashed lines.*

Definition 8. *[58] A place $p \in P$ is said to be a* composition pre-place *of another place $q \in P$ if there exists $t \in T_c$ such that $(p, t) \in A_c$ and $(t, q) \in A_c$.*

The role of the compositional part in a ComP-net is to represent information about the internal structure of (elements represented as) places. Moreover, the composition transitions are passive, which means that they do not contribute to the dynamics of the ComP-net. As such, the properties of standard Petri nets can be generalized to composition Petri nets, considering only the active part of the network. However, this does not render the compositional part useless: ComP-nets are designed so that both the structure and the dynamics of a system can be modeled within the same formalism, yielding a clearer, more detailed representation. For clarity of the graphical representation, the compositional part of the network could also be kept separate from the active part, using logical places. We show an example of a ComP-net model in Figure 5.2.

5.1.3 Composition Colored Petri Nets (ComCP-nets)

We proposed also a colored version of composition Petri nets, *composition colored Petri nets (ComCP-nets)*. The idea is to add the compositional information in a colored Petri net, in such a way that the color set of a complex place is a composite color set comprising the color sets of the (atomic)

places that the complex place consists of. In our definition of ComCP-nets, we propose such composite color sets to be the set of bags (multisets) of colors of tokens from the composition pre-places of the place being assigned that color set, with multiplicities given by the corresponding composition arcs. For example, a place P that has as composition pre-places Q and R such that the cardinality of the composition arc expressions are respectively 1 and 2 will get as color set the set of all bags containing one element of type $C(\mathsf{Q})$ and two elements of type $C(\mathsf{R})$. Using this coloring strategy reflects the composition of places and also sets the ground for an easy full structural refinement, as shown in [58].

Note that ComCP-nets are different from hierarchical colored Petri nets, where several submodels with common actors can be combined in a larger model using interfaces and substitution transitions (where a transition in a module called *page* is substituted with a more detailed description of the transition).

Before introducing the definition of ComCP-nets, we recall several notations we use. $^{++}\sum$ is used to denote multiset addition. $i\backslash c$ with i a natural number and c a color denotes i elements (tokens) with color c. By extension, $i\backslash S$ where S is a color set denotes a multiset of i elements whose colors belong to S.

Definition 9. *[58] A* composition colored Petri net (ComCP-net) *is a tuple* $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ *that satisfies the following requirements:*

1. P, T_c, T, A_c, A *satisfy the constraints of Definition 7.*
2. Σ, V, I *are respectively the set of color sets, the set of variables, and the initialization function.*
3. $C : P \to \Sigma$ *is the* color function *assigning color sets to places such that:*
   - *all atomic places have disjoint color sets, and*
   - *for all complex places* $p \in P$, $C(p) = {}^{++}\sum_{p' \in {}^\bullet t_c} |E(p', t_c)| \backslash C(p')$, *where* t_c *stands for the composition transition encoding the composition of* p, *i.e.,* $t_c^\bullet = \{p\}$, *;*
4. $G : T_c \cup T \to EXPR_V$ *is the* guard function*, such that for each composition transition* $t_c \in T_c$ *with* $t_c^\bullet = \{p\}$ *there exists exactly one binding for which the guard is true for each color in* $C(p)$.
5. $E : A \cup A_c \to EXPR_V$ *is the* arc expression function*, defined such that for every composition transition* $t_c \in T_c$ *with* $t_c^\bullet = \{p\}$*:* $E(t_c, p) = {}^{++}\sum_{p' \in {}^\bullet t_c} E(p', t_c)$.

We say that $(P, T_c, A_c, \Sigma, V, C, G|_{T_c}, E|_{A_c}))$ *is the* compositional part *of the network, and* $(P, T, A, \Sigma, V, C, G|_T, E|_A, I)$ *is the* active part *of the network.*

Assigning color sets in a ComCP-net could be automatized, such that the modeler only assigns color sets to the atomic places, and the color sets of complex places are assigned automatically, based on the color sets of places composing them. We provide an example of a ComCP-net model in Figure 5.3.

5.2 Modeling with Com(C)P-nets

As discussed in Chapter 1, a widely used way of expressing biological models is as a set of reactions. Species can be atomic or complex, where complex species contain at least two instances of one or more atomic species. In order to represent a reaction-based model as a Petri net, species are represented as places, and reactions are represented as transitions, like we have detailed in Chapter 4. This is precisely how a reaction-based model can be represented as the active part of a composition (colored) Petri net. But ComP-nets can further represent the composition of complex species via its passive part, as described next. Each species is already assigned a place. For every complex species, add to the network a composition transition and a composition arc from the composition transition to the place representing that complex species. Further, for each atomic species contained in the complex species, add a composition arc from the place representing that atomic species to the composition transition, with weight precisely the multiplicity of the atomic species in the complex one.

We have discussed in Chapter 3 about complex species as multisets of atomic species only. This is surely not the sole possible approach on representing the internal structure of species. One other way of looking at complex species is as multisets (or even ordered multisets, cartesian products, etc.) of atomic or complex species, such that there are no cyclic definitions of species. We allow in our definition of ComCP-nets for complex places to have both atomic and complex places in their composition, and enforce the acyclicity condition.

In the case of ComCP-nets, the color sets of atomic places can be simple color sets, while for a complex place its color set should be composed of the color sets of the corresponding pre-places. We include next examples of a ComP-net and a ComCP-net that represent reactions $(1) - (2)$ from Table 2.1.

5.2.1 Example of a ComP-net

Example 5. *Consider the first two reactions in the HSR model. The species involved in the reactions are* hsf, hsf_2 *and* hsf_3*, with* hsf_2 *containing two* hsf*s, and* hsf_3 *containing three* hsf*s. The model will thus have one atomic place,* HSF*, to represent* hsf*, and two complex places,* HSF_2 *and* HSF_3 *to represent the dimer and trimer. The compositional part of the network will contain two composition transitions (one for* hsf_2 *and one for* hsf_3*) and the arcs making the connections between the atomic and complex places, as depicted in Figure 5.2.*

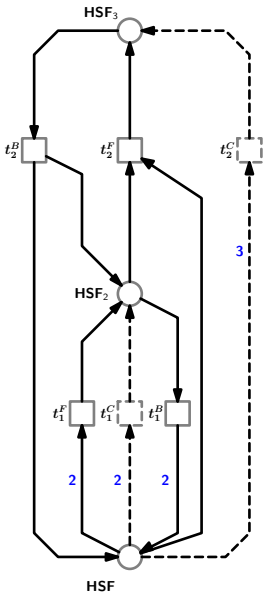


Figure 5.2: ComP-net for the dimerization and trimerization of hsf. Composition transitions and their adjacent arcs are depicted with dotted lines. Only arc expressions greater than 1 are displayed. Figure generated with support of Snoopy [68].

5.2.2 Example of a ComCP-net

Example 6. *Consider the same part of the HSR model as in Example 5. For the colored representation as a compositional Petri net, the main changes are in setting the color sets and adjusting the arc expressions. The model and color set definitions are depicted in Figure 5.3. The complex color sets* Di *and* Tri *are defined as multisets with two and respectively three elements of type* Mono*. With the considered definition of* Mono*, each color set contains only one color.*

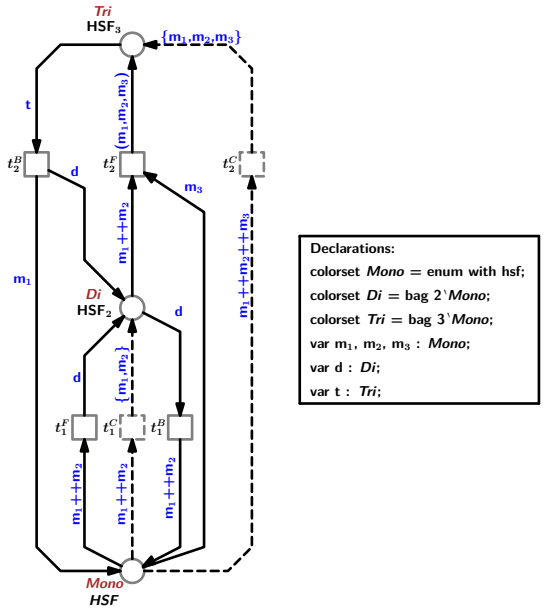


Figure 5.3: ComCP-net for the dimerization and trimerization of hsf. Composition transitions and their adjacent arcs are depicted with dotted lines. Arc expressions are displayed in blue. The inset contains the color set definitions. Figure generated with support of Snoopy [68].

The arc expressions use variables, even though all color sets are singletons. This is meant to facilitate the refinement of the model, and gives what is called a natural coloring *of the network, see [58]. It is important to note that the outgoing arcs from composition transitions use precisely the variables on the incoming arcs of the corresponding composition transition. This ensures that the produced token is composed precisely of the subparts that enter the transition. While seemingly redundant for color sets of only one element, this is of utter importance with color sets of cardinality greater than one, as will be discussed in Section 5.3.*

5.3 Model Refinement with ComCP-nets

The refinements we have implemented in our papers [72, 62] are partial refinements, in the sense that we only include some of the possible refined reactions, not the full structural refinement. We discuss here first how a full structural refinement could be implemented using ComCP-nets, and then show how to exclude from the model some of the reactions, to obtain the desired partial refinement.

A full structural refinement can be implemented as a type refinement of

the color sets to include new details, with arc expressions allowing for all combinations of left- and right- hand side of a reaction, thus not enforcing a condition that the reactants on the left hand side have to be present (possibly in a converted form) on the right hand side of the reaction. This is in line with the reasoning in [57], where such correspondence is not required. At a subsequent step, the modeler can modify the arc expressions and/or transition guards to obtain the desired subset of reactions in the model.

Example 7. *Consider the set of reactions in Example 6, and consider as a refinement that* hsf *has two possible states. Moreover, let us consider that the order of molecules in a complex is not relevant, i.e., the dimer and trimer refine to three and respectively four species, as detailed in Table 2.2. In order to represent the full structural refinement of the model proposed in Figure 5.3, the needed changes are to modify the color set for* HSF*, make sure that the guards of composition transitions only allow for one binding of variables on the arc expressions for each color in the color set of a complex place, and modify the arc expressions in the active part of the network such that all reactions for a full refinement are considered, each exactly one time. The resulting ComCP-net model of the desired full refinement is presented in Figure 5.4. The colors in each color set are listed in Table 5.1.*

The definition of ComCP-nets enforces precisely one decomposition of a complex place into atomic places. Because of this, whenever the expression of a composition arc has cardinality more than one and the color set definition of the adjacent place is not a singleton, a guard is needed on the adjacent composition transition to impose the uniqueness condition. This is why both composition transitions have guards. For $\boldsymbol{t_1^C}$*, there are two possible bindings of* m_1 *and* m_2 *to yield an* HSF_2 *token with color* $\{0,1\}$*:* $\mathsf{m}_1 = 0, \mathsf{m}_2 = 1$ *and* $\mathsf{m}_1 = 1, \mathsf{m}_2 = 0$*. The guard* $G(\boldsymbol{t_1^C})$ *only allows the first binding. The extra guards* $G(\boldsymbol{t_1^F})$ *and* $G(\boldsymbol{t_1^B})$ *are also meant to eliminate duplicates of the same reaction. The model represents a full structural refinement as all possible refined reactions are considered, there are no restrictions on the left- and right- hand side of any reaction, and precisely one binding for each reaction is allowed.*

Table 5.1: List of colors in the color sets of the refined hsf, hsf_2 and hsf_3.

Color set	List of colors
Mono	$\{0,1\}$
Di	$\{\{0,0\},\{0,1\},\{1,1\}\}$
Tri	$\{\{0,0,0\},\{0,0,1\},\{0,1,1\},\{1,1,1\}\}$

At a further step it is possible to restrict the model presented in Example 7 such that only a subset of reactions are modeled. This can be done via

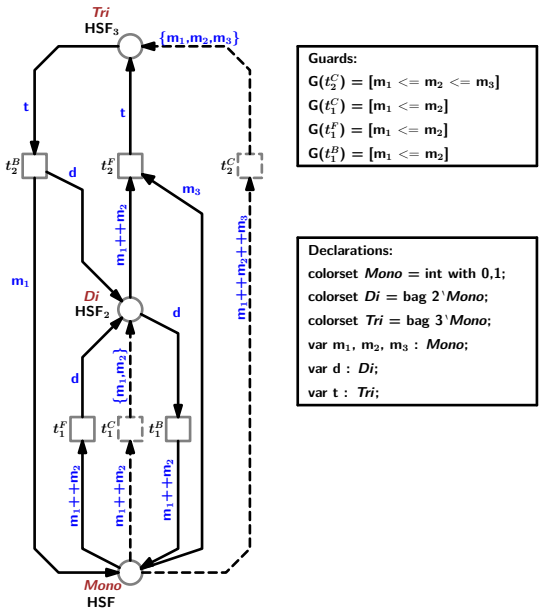


Figure 5.4: ComCP-net of the full structural refinement of reactions (1)-(2) of the HSR model. Composition transitions and their adjacent arcs are depicted with dotted lines. Arc expressions are displayed in blue. The guards that are different than true are listed in the upper inset and the color set definitions are listed in the lower inset. Figure generated with support of Snoopy [68].

guards, arc expressions, or a combination of the two. For example, to enforce a correspondence between the left- and right- hand sides of reaction (1), forward direction, the arc expression of the arc from $\boldsymbol{t_1^F}$ to place $\mathsf{HSF_2}$ should be changed to $\{\mathsf{m1}, \mathsf{m2}\}$ such that the variables on the arc expression from pre-places to the transition are used. Another option is to list in the guard of $\boldsymbol{t_1^F}$ either all variable bindings that are permitted, or the ones that are not allowed, and negate the latter list, e.g., $!(\mathsf{m1} = 0 \& \mathsf{m2} = 1 \& \mathsf{d} = \{1, 1\})$.

The examples in this chapter illustrate how the choice of color sets is important when modeling with colored Petri nets, and having a basis for an automatizable color set declaration reduces the risk of human errors. Moreover, they show how a full structural refinement of a model can be obtained, and provide hints on how to further restrict such a model, in case the modeler is only interested in a subpart of the full structural refinement.

6 | Original Contributions

We present in this chapter our journey in the modeling world, the path forward from one research question to another, and the contribution of each paper included in the thesis.

We started in [61] by presenting how modeling is done in the framework of ordinary differential equations (ODEs), and how a model can be refined in this framework. Due to the explicitness of the formalism, for every new species there is a new variable introduced, and an ODE describing its time evolution. This makes the framework unscalable when it comes to refinements that induce a combinatorial explosion of a model.

We explored next how this combinatorial explosion of a model induced by refinement is handled in several other modeling formalisms on a case study of the heat shock response, see [72]. We considered the rule based modeling formalism, Petri nets, and guarded command languages, to which we added ODEs for a more complete analysis. In two of the frameworks (ODEs and guarded command languages) the model explosion has to be modeled explicitly. The other two frameworks allowed for a compact representation of the refined model, by hiding some of the details in their data structures. The rule based modeling framework was specifically tailored for modeling biological systems, and allows the representation of complex molecules as actors with several internal states and binding sites. Moreover, it is not necessary to specify all the characteristics of a molecule unless important. Allowing partial specifications of molecules is a good strategy to hide the explicit complexity of a model in a concise description. In the framework of standard Petri nets we encountered the same problem of model explosion. But the high-level formalism of colored Petri nets offers a degree of programmability for the type of tokens hosted in places and the arc expressions, thus allowing for a compact representation of the refined model. We proposed two strategies for implementing the refinement: one preserved the network structure of the initial model and used no guards, while the other added some additional transitions and used guards (the same could have been done with using guards only).

Next we focused in [62] on one of the coloring strategies we have explored

in [72], namely the structure-preserving strategy. We have proved that there is a bisimulation relation between the refined model and the initial one, by fully expanding the refined model. This prompted us to investigate what coloring strategies would be most suited for model refinement while preserving the network structure. We then proved in [63] that considering complex species as multisets over a set of atomic species, and refining only atomic species can easily implement a full structural refinement of a model.

The next step was to include information about the internal structure of modeled entities in the Petri net framework, and we did this in [58]. Our proposal is to have compositionality of elements indicated by paths from atomic places to the places that contain them, via what we call *composition transitions*. This is to some extent similar to the idea of the rule based modeling framework of representing the binding sites and states of molecules. Further, we also proposed a coloring strategy for the complex places, driven from the ideas in [63], which makes data refinement straightforward and automatically reflected in the whole model.

We briefly present next the contribution of each paper included in this thesis.

1. Diana-Elena Gratie, Bogdan Iancu, and Ion Petre. *ODE analysis of biological systems*. In: Marco Bernardo, Erik de Vink, Alessandra Di Pierro, and Herbert Wiklicky, editors, *Formal Methods for Dynamical Systems*, volume 7938 of Lecture Notes in Computer Science, pages 29 – 62. Springer Berlin Heidelberg, 2013.

    - This is a review paper offering an introduction to modeling with ODEs and analysis of ODE models.
    - We presented the notions of modeling reaction systems and their translation to a representation as a system of ODEs for some of the most used kinetics (mass action, Michaelis-Menten, enzymatic reactions) from an engineer's point of view.
    - We presented some analysis techniques and exemplified those on our case study.
    - We exemplified the discussed topic on a case study of the heat shock response.

2. Bogdan Iancu, Diana-Elena Gratie, Sepinoud Azimi, and Ion Petre. *On the implementation of quantitative model refinement*. In: Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, Algorithms for Computational Biology, volume 8542 of Lecture Notes in Computer Science, pages 95 – 106. Springer International Publishing, 2014.

- We compared four modeling frameworks (ODEs, rule based models, Petri nets, guarded command languages) with respect to their scalability with regards to model refinements.
- We considered a case study of the eukaryotic heat shock response, modeled it in representative softwares for each of the chosen frameworks, and refined it to include a post-translational modification of one of the proteins involved in the response.
- We concluded that two of the frameworks, namely rule based and colored Petri nets, allow for a compact representation of the refined model, while in the remaining two frameworks the refinement has to be fully specified, with no means of compacting the representation of the refined model.

3. Diana-Elena Gratie and Ion Petre. *Hiding the combinatorial state space explosion of biomodels through colored Petri nets.* Annals of University of Bucharest, LXI:23 – 41. Editura Universității din București, 2014.

   - We presented a Petri net model of the eukaryotic heat shock response, and a possible refinement of it as a colored Petri net that preserves the network structure.
   - We proved that the initial model and its refinement are equivalent, namely that the refined model is in a bisimulation relation with the initial model.
   - We concluded that colored Petri nets allow for a compact representation of the given refinement, thus hiding the model explosion inherent to refinement.

4. Diana-Elena Gratie and Ion Petre. *Full structural model refinement as type refinement of colored Petri nets.* In: Monika Heiner and Annegret K. Wagler, editors, Proceedings of the 6th International Workshop on Biological Processes and Petri Nets, volume 1373 of Ceur Workshop Proceedings, page 70 – 84. CEUR Workshop Proceedings, 2015.

   - We considered the notion of type refinement of colored Petri nets, namely the modification of color sets (and possibly of arc expressions) in a colored Petri net model.
   - We showed that it can be used to automatize generating a full structural refinement of a model, with a proper choice of initial color sets.

5. Diana-Elena Gratie and Cristian Gratie. *Composition colored Petri nets for the refinement of reaction-based models.* Electronic Notes in Theoretical Computer Science, in press.

- We extended Petri nets with a compositional part, both for standard Petri nets (we call such networks composition Petri nets, ComP-nets) and colored Petri nets (which are in turn called composition colored Petri nets or ComCP-nets). The compositional part includes information about the internal structure of the actors represented in the places of the network.
- We gave an algorithm for implementing a model as a composition (colored) Petri net.
- We proposed a natural coloring of a ComCP-net, and showed how it can be used to easily provide a full structural refinement of the model.

7 | Discussion

We start this chapter with a few punctual extension ideas for the work presented in this thesis, and continue with a more general overview of the field and the current challenges of (bio)modeling.

The work included in this thesis focused on our contributions in modeling strategies for refinement. As discussed in the previous chapters, many modeling frameworks exist that are tailored for biomodeling, and out of these some are more suitable for refinements than others. The more suitable ones scale up to some extent by allowing compact representations of systems by packing information in some internal structures. In the framework of rule based modeling, this is done via partial descriptions of actors, while in the framework of colored Petri nets it is done via colors. Our proposal of composition (colored) Petri nets is meant as a step towards the automatization of model refinement, while increasing the expressiveness of the network model. Refinements can be seen as a scaling up of a model, but the limitations of the Petri net framework are not improved (in terms of simulations and state space limitations).

A natural continuation of the work presented here is an implementation of the compositional part as a plugin of existing Petri net software. An interesting research direction with ComCP-nets is exploring how introducing guards on the composition transitions can be used to automatically restrict the guards of active transitions. Namely, how the presence of a non-trivial guard on a composition transition having p as only post-place can add the same restrictions on the guards of all pre-transitions of p, thus limiting the possible transitions.

Another interesting research direction is to address the reverse problem, namely to group places together in an abstract place when they have similar connections to transitions, in order to get a coarser model. In a biological setting this could mean for example grouping several proteins from the same family in a generic protein fulfilling the same function. This can be useful when a detailed model is available, but its integration into a larger system requires dropping some details in order to ensure communication with the rest of the system. It would also contribute to the transition between

different levels of detail of a model.

The field of systems biology faces many challenges ahead, of which the most pregnant is the scalability problem. As advances in genomics make it possible to model more and more details of biological systems, scaling up previous models is not an easy task. Modeling frameworks come with advantages, but also limitations, and pushing the limits of these constraints is a demanding process.

Current modeling efforts address the problem of developing whole cell models, with the current approach being to simulate different processes in different frameworks, and to integrate them by establishing a communication protocol between the individual parts of the whole system, see [80, 81]. The reasoning behind the model integration strategy is that different cellular processes are best described by different representations. Extra difficulty is added by the fact that different processes may include common components on different levels of detail. Thus, the process of integrating models is not only one of integrating different modeling methods, but also of establishing communication between different specifications of the same species. Our formalism ComCP-nets addresses the problem of considering species on different levels of detail by allowing a specification of the internal structure of species. Another aspect that would make it a good candidate for modeling to-be-integrated parts of a system is that Petri nets allow stochastic, continuous and hybrid simulations.

Another topical research modeling effort is the brain project, with international large consortia trying to decypher the neurological functioning mechanisms. Models at different levels of resolution and using different modeling and simulation theories (e.g., quantum physics, Newtonian physics, Brownian motion) are joined in large-scale models that capture more details and submodules than the individual parts to analyze emerging system-level properties. The integration and multi-scale modeling efforts are common to both the qualitative and the quantitative modeling communities. The rationale behind integrating models at different resolutions as opposed to modeling the whole system at the same level of resolution is that for some subprocesses in the system the drivers are at a very low level and thus need to be modeled, while for the rest of the system a higher level of resolution captures the desired behavior. Modeling everything at a very low level is unfeasible, even with great computational support. With hardware and complexity limits in sight, model integration is a good solution for systems modeling.

Bibliography

[1] Jean-Raymond Abrial. *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, 2010.

[2] Alan Aderem. Systems Biology: Its Practice and Challenges. *Cell*, 121(4):511–513, 2005.

[3] Tilak Agerwala. Special Feature: Putting Petri Nets to Work. *Computer*, 12(12):85–94, 1979.

[4] Tilak Agerwala and Mike Flynn. Comments on Capabilities, Limitations and "Correctness" of Petri Nets. In Stephen Anthony Szygenda, editor, *Proceedings of the 1st Annual Symposium on Computer Architecture*, ISCA '73, pages 81–86. ACM, 1973.

[5] Marco Ajmone Marsan, Gianni Conte, and Gianfranco Balbo. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Transactions on Computer Systems (TOCS)*, 2(2):93–122, 1984.

[6] Hassane Alla and René David. Continuous and Hybrid Petri Nets. *Journal of Circuits, Systems, and Computers*, 8(01):159–188, 1998.

[7] Rui Alves, Fernando Antunes, and Armindo Salvador. Tools for kinetic modeling of biochemical networks. *Nature Biotechnology*, 24(6):667–672, 2006.

[8] Bastian R. Angermann, Frederick Klauschen, Alex D. Garcia, Thorsten Prustel, Fengkai Zhang, Ronald N. Germain, and Martin Meier-Schellersheim. Computational modeling of cellular signaling processes embedded into dynamic spatial contexts. *Nature Methods*, 9(3):283–289, 2012.

[9] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[10] BioVisions at Harvard University. The Inner Life. `http://multimedia.mcb.harvard.edu/`, 2016. Accessed: 2016-09-13.

[11] Sepinoud Azimi, Eugen Czeizler, Cristian Gratie, Diana Gratie, Bogdan Iancu, Nebiat Ibssa, Ion Petre, Vladimir Rogojin, Tolou Shadbahr, and Fatemeh Shokri. An Excursion Through Quantitative Model Refinement. In *Membrane Computing: 16th International Conference, CMC 2015, Valencia, Spain, August 17-21, 2015, Revised Selected Papers*. Springer International Publishing, 2015.

[12] Ralph-Johan Back and Joakim von Wright. *Refinement Calculus: A Systematic Introduction*. Texts in Computer Science. Springer-Verlag New York, 1998.

[13] Eric Badouel, Marisa Llorens, and Javier Oliver. Modeling Concurrent Systems: Reconfigurable Nets. In *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'03)*, pages 1568–1574, 2003.

[14] Eric Badouel and Javier Oliver. Reconfigurable Nets, a Class of High Level Petri Nets Supporting Dynamic Changes within Workflow Systems. Technical Report 3339, Institut national de recherche en informatique et en automatique, 1998.

[15] Jos C. M. Baeten. A Brief History of Process Algebra. *Theoretical Computer Science*, 335(2–3):131–146, 2005.

[16] Paolo Baldan, Nicoletta Cocco, Andrea Marin, and Marta Simeoni. Petri Nets for Modelling Metabolic Pathways: A Survey. *Natural Computing*, 9(4):955–989, 2010.

[17] Ezio Bartocci and Pietro Lió. Computational Modeling, Formal Analysis, and Tools for Systems Biology. *PLoS Computational Biology*, 12(1):1–22, 2016.

[18] Don Batory, Jacob Neal Sarvela, and Axel Rauschmayer. Scaling Step-Wise Refinement. *IEEE Transactions on Software Engineering*, 30(6):355–371, 2004.

[19] Jan A. Bergstra and Jan Willem Klop. Process Algebra for Synchronous Communication. *Information and Control*, 60(1-3):109–137, 1984.

[20] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. A Static Analyzer for Large Safety-critical Software. In Ron Cytron

and Rajiv Gupta, editors, *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI'03)*, volume 38, pages 196–207. ACM, 2003.

[21] Mary Ann Blätke, Monika Heiner, and Wolfgang Marwan. *Algebraic and Discrete Mathematical Methods for Modern Biology*, chapter BioModel Engineering with Petri Nets. Academic Press, 2015.

[22] Jonathan Bowen and Victoria Stavridou. Safety-Critical Systems, Formal Methods and Standards. *Software Engineering Journal*, 8(4):189–209, 1993.

[23] Alvis Brazma, Maria Krestyaninova, and Ugis Sarkans. Standards for systems biology. *Nature Reviews Genetics*, 7(8):593–605, 2006.

[24] Stephen D. Brookes, Charles Anthony Richard Hoare, and Andrew William Roscoe. A Theory of Communicating Sequential Processes. *Journal of the Association for Computing Machinery*, 31(3):560–599, 1984.

[25] Muffy Calder, Vladislav Vyshemirsky, David Gilbert, and Richard 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*. Springer Berlin Heidelberg, 2006.

[26] Carlos Canal, Ernesto Pimentel, and José M Troya. Specification and Refinement of Dynamic Software Architectures. In Patrick Donohoe, editor, *Software Architecture*, volume 140 of *IFIP – The International Federation for Information Processing*, pages 107–126. Springer US, 1999.

[27] Deepak Chandran, Wilbert B. Copeland, Sean C. Sleight, and Herbert M. Sauro. Mathematical modeling and synthetic biology. *Drug Discovery Today: Disease Models*, 5(4):299–309, 2008.

[28] Claudine Chaouiya. Petri net modelling of biological networks. *Briefings in Bioinformatics*, 8(4):210–219, 2007.

[29] Shyi-Ming Chen, Jyh-Sheng Ke, and Jin-Fu Chang. Knowledge Representation Using Fuzzy Petri Nets. *IEEE Transactions on Knowledge and Data Engineering*, 2(3):311–319, 1990.

[30] Christine Choppy, Laure Petrucci, and Alfred Sanogo. Coloured Petri Nets Refinements. In Daniel Moldt, editor, *Joint Proceedings of the International Workshop on Petri Nets and Software Engineering*

(PNSE'13) and the International Workshop on Modeling and Business Environments (ModBE'13), volume 989 of *CEUR Workshop Proceedings*, pages 187–201. CEUR Workshop Proceedings, 2013.

[31] Lily A. Chylek, Leonard A. Harris, Chang-Shung Tung, James R. Faeder, Carlos F. Lopez, and William S. Hlavacek. Rule-based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 6(1):13–36, 2014.

[32] Federica Ciocchetta and Jane Hillston. Bio-PEPA: An Extension of the Process Algebra PEPA for Biochemical Networks. *Electronic Notes in Theoretical Computer Science*, 194(3):103–117, 2008.

[33] Federica Ciocchetta and Jane Hillston. Process algebras in systems biology. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008 Bertinoro, Italy, June 2-7, 2008 Advanced Lectures*, volume 5016 of *Lecture Notes in Computer Science*, pages 265–312. Springer Berlin Heidelberg, 2008.

[34] David E. Clark and Stephen D. Pickett. Computational methods for the prediction of 'drug-likeness'. *Drug Discovery Today*, 5(2):49–58, 2000.

[35] Fabricio F. Costa. Big data in biomedicine. *Drug Discovery Today*, 19(4):433–440, 2014.

[36] Gheorghe Craciun and Casian Pantea. Identifiability of chemical reaction networks. *Journal of Mathematical Chemistry*, 44(1):244–259, 2008.

[37] Elena Czeizler, Eugen Czeizler, Bogdan Iancu, and Ion Petre. Quantitative Model Refinement as a Solution to the Combinatorial Size Explosion of Biomodels. *Electronic Notes in Theoretical Computer Science*, 284:35–53, 2012.

[38] Eugen Czeizler, Vladimir Rogojin, and Ion Petre. The Phosphorylation of the Heat Shock Factor as a Modulator for the Heat Shock Response. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(5):1326–1337, 2012.

[39] Vincent Danos, Jérôme Feret, Walter Fontana, Russ Harmer, and Jean Krivine. Rule-Based Modelling and Model Perturbation. In Corrado Priami, Ralph-Johan Back, and Ion Petre, editors, *Transactions on*

Computational Systems Biology XI, volume 5750 of *Lecture Notes in Computer Science*, pages 116–137. Springer Berlin Heidelberg, 2009.

[40] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-Based Modelling of Cellular Signalling. In Luís Caires and Vasco T. Vasconcelos, editors, *CONCUR 2007 – Concurrency Theory: 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007. Proceedings*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41. Springer Berlin Heidelberg, 2007.

[41] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-Based Modelling, Symmetries, Refinements. In Jasmin Fisher, editor, *Formal Methods in Systems Biology: First International Workshop, FMSB 2008, Cambridge, UK, June 4-5, 2008. Proceedings*, volume 5054 of *Lecture Notes in Computer Science*, pages 1103–122. Springer Berlin Heidelberg, 2008.

[42] René David and Hassane Alla. Continuous Petri nets. In *8th European Workshop on Application and Theory of Petri Nets*, volume 340, pages 275–294, 1987.

[43] René David and Hassane Alla. Petri nets for modeling of dynamic systems: A survey. *Automatica*, 30(2):175–202, 1994.

[44] René David and Hassane Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer Berlin Heidelberg, 2005.

[45] Lorenzo Dematté, Corrado Priami, and Alessandro Romanel. The Beta Workbench: a computational tool to study the dynamics of biological systems. *Briefings in Bioinformatics*, 9(5):437–449, 2008.

[46] Jörg Desel and Gabriel Juhás. "What Is a Petri Net?" Informal Answers for the Informed Reader. In Hartmut Ehrig, Julia Padberg, Gabriel Juhás, and Grzegorz Rozenberg, editors, *Unifying Petri Nets: Advances in Petri Nets*, volume 2128 of *Lecture Notes in Computer Science*, pages 1–25. Springer Berlin Heidelberg, 2001.

[47] Nesma ElKalaawy and Amr Wassal. Methodologies for the modeling and simulation of biochemical networks, illustrated for signal transduction pathways: A primer. *Biosystems*, 129:1–18, 2015.

[48] Philipp Erdrich, Henning Knoop, Ralf Steuer, and Steffen Klamt. Cyanobacterial biofuels: new insights and strain design strategies revealed by computational modeling. *Microbial Cell Factories*, 13(1):1–15, 2014.

[49] James R. Faeder, Michael L. Blinov, and William S. Hlavacek. Rule-Based Modeling of Biochemical Systems with BioNetGen. In Ivan V. Maly, editor, *Systems Biology*, volume 500 of *Methods in Molecular Biology*, pages 113–167. Humana Press, 2009.

[50] Jasmin Fisher and Thomas A. Henzinger. Executable cell biology. *Nature Biotechnology*, 25(11):1239–1249, 2007.

[51] Wan Fokkink. *Introduction to Process Algebra*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2000.

[52] Carmen Garrido, Sandeep Gurbuxani, Luigi Ravagnan, and Guido Kroemer. Heat Shock Proteins: Endogenous Modulators of Apoptotic Cell Death. *Biochemical and Biophysical Research Communications*, 286(3):433–442, 2001.

[53] Hartmann J. Genrich and Kurt Lautenbach. System modelling with high-level Petri nets. *Theoretical Computer Science*, 13(1):109–135, 1981.

[54] Geoffrey S. Ginsburg and Jeanette J. McCarthy. Personalized medicine: revolutionizing drug discovery and patient care. *Trends in Biotechnology*, 19(12):491–496, 2001.

[55] Assaf Gottlieb, Gideon Y. Stein, Eytan Ruppin, and Roded Sharan. PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology*, 7(1), 2011.

[56] Susanne Graf and Hassen Saidi. Construction of Abstract State Graphs with PVS. In *Computer Aided Verification: 9th International Conference, CAV'97 Haifa, Israel, June 22–25, 1997 Proceedings*, volume 1254 of *Lecture Notes in Computer Science*, pages 72–83. Springer Berlin Heidelberg, 1997.

[57] Cristian Gratie and Ion Petre. Fit-Preserving Data Refinement of Mass-Action Reaction Networks. In Arnold Beckmann, Erzsébet Csuhaj-Varjú, and Klaus Meer, editors, *Language, Life, Limits: 10th Conference on Computability in Europe, CiE 2014, Budapest, Hungary, June 23-27, 2014. Proceedings*, volume 8493 of *Lecture Notes in Computer Science*, pages 204–213. Springer International Publishing, 2014.

[58] Diana-Elena Gratie and Cristian Gratie. Composition Colored Petri Nets for the Refinement of Reaction-Based Models. *Electronic Notes in Theoretical Computer Science*, 326C:51–72, in press.

[59] Diana-Elena Gratie, Bogdan Iancu, Sepinoud Azimi, and Ion Petre. Quantitative Model Refinement in Four Different Frameworks, with Applications to the Heat Shock Response. Technical Report 1067, Turku Centre for Computer Science, 2013.

[60] Diana-Elena Gratie, Bogdan Iancu, Sepinoud Azimi, and Ion Petre. Quantitative Model Refinement in Four Different Frameworks, with Applications to the Heat Shock Response. In Luigia Petre and Emil Sekerinski, editors, *From Action Systems to Distributed Systems*, pages 201–214. Chapman and Hall/CRC, 2016.

[61] Diana-Elena Gratie, Bogdan Iancu, and Ion Petre. ODE Analysis of Biological Systems. In Marco Bernardo, Erik de Vink, Alessandra Di Pierro, and Herbert Wiklicky, editors, *Formal Methods for Dynamical Systems*, volume 7938 of *Lecture Notes in Computer Science*, pages 29–62. Springer Berlin Heidelberg, 2013.

[62] Diana-Elena Gratie and Ion Petre. Hiding the Combinatorial State Space Explosion of Biomodels through Colored Petri Nets. *Annals of University of Bucharest*, LXI:23–41, 2014.

[63] Diana-Elena Gratie and Ion Petre. Full Structural Model Refinement as Type Refinement of Colored Petri Nets. In Monika Heiner and Annegret K. Wagler, editors, *Proceedings of the 6th International Workshop on Biological Processes and Petri Nets*, volume 1373 of *CEUR Workshop Proceedings*, pages 70–84. CEUR Workshop Proceedings, 2015.

[64] Cato Maximilian Guldberg and Peter Waage. Ueber die chemische affinität. § 1. Einleitung. *Journal für Praktische Chemie*, 19(1):69–114, 1879.

[65] Simon Hardy and Pierre N. Robillard. Modeling and simulation of molecular biology systems using Petri nets: modeling goals of various approaches. *Journal of Bioinformatics and Computational Biology*, 2(4):619–637, 2004.

[66] David Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, 1987.

[67] Monika Heiner, David Gilbert, and Robin Donaldson. Petri Nets for Systems and Synthetic Biology. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer Berlin Heidelberg, 2008.

[68] Monika Heiner, Mostafa Herajy, Fei Liu, Christian Rohr, and Martin Schwarick. Snoopy – A Unifying Petri Net Tool. In *Application and Theory of Petri Nets: 33rd International Conference, PETRI NETS 2012, Hamburg, Germany, June 25-29, 2012. Proceedings*, volume 7347 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012.

[69] Monika Heiner, Ina Koch, and Jürgen Will. Model validation of biological pathways using Petri nets – demonstrated for apoptosis. *Biosystems*, 75(1):15–28, 2004.

[70] Michael Hucka, Andrew Finney, Herbert M. Sauro, Hamid Bolouri, John C. Doyle, Hiroaki Kitano, Adam P. Arkin, Benjamin J. Bornstein, Dennis Bray, Athel Cornish-Bowden, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[71] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[72] Bogdan Iancu, Diana-Elena Gratie, Sepinoud Azimi, and Ion Petre. On the Implementation of Quantitative Model Refinement. In Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, *Algorithms for Computational Biology*, volume 8542 of *Lecture Notes in Computer Science*, pages 95–106. Springer International Publishing, 2014.

[73] Kurt Jensen. Coloured Petri nets and the invariant-method. *Theoretical Computer Science*, 14(3):317–336, 1981.

[74] Kurt Jensen. Coloured Petri nets. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Petri Nets: Central Models and Their Properties*, volume 254 of *Lecture Notes in Computer Science*, pages 248–299. Springer Berlin Heidelberg, 1987.

[75] Kurt Jensen. Coloured Petri Nets: A High Level Language for System Design and Analysis. In Grzegorz Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 342–416. Springer Berlin Heidelberg, 1991.

[76] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use Volume 1*. EATCS Monographs on Theoretical Computer Science. Springer Berlin Heidelberg, 1992.

[77] Kurt Jensen and Lars M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer Berlin Heidelberg, 2009.

[78] Kurt Jensen and Grzegorz Rozenberg, editors. *High-level Petri Nets: Theory and Applications*. Springer Berlin Heidelberg, 1991.

[79] Izet M. Kapetanovic. Computer-aided drug discovery and development (CADDD): In silico-chemico-biological approach. *Chemico-Biological Interactions*, 171(2):165–176, 2008. Frontiers of Pharmacology and Toxicology.

[80] Jonathan R. Karr, Jayodita C. Sanghvi, Derek N. Macklin, Miriam V. Gutschow, Jared M. Jacobs, Benjamin Bolival, Nacyra Assad-Garcia, John I. Glass, and Markus W. Covert. A Whole-Cell Computational Model Predicts Phenotype from Genotype. *Cell*, 150(2):389–401, 2012.

[81] Jonathan R. Karr, Koichi Takahashi, and Akira Funahashi. The principles of whole-cell modeling. *Current Opinion in Microbiology*, 27:18–24, 2015.

[82] Hiroaki Kitano, Akira Funahashi, Yukiko Matsuoka, and Kanae Oda. Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology*, 23(8):961–966, 2005.

[83] Cecilia Klein, Andrea Marino, Marie-France Sagot, Paulo Vieira Milreu, and Matteo Brilli. Structural and dynamical analysis of biological networks. *Briefings in Functional Genomics*, 11(6):420–433, 2012.

[84] Ina Koch, Wolfgang Reisig, and Falk Schreiber, editors. *Modeling in Systems Biology: The Petri Net Approach*, volume 16 of *Computational Biology*. Springer London, 2011.

[85] Christina Kossow, Tobias Helms, Jan M. Kreutzer, Alke Martens, and Adelinde M. Uhrmacher. Evaluating Different Modeling Languages Based on a User Study. In *ANSS '16 Proceedings of the 49th Annual Simulation Symposium*, pages 18:1–18:8. Society for Computer Simulation International, 2016.

[86] Marta Kwiatkowska, Gethin Norman, and David Parker. Using Probabilistic Model Checking in Systems Biology. *ACM SIGMETRICS Performance Evaluation Review*, 35(4):14–21, March 2008.

[87] Charles Lakos. From Coloured Petri Nets to Object Petri Nets. In Giorgio De Michelis and Michel Diaz, editors, *Application and Theory of Petri Nets 1995: 16th International Conference Turin, Italy, June*

26?30, 1995 Proceedings, volume 935 of *Lecture Notes in Computer Science*, pages 278–297. Springer Berlin Heidelberg, 1995.

[88] Charles Lakos. Composing Abstractions of Coloured Petri Nets. In Mogens Nielsen and Dan Simpson, editors, *Application and Theory of Petri Nets 2000: 21st International Conference, ICATPN 2000 Aarhus, Denmark, June 26?30, 2000 Proceedings*, volume 1825 of *Lecture Notes in Computer Science*, pages 323–342. Springer Berlin Heidelberg, 2000.

[89] Morgan G.I. Langille and Jonathan A. Eisen. BioTorrents: A File Sharing Service for Scientific Data. *PLoS One*, 5(4):e10071, 2010.

[90] Nicolas Le Novère, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I Aladjem, Sarala M Wimalaratne, et al. The systems biology graphical notation. *Nature Biotechnology*, 27(8):735–741, 2009.

[91] Susan Lindquist. The heat-shock response. *Annual Review of Biochemistry*, 55(1):1151–1191, 1986.

[92] Carl G. Looney. Fuzzy Petri nets for rule-based decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):178–183, January/February 1988.

[93] Carlos F. Lopez, Jeremy L. Muhlich, John A. Bachman, and Peter K. Sorger. Programming biological models in Python using PySB. *Molecular Systems Biology*, 9(1):646, 2013.

[94] Daniel Machado, Rafael S. Costa, Miguel Rocha, Eugénio C. Ferreira, Bruce Tidor, and Isabel Rocha. Modeling formalisms in Systems Biology. *AMB Express*, 1(1):1–14, 2011.

[95] Thomas Madej, Christopher J. Lanczycki, Dachuan Zhang, Paul A. Thiessen, Renata C. Geer, Aron Marchler-Bauer, and Stephen H. Bryant. MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Research*, 42(D1):D297–D303, 2013.

[96] Vivien Marx. Biology: The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

[97] Campbell McInnes. Virtual screening strategies in drug discovery. *Current Opinion in Chemical Biology*, 11(5):494–502, 2007. Analytical Techniques / Mechanisms.

[98] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1980.

[99] Andrzej Mizera and Barbara Gambin. Stochastic modelling of the eukaryotic heat shock response. *Journal of Theoretical Biology*, 265(3):455–466, 2010.

[100] Alex Mogilner and David Odde. Modeling cellular processes in 3D. *Trends in Cell Biology*, 21(12):692–700, 2011.

[101] Michael K. Molloy. Performance Analysis Using Stochastic Petri Nets. *IEEE Transactions on Computers*, C-31(9):913–917, 1982.

[102] Tadao Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[103] Laura O'Hara, Alessandra Livigni, Thanos Theo, Benjamin Boyer, Tim Angus, Derek Wright, Sz-Hau Chen, Sobia Raza, Mark W. Barnett, Paul Digard, et al. Modelling the structure and dynamics of biological pathways. *PLoS Biol*, 14(8):e1002530, 2016.

[104] Julia Padberg and Milan Urbášek. Rule-based refinement of Petri nets: A survey. In Hartmut Ehrig, Wolfgang Reisig, Grzegorz Rozenberg, and Herbert Weber, editors, *Petri Net Technology for Communication-Based Systems*, volume 2472 of *Lecture Notes in Computer Science*, pages 161–196. Springer Berlin Heidelberg, 2003.

[105] Michael Pedersen, Andrew Phillips, and Gordon D. Plotkin. A High-Level Language for Rule-Based Modelling. *PLoS ONE*, 10(6):e0114296, June 2015.

[106] James L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Inc., 1981.

[107] Ion Petre, Andrzej Mizera, Claire L. Hyder, Annika Meinander, Andrey Mikhailov, Richard I. Morimoto, Lea Sistonen, John E. Eriksson, and Ralph-Johan Back. A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. *Natural Computing*, 10(1):595–612, 2011.

[108] Carl Adam Petri. *Kommunikation mit automaten*. PhD thesis, Institut für Instrumentelle Mathematik, 1962.

[109] Petri Nets Tools Database Quick Overview. `https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html`. Accessed: 2016-09-13.

[110] Andrew Phillips and Luca Cardelli. Efficient, Correct Simulation of Biological Processes in the Stochastic Pi-calculus. In Muffy Calder and Stephen Gilmore, editors, *Computational Methods in Systems Biology: International Conference CMSB 2007, Edinburgh, Scotland, September 20-21, 2007. Proceedings*, volume 4695 of *Lecture Notes in Computer Science*, pages 184–199. Springer Berlin Heidelberg, 2007.

[111] Russell A. Poldrack and Krzysztof J. Gorgolewski. Making big data open: data sharing in neuroimaging. *Nature Neuroscience*, 17(11):1510–1517, 2014.

[112] Corrado Priami and Melissa J. Morine. *Analysis of Biological Systems*. World Scientific, 2015.

[113] Naira Quintana, Frank Van der Kooy, Miranda D. Van de Rhee, Gerben P. Voshol, and Robert Verpoorte. Renewable energy from Cyanobacteria: energy production optimization by metabolic pathway engineering. *Applied Microbiology and Biotechnology*, 91(3):471–490, 2011.

[114] Chander Ramchandani. *Analysis of asynchronous concurrent systems by timed Petri nets.* PhD thesis, Massachusetts Institute of Technology, 1973.

[115] Venkatramana N. Reddy, Michael N. Liebman, and Michael L. Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, 26(1):9–24, 1996.

[116] Venkatramana N. Reddy, Michael L. Mavrovouniotis, and Michael N. Liebman. Petri Net Representations in Metabolic Pathways. In *ISMB-93 Proceedings*, pages 328–336, 1993.

[117] Wolfgang Reisig. *Petri Nets: An Introduction*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag Berlin Heidelberg, 1985.

[118] Wolfgang Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer Berlin Heidelberg, 2013.

[119] Theodore R. Rieger, Richard I. Morimoto, and Vassily Hatzimanikatis. Mathematical Modeling of the Eukaryotic Heat-Shock Response: Dynamics of the hsp70 Promoter. *Biophysical Journal*, 88(3):1646–1658, 2005.

[120] The Royal Swedish Academy of Sciences. Scientific Background on the Nobel Prize in Chemistry 2013. `http://www.nobelprize.`

org/nobel_prizes/chemistry/laureates/2013/advanced-chemistryprize2013.pdf, 2013. Accessed: 2016-09-13.

[121] Andrea Sackmann, Monika Heiner, and Ina Koch. Application of Petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics*, 7(1):482, 2006.

[122] Henning Schmidt and Mats Jirstrand. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics*, 22(4):514–515, 2006.

[123] Tirthankar Sengupta, Mani Bhushan, and Pramod P. Wangikar. Metabolic modeling for multi-objective optimization of ethanol production in a Synechocystis mutant. *Photosynthesis Research*, 118(1):155–165, 2013.

[124] Colin Snook and Michael Butler. UML-B: Formal Modeling and Design Aided by UML. *ACM Transactions on Software Engineering and Methodology*, 15(1):92–122, 2006.

[125] Kansuporn Sriyudthsak, Fumihide Shiraishi, and Masami Yokota Hirai. Mathematical Modeling and Dynamic Simulation of Metabolic Reaction Systems Using Metabolome Time Series Data. *Frontiers in Molecular Biosciences*, 3, 2016.

[126] Lena Strömbäck, David Hall, and Patrick Lambrix. A review of standards for data exchange within systems biology. *Proteomics*, 7(6):857–867, 2007.

[127] Ichiro Suzuki and Tadao Murata. A method for stepwise refinement and abstraction of Petri nets. *Journal of Computer and System Sciences*, 27(1):51–76, 1983.

[128] Zuzanna Szymańska and Maciej Zylicz. Mathematical modeling of heat shock protein synthesis in response to temperature change. *Journal of Theoretical Biology*, 259(3):562–569, 2009.

[129] Robert Valette. Analysis of Petri Nets by Stepwise Refinements. *Journal of Computer and System Sciences*, 18(1):35–46, 1979.

[130] Rüdiger Valk. Self-modifying nets, a natural extension of Petri nets. In Giorgio Ausiello and Corrado Böhm, editors, *Automata, Languages and Programming*, number 62 in LectureNotes in Computer Science, pages 464–476. Springer, 1978.

[131] Walter Vogler. Behaviour preserving refinements of Petri nets. In Gottfried Tinhofer and Gunther Schmidt, editors, *Graph-Theoretic Concepts in Computer Science*, volume 246 of *Lecture Notes in Computer Science*, pages 82–93. Springer Berlin Heidelberg, 1987.

[132] Eberhard O. Voit, Harald A. Martens, and Stig W. Omholt. 150 Years of the Mass Action Law. *PLoS Computational Biology*, 11(1):1–7, 2015.

[133] Jiacun Wang. *Timed Petri Nets: Theory and Application*, volume 9. Springer Science & Business Media, 2012.

[134] Sandy D. Westerheide, Julius Anckar, Stanley M. Stevens, Lea Sistonen, and Richard I. Morimoto. Stress-inducible regulation of heat shock factor 1 by the deacetylase SIRT1. *Science*, 323(5917):1063–1066, 2009.

[135] Andrea D. Weston and Leroy Hood. Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine. *Journal of Proteome Research*, 3(2):179–196, 2004.

[136] John C. Wooley and Herbert S. Lin, editors. *Catalyzing Inquiry at the Interface of Computing and Biology*, chapter Computational Modeling and Simulation as Enablers for Biological Discovery, pages 117–204. National Academies Press (US), 2005.

[137] Yuting Zheng and Ganesh Sriram. Mathematical Modeling: Bridging the Gap between Concept and Realization in Synthetic Biology. *BioMed Research International*, 2010, 2010.

Complete List of Publications

1. Diana-Elena Gratie, Bogdan Iancu, and Ion Petre. ODE analysis of biological systems. In: Marco Bernardo, Erik de Vink, Alessandra Di Pierro, and Herbert Wiklicky (Eds.), *Formal Methods for Dynamical Systems*, volume 7938 of Lecture Notes in Computer Science, pp. 29 – 62. Springer Berlin Heidelberg, 2013.

2. Bogdan Iancu, Diana-Elena Gratie, Sepinoud Azimi, and Ion Petre. On the implementation of quantitative model refinement. In: Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe (Eds.), *Algorithms for Computational Biology*, volume 8542 of Lecture Notes in Computer Science, pp. 95 – 106. Springer International Publishing, 2014.

3. Diana-Elena Gratie and Ion Petre. Hiding the combinatorial state space explosion of biomodels through colored Petri nets. *Annals of University of Bucharest*, LXI:23 – 41, Editura Universității din București, 2014.

4. Diana-Elena Gratie and Ion Petre. Full structural model refinement as type refinement of colored Petri nets. In: Monika Heiner and Annegret K. Wagler (Eds.), *Proceedings of the 6th International Workshop on Biological Processes and Petri Nets*, volume 1373 of Ceur Workshop Proceedings, pp. 70 – 84. CEUR Workshop Proceedings, 2015.

5. Sepinoud Azimi, Eugen Czeizler, Cristian Gratie, Diana Gratie, Bogdan Iancu, Nebiat Ibssa, Ion Petre, Vladimir Rogojin, Tolou Shadbahr and Fatemeh Shokri. An Excursion Through Quantitative Model Refinement. In: Grzegorz Rozenberg, Arto Salomaa, José M. Sempere and Claudio Zandron (Eds.), *Membrane Computing*, volume 9504 of Lecture Notes in Computer Science, pp. 25 – 47. Springer International Publishing, 2015.

6. Diana-Elena Gratie, Bogdan Iancu, Sepinoud Azimi and Ion Petre. Quantitative Model Refinement in Four Different Frameworks. In:

Luigia Petre and Emil Sekerinski (Eds.), *From Action Sytems to Distributed Systems: The Refinement Approach*, pp. 201 – 214. Chapman and Hall/CRC, 2016.

7. Diana-Elena Gratie and Cristian Gratie. Composition colored Petri nets for the refinement of reaction-based models. *Electronic Notes in Theoretical Computer Science*, Vol. 326C, pp. 51 – 72, in press.

Part II

Original Publications

Paper I

ODE analysis of biological systems

Diana-Elena Gratie, Bogdan Iancu and Ion Petre

Originally published in Marco Bernardo, Erik de Vink, Alessandra Di Pierro, and Herbert Wiklicky (Eds.) *Formal Methods for Dynamical Systems.* Vol. 7938 of Lecture Notes in Computer Science, pp. 29–62. Springer Berlin Heidelberg, 2013. The publication is available at Springer via `http://dx.doi.org/10.1007/978-3-642-38874-3_2`.

ODE Analysis of Biological Systems

Diana-Elena Gratie, Bogdan Iancu, and Ion Petre

Computational Biomodeling Laboratory
Turku Centre for Computer Science and Åbo Akademi University
Turku, Finland
{dgratie,biancu,ipetre}@abo.fi

Abstract. This chapter aims to introduce some of the basics of modeling with ODEs in biology. We focus on computational, numerical techniques, rather than on symbolic ones. We restrict our attention to reaction-based models, where the biological interactions are mechanistically described in terms of reactions, reactants and products. We discuss how to build the ODE model associated to a reaction-based model; how to fit it to experimental data and estimate the quality of its fit; how to calculate its steady state(s), mass conservation relations, and its sensitivity coefficients. We apply some of these techniques to a model for the heat shock response in eukaryotes.

Keywords: Biomodeling, reaction-based models, ODE-based models, ODE analysis, parameter estimation, model identifiability, model refinement, heat shock response.

1 Introduction

Mathematical modeling with ordinary differential equations (ODEs) has a very long tradition in biology and ecology. Efforts to apply ODEs to understand population dynamics started already in the 18th century (see, e.g., Malthus's growth model [40]) as an effort to apply the principles of physical sciences to biological sciences as well. This research area led to major developments both in biology and ecology, as well as in mathematics. The field has long been called *biomathematics*, *mathematical biology* or *theoretical biology* and it typically involved researchers from life sciences (biology, biochemistry and ecology in particular), mathematics and more recently, from computer science and engineering. It has recently witnessed an explosion of interest in the computer science community due to the fast-paced developments in quantitative laboratory technologies. The developments on the computational side have also been influential, allowing for analyzing ever larger models and opening the door to new fields of research such as computational drug design or personalized medicine.

This chapter is primarily targeting the computer science community. Many computer scientists working in biomodeling seem to prefer a discrete stochastic approach rather than one based on ODEs. Such a choice is in some ways natural for computer scientists as it leads to new types of applications of formalisms that are well-studied in computer science, such as Petri nets, process algebra, finite automata, etc. On the other hand, such

M. Bernardo et al. (Eds.): SFM 2013, LNCS 7938, pp. 29–62, 2013.

methods come with their own limitations, especially in terms of numerical simulations of large models and in parameter estimation. Moreover, ODE-based modeling offers a huge array of analysis methods, some of which do not have a correspondent on the discrete side. Our chapter aims to introduce some of the basics of modeling with ODEs in biology, especially in terms of building such a model, analyzing some of its properties, and estimating its parameters. We focus on computational, numerical techniques, rather than on symbolic ones. We restrict our attention to reaction-based models, where the biological interactions are mechanistically described in terms of reactions, reactants and products.

The chapter is structured as follows. We discuss in Section 2 the notion of reaction-based models and introduce briefly the stochastic modeling approach in terms of continuous time Markov chains. We then discuss in more details the modeling with ODEs in Section 3. The parameter estimation problem is discussed in Section 4. We then introduce in Section 5 several analysis techniques, including steady state analysis, sensitivity analysis, and identification of mass conservation relations. As a case-study we discuss the modeling of the heat shock response in Section 6. We conclude with discussions in Section 7.

2 Reaction-Based Models

Reaction-based models are formalized as sets of reactions that describe the given system in terms of mechanistic interactions between the species of interest. We discuss separately two types of reactions: reversible and irreversible. In the following we consider a model M consisting of a set of m species $\Sigma = \{S_1, S_2, ..., S_m\}$ and n (reversible or irreversible) reactions R_j, $1 \leq j \leq n$.

Generalities. If reaction R_j, $1 \leq j \leq n$ is irreversible, then it has the following form:

$$R_j \ : \ c_{1,j}S_1 + c_{2,j}S_2 + \ldots + c_{m,j}S_m \xrightarrow{k_j} c'_{1,j}S_1 + c'_{2,j}S_2 + \ldots + c'_{m,j}S_m. \tag{1}$$

On the other hand, if it is reversible, then it is of the following form:

$$R_j \ : \ c_{1,j}S_1 + c_{2,j}S_2 + \ldots + c_{m,j}S_m \underset{k_j^-}{\overset{k_j^+}{\rightleftarrows}} c'_{1,j}S_1 + c'_{2,j}S_2 + \ldots + c'_{m,j}S_m. \tag{2}$$

In both cases, $k_j \geq 0$ ($k_j^+, k_j^- \geq 0$, resp.) is the *kinetic rate constant* of the irreversible (reversible, resp.) reaction R_j and $c_{1,j}, \ldots, c_{m,j}, c'_{1,j}, \ldots, c'_{m,j} \geq 0$ are non-negative integers that represent the quantitative proportion in which species participate in a reaction. The *stoichiometric coefficient* of molecular species S_i in reaction R_j is $n_{i,j} = c'_{i,j} - c_{i,j}$. The stoichiometric coefficients can be represented in a *stoichiometric matrix* $\mathbf{N} = (n_{i,j})_{m \times n}$. The (i, j) entry of the matrix is the stoichiometric coefficient of species S_i in reaction R_j. If $n_{i,j} > 0$ ($n_{i,j} < 0$, resp.), then we say that S_i is *produced* (*consumed*, resp.) in reaction R_j.

The reactants (the species indicated on the left-hand side of the reaction) are referred to as *substrates*, while the species produced as a result of the reaction being triggered

(indicated on the right-hand side of the reaction) are called *products*. A species S_i with $c_{i,j} = 0$ ($c'_{i,j} = 0$, resp.) is usually omitted from the left- (right-, resp.) hand side of reaction R_j.

Note that a reversible reaction can be divided into two different irreversible reactions, as follows:

$$R_j{}^{(1)} \ : \ c_{1,j}S_1 + c_{2,j}S_2 + \ldots + c_{m,j}S_m \xrightarrow{k_j{}^+} c'_{1,j}S_1 + c'_{2,j}S_2 + \ldots + c'_{m,j}S_m$$

$$R_j{}^{(2)} \ : \ c'_{1,j}S_1 + c'_{2,j}S_2 + \ldots + c'_{m,j}S_m \xrightarrow{k_j{}^-} c_{1,j}S_1 + c_{2,j}S_2 + \ldots + c_{m,j}S_m$$

The sum $\sum_{i=1}^{m} c_{i,j}$ for an irreversible reaction R_j is called the *molecularity* of reaction R_j. We consider here only reactions with molecularity at most two. Reactions with molecularity three are very rare, due to the high improbability of having three molecular entities simultaneously colliding and forming a correct configuration that leads to the constitution of a molecular complex; a molecularity greater than three for an elementary reaction is unattainable, since a number of molecules greater than three cannot concomitantly collide [49].

Example 1. We consider here the representation of a simple ecological prey-predator model through coupled chemical reactions: the Lotka-Volterra system, [39, 60]. The model consists of species *Prey* and *Predator* and its reactions are shown in Table 1.

Table 1. The Lotka-Volterra model [39, 60]

$Prey \xrightarrow{k_1} 2 \times Prey,$	growth of prey population
$Prey + Predator \xrightarrow{k_2} 2 \times Predator,$	consumption of preys
$Predator \xrightarrow{k_3} \emptyset$	death of predators

The dynamics of the Lotka-Volterra system is periodical: the population of preys grows at a rate proportional to the current population, the presence of predators in the system induces a decrease in the population of preys at a rate proportional to the number of prey-predator encounters, the population of predators declines at a rate proportional to the current population of predators. We return to this example in the next section where we associate to it an ODE-based model. A plot of its numerical simulation is shown in Figure 1.

Associating a mathematical model. After building a reaction-based model, one then associates to it a mathematical model to facilitate quantitative analysis and simulation of the model. There are many approaches available for this, see e.g. [9, 12]. The two approaches that are most used (either in a direct way, or an indirect way, as the underlying semantic of a higher-level model) are the ODE-based approach and the one based

on continuous time Markov chains (CTMCs). The modeling with CTMCs is described in more details elsewhere in this book; we only give it here a very brief presentation so that we can draw some comparison between the two in Section 7. We introduce the modeling with ODEs in more details in Section 3.

The stochastic approach is typically argued for on the basis of physical difficulties of ODE-based models with small populations [14, 15], or in terms of the network being too complex to describe in a deterministic way [61]. The stochastic formulation of a biochemical reaction network assumes homogeneity of substances and thermal equilibrium, see [16]. In this case, the model is usually described mathematically as a *continuous time Markov chain*, see [57]. Each species of the model becomes a time-dependent discrete stochastic variable indicating the number of individuals in that species, where time is modeled as a continuous variable. Formally, a stochastic process, $\{X(t), t \geq 0\}$, is a continuous-time Markov chain if for all $s,t \geq 0$, the following property is satisfied:

$$Pr\{X(s+t) = x_{s+t} | X(s) = x_s, X(u) = x_u, 0 \leq u \leq s\} = Pr\{X(s+t) = x_{s+t} | X(s) = x_s\}.$$

Intuitively, we say that the Markov chain is *memoryless*: its future dynamics depends only on the current state and not on the past states.

A continuous-time Markov chain is *time-homogeneous* if the following relation is satisfied:

$$Pr\{X(s+t) = j | X(s) = i\} = Pr\{X(t) = j | X(0) = i\}.$$

We discuss here only time-homogeneous systems.

Given a vector of non-negative integers $\mathbf{X} = (X_1, X_2, ..., X_m)$ and species $S_1, S_2, ..., S_m$ the *grand probability function* of the model, $Pr(\mathbf{X}, t)$, is the probability that there are X_1 species S_1, X_2 species S_2, ..., X_m species S_m at time t. We consider all species to be distributed randomly and homogeneously in the volume V. The central hypothesis for the stochastic formulation of chemical kinetics is that the probability of a particular combination of reactants to react according to a given reaction R in the next infinitesimal time interval $(t, t+dt)$ is $c_R dt$, for a certain constant c_R, called the *stoichiometric constant* of the reaction. The probability of a reaction occurring in the interval $(t, t+dt)$ is given by the formula $N_R \cdot c_R \cdot dt$, where by N_R we denote the number of combinations of reactants in the current state. For instance, for reaction $R^{(1)} : S_1 + S_2 \rightarrow S_3$, we have $N_{R^{(1)}} = X_1 \cdot X_2$. For reaction $R^{(2)} : 2S_1 \rightarrow S_3$, $N_{R^{(2)}} = X_1 \cdot (X_1 - 1)/2$.

For an infinitesimally small dt, the probability of the system being in a certain state at time $t+dt$ may be given by the following two scenarios: the system was in the current state at time t and no reaction occurred, or the system reached the current state as a result of a single reaction being triggered (the probability of having had two or more reactions is negligible). Denote by $a_k dt$ the probability of a reaction R_k occurring in the interval $(t, t+dt)$, given the state characterized by $\mathbf{X}$ at time t, and by $B_k dt$ the probability that reaction R_k occurs in the time interval $(t, t+dt)$ resulting in a state characterized by $\mathbf{X}$. The reasoning above can be formally written as follows:

$$Pr(\mathbf{X},t+dt) = Pr(\mathbf{X},t)(1-\sum_{k=1}^{n} a_k dt) + \sum_{k=1}^{n} B_k dt, \text{ i.e.,}$$

$$(Pr(\mathbf{X},t+dt) - Pr(\mathbf{X},t))/dt = -\sum_{k=1}^{n} a_k Pr(\mathbf{X},t) + \sum_{k=1}^{n} B_k, \text{ and so,}$$

$$\frac{\partial Pr(\mathbf{X},t)}{\partial t} = \sum_{k=1}^{n} (B_k - a_k Pr(\mathbf{X},t)). \tag{3}$$

Equation (3) is known in the literature as the *Chemical Master Equation*. A detailed mathematical analysis of a complex system using the chemical master equation has been proven to be intractable, see [61]. However, an alternative to the aforementioned approach is Gillespie's algorithm, introduced in [14, 15], that generates a random walk through the state space of the model, avoiding the solving of the master equation.

3 ODE-Based Models

We discuss in this section how to associate an ODE-based model to a reaction model. In this case, the dynamic behavior of the system is expressed in terms of the time-dependent evolution of each species' concentration. The deterministic framework of ordinary differential equations (ODEs) is often chosen as the default mathematical counterpart of a reaction-based system, sometimes followed-up by other modeling approaches. The basic quantities describing the ODE model are the concentrations $[S_1]$, $[S_2]$, ..., $[S_m]$ of the m species in the model, and the fluxes $v_1, v_2, \ldots, v_n$ of the n reactions in the model. The concentration is generally expressed either in terms of particle numbers (i.e. the number of molecules of species S, denoted $\#S$, in a solution with volume V), or in terms of moles of species S per volume V. The correspondence between the number of molecules and the number of moles is given by the relation:

$$\#S = [S] \cdot N_A,$$

where $N_A \approx 6.02214179 \cdot 10^{23}$particles/mol. The unit of $[S]$ is commonly denoted by M = mol $\cdot \mathrm{L}^{-1}$, where L is litre.

Without loss of generality, we will assume that all reactions are reversible and have the form in (2); an irreversible reaction is then a particular case, where one of the two kinetic constants is zero.

Each species S_i of the reaction model can be modeled as a function $[S_i] : \mathbb{R}_+ \to \mathbb{R}_+$ representing the time evolution of its concentration. The dependencies between the species can then be expressed in terms of a systems of ODEs in the variables $[S_i]$ modeling the change in $[S_i]$ as a function of all other variables:

$$d[S_i]/dt = \sum_{j=1}^{n} n_{i,j} v_j,$$

where v_j is the flux of reaction r_j and $n_{i,j}$ is the (i,j) stoichiometric coefficient. Here, we make the assumption that the only factor affecting the concentrations of the species are the reactions. Considering the vector of all reaction fluxes $v = (v_1, v_2, ..., v_n)^T$, the ODE representation of the entire reaction model can be written in a compact way as follows:

$$d[S]/dt = Nv, \tag{4}$$

where $[S] = ([S_1], [S_2], ..., [S_m])^T$ is a vector of the concentrations of all species in the reaction-based model, see [25].

Example 2. Consider the following reaction model:

$$2A \to B,$$
$$A + B \rightleftarrows C.$$

Denote by v_1 and v_2 the fluxes of the two reactions in the model, respectively. (We discuss in the next section how the flux of a reaction is defined, depending on the kinetic law the modeler chooses.) Then the corresponding ODE model is:

$$\begin{bmatrix} \frac{d[A]}{dt} \\ \frac{d[B]}{dt} \\ \frac{d[C]}{dt} \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

While the stoichiometries in a reaction-based model are constant, the concentrations of all species will vary in time as a function of the reaction fluxes, which are in turn dependent on the kinetics of each reaction and on the concentrations of all reactants. In the following, we describe in details two of the most common reaction kinetics: the mass-action principle, and Michaelis-Menten kinetics.

3.1 Law of Mass-Action

The most common biochemical kinetics follow the *mass action law*. It was introduced in [20,21], and it states that the *flux* (also called sometimes *rate*) of a reaction is proportional to the probability of the reactants colliding. Assuming a well-stirred environment, the probability of the substrates of a reaction colliding is proportional to their concentration to the power of their molecularity.

Example 3. For a simple reaction of the form

$$S_1 + S_2 \underset{k^-}{\overset{k^+}{\rightleftarrows}} 2P,$$

the reaction flux is

$$v = v_+ - v_- = k^+[S_1][S_2] - k^-[P]^2,$$

where v_+ represents the left-to-right (forward) reaction rate, v_- represents the right-to-left (backward) flux, k^+ is the left-to-right kinetic rate constant, and k^- represents the right-to-left kinetic rate constant. For the forward reaction, the molecularity of each of

the two substrates S_1, S_2 is 1, and for the backward reaction the molecularity is 2. If the time is measured in seconds (s), and the concentration in M, then the unit for the reaction rates is $\mathrm{M} \cdot \mathrm{s}^{-1}$. It follows that for monomolecular reactions (e.g. $S \to \emptyset$), the rate constant has unit s^{-1}, while for bimolecular reactions, the rate constant is measured in $(\mathrm{M} \cdot \mathrm{s})^{-1}$.

Considering a general reversible reaction of the form (2), the reaction rate reads

$$v = v_+ - v_- = k_j^+ \prod_{i=1}^{m} [S_i]^{c_{i,j}} - k_j^- \prod_{i=1}^{m} [S_i]^{c'_{i,j}}.$$

The corresponding system of ODEs, following (4), is

$$\frac{d[S_i]}{dt} = n_{i,j} v = (c'_{i,j} - c_{i,j}) \left(k_j^+ \prod_{l=1}^{m} [S_l]^{c_{l,j}} - k_j^- \prod_{l=1}^{m} [S_l]^{c'_{l,j}} \right), 1 \leq l \leq m.$$

For reversible reactions, the ratio of substrate and product at steady state (i.e., when the forward and backward reaction rates are equal, $v_+ = v_-$) is a constant, K_{eq}, called the equilibrium constant:

$$K_{eq} = \frac{k_j^+}{k_j^-} = \frac{\prod_{i=1}^{m} [S_i]_{eq}^{c'_{i,j}}}{\prod_{i=1}^{m} [S_i]_{eq}^{c_{i,j}}},$$

where $[S_i]_{eq}$ represents the equilibrium concentration of species S_i.

The time course for a species S is obtained by integrating the corresponding ODE. For a simple decomposition reaction $S \xrightarrow{k} P_1 + P_2$, the time dynamics is described by the ODE $d[S]/dt = -k[S]$. Integrating over the interval [0, t) yields the analytical solution

$$\int_{S_0}^{S} d[S]/dt = -\int_{t=0}^{t} k dt \Rightarrow [S](t) = S_0 e^{-kt}.$$

Calculating the analytical solution for more complex models is however rarely possible.

Example 4. For the Lotka-Volterra model introduced in Example 1, the mass-action reaction fluxes for the three reactions in the system are the following:

$$v_1 = k_1[Prey], \quad v_2 = k_2[Prey][Predator], \quad v_3 = k_3[Predator].$$

The system of ODEs describing the dynamics of the Lotka-Volterra model is:

$$\begin{aligned} d[Prey]/dt &= v_1 - v_2 = k_1[Prey] - k_2[Prey][Predator] \\ d[Predator]/dt &= v_2 - v_3 = k_2[Prey][Predator] - k_3[Predator]. \end{aligned} \tag{5}$$

The periodic dynamics of the Lotka-Volterra model is depicted in Figure 1.

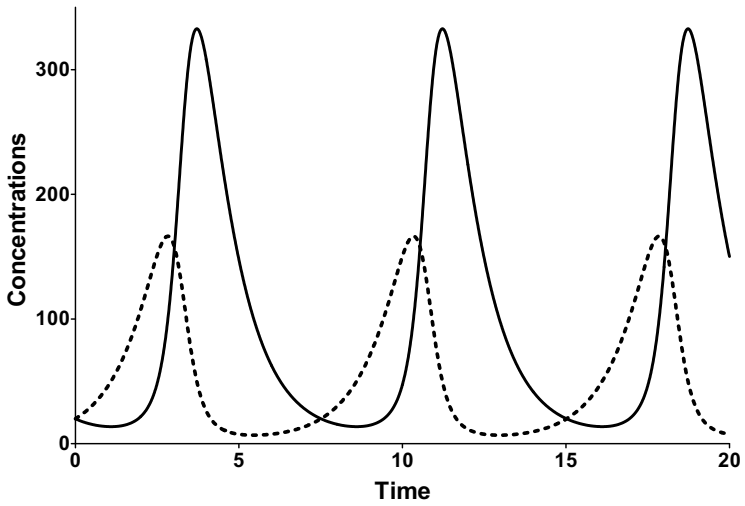


Fig. 1. The periodic time dynamics of the Lotka-Volterra model. The solid line represents the concentration of *Predator*, while the dotted line represents that of *Prey*. As the *Prey* population grows, the *Predator* population also grows; then there are more encounters *Predator-Prey* that reduce the *Prey* population; this reflects on the *Predator*, as they only multiply as long as they find food. When the size of *Predator* drops, the *Prey* population starts to grow, and the cycle repeats.

3.2 Kinetics of Enzymatic Reactions

Enzymatic reactions are a special class of biochemical reactions, where an enzyme is required for a reaction to take place, but the enzyme itself is not consumed during the reaction. The general form of an enzymatic reaction, as proposed in [7] based on previous experimental results of [28, 43], is:

$$S + E \underset{k_2}{\overset{k_1}{\rightleftarrows}} S:E \overset{k_3}{\to} P + E,$$

where E is an enzyme, S is the substrate of the reaction, $S:E$ is a substrate-enzyme complex, and P is the product. This system of reactions represents in fact the reaction $S \to P$, catalyzed by enzyme E. The system can be represented using mass-action kinetics, considering the following irreversible reactions:

$$S + E \overset{k_1}{\to} S:E, \quad S:E \overset{k_2}{\to} S + E, \quad S:E \overset{k_3}{\to} P + E. \tag{6}$$

The system of ODEs describing the mass-action dynamics of the reaction-based model (6) is the following:

$$\frac{d[S]}{dt} = -k_1[S][E] + k_2[S:E]; \tag{7}$$

$$\frac{d[E]}{dt} = -k_1[S][E] + k_2[S:E] + k_3[S:E]; \tag{8}$$

$$\frac{d[S:E]}{dt} = k_1[S][E] - k_2[S:E] - k_3[S:E]; \tag{9}$$

$$\frac{d[P]}{dt} = k_3[S:E]. \tag{10}$$

Michaelis-Menten kinetics. Because the system of ODEs (7) - (10) cannot be solved analytically, simplifying assumptions have been proposed. For example, the kinetic constants k_1, k_2 could be assumed to be much greater than k_3 ($k_1, k_2 \gg k_3$, see [43]), i.e. $[S:E]$ is negligible compared to $[S]$ and $[P]$, because the substrate-enzyme complex concentration is very low. This is called the *quasi-equilibrium* between the free enzyme E and the compound $S:E$.

This assumption has been further extended (see [7]) to considering that the system will eventually reach a state where the concentration of substrate-enzyme complex remains unchanged (*quasi-steady state* of $S:E$); the assumption only holds when $S_0 \gg E_0$. In this case we obtain:

$$d[S:E]/dt = 0, \text{ i.e., } k_1[S][E] - k_2[S:E] - k_3[S:E] = 0. \tag{11}$$

Note that the right hand side of (8) is the complement of the right hand side of (9). Adding them we get that $d[E]/dt + d[S:E]/dt = 0$. Equivalently,

$$[E] + [S:E] = E_{tot}, \text{or equivalently } [E] = E_{tot} - [S:E], \tag{12}$$

where E_{tot} is constant, standing for the total amount of enzyme in the system, either free or as part of the substrate-enzyme complex.

Considering the *quasi-steady state* assumption and (12), equation (11) can be rewritten as follows:

$$\begin{aligned} &k_1[S]E_{tot} = k_1[S][S:E] + k_2[S:E] + k_3[S:E], \text{ i.e.,} \\ &[S:E] = \frac{k_1[S]E_{tot}}{k_1[S] + k_2 + k_3}, \text{ i.e.,} \\ &[S:E] = \frac{[S]E_{tot}}{[S] + \frac{k_2+k_3}{k_1}} \end{aligned} \tag{13}$$

Introducing (13) into (10) yields the result

$$\frac{d[P]}{dt} = \frac{k_3[S]E_{tot}}{[S] + \frac{k_2+k_3}{k_1}}. \tag{14}$$

The Michaelis-Menten equation relates the reaction rate v of synthesizing the product P to the concentration of the substrate, $[S]$, by the relation:

$$v = \frac{d[P]}{dt} = \frac{V_{max}[S]}{[S] + K_m}, \tag{15}$$

where V_{max} represents the maximum rate achieved by the system, for saturated values of $[S]$. The Michaelis constant K_m is the concentration of substrate for which the reaction rate is half-maximal. Identifying the parameters of (15) into (14) yields the connection between the Michaelis-Menten kinetics and the mass-action deduced kinetics of an enzymatic reaction:

$$V_{max} = k_3 E_{tot}, \quad K_m = \frac{k_2 + k_3}{k_1}.$$

Assuming the *quasi-equilibrium*, the quantity k_3/k_1 is negligible, thus $K_m \cong k_2/k_1$. Figure 2 shows the dependency of the reaction rate v with $[S]$. For more details on Michaelis-Menten kinetics, we refer the reader to [37].

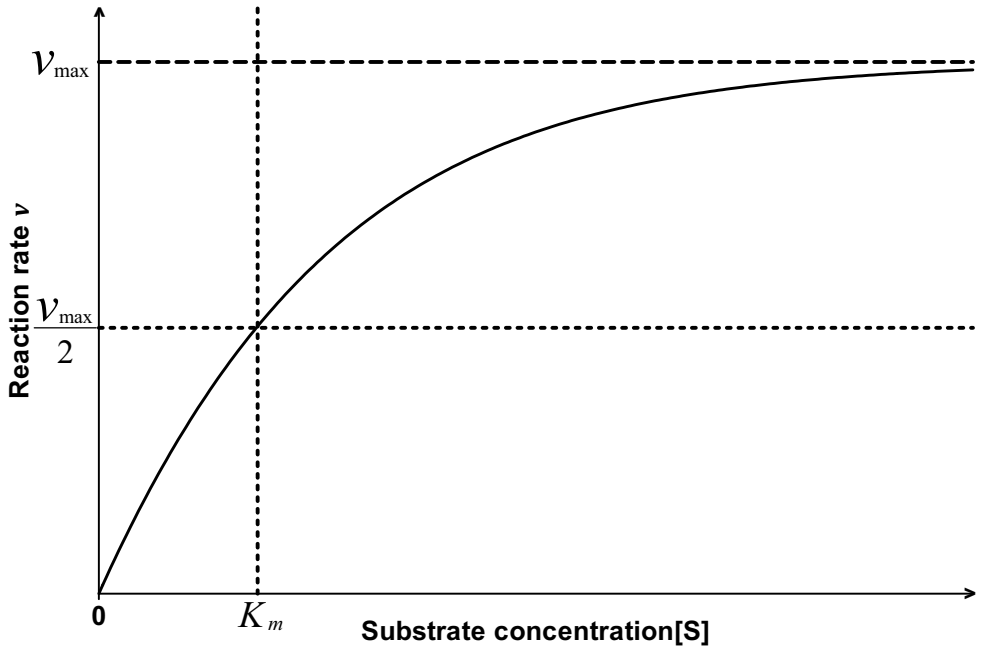


Fig. 2. Dependency of the reaction rate v with $[S]$ for Michaelis-Menten kinetics. v_{max} represents the maximum velocity, and K_m is the concentration of substrate for which the reaction rate is half-maximal.

Reversible Michaelis-Menten kinetics. Enzyme kinetics can often be reversible, and the Michaelis-Menten equation can be extended to a reversible reaction of the form

$$S + E \underset{k_2}{\overset{k_1}{\rightleftarrows}} X \underset{k_4}{\overset{k_3}{\rightleftarrows}} P + E, \tag{16}$$

where S and P are substrates, E is the enzyme, and X represents the intermediary enzyme-substrate compound. The mass-action irreversible reactions describing this system are:

$$\begin{aligned} S+E &\xrightarrow{k_1} X;\\ X &\xrightarrow{k_2} S+E;\\ X &\xrightarrow{k_3} P+E;\\ P+E &\xrightarrow{k_4} X. \end{aligned} \tag{17}$$

The system of ODEs describing the dynamics of the reaction-based model (17) is:

$$\frac{d[S]}{dt} = -k_1[S][E] + k_2[X]; \tag{18}$$

$$\frac{d[E]}{dt} = -k_1[S][E] + k_2[X] + k_3[X] - k_4[P][E]; \tag{19}$$

$$\frac{d[X]}{dt} = k_1[S][E] - k_2[X] - k_3[X] + k_4[P][E]; \tag{20}$$

$$\frac{d[P]}{dt} = k_3[X] - k_4[P][E]. \tag{21}$$

Following the reasoning for simple Michaelis-Menten equations, adding (19) and (20) yields

$$\frac{d[E]}{dt} + \frac{d[X]}{dt} = 0 \Rightarrow [E] + [X] = E_{tot}.$$

For the *quasi-steady state*, $d[X]/dt = 0$, i.e., $k_1[S](E_{tot} - [X]) - [X](k_2 + k_3) + k_4[P](E_{tot} - [X]) = 0$, which leads to

$$[X] = \frac{k_1[S]E_{tot} + k_4[P]E_{tot}}{k_1[S] + k_4[P] + k_2 + k_3}. \tag{22}$$

Introducing (22) into equation (21), after a few computations the formula reads

$$v = \frac{k_1k_3[S]E_{tot} - k_2k_4[P]E_{tot}}{k_1[S] + k_4[P] + k_2 + k_3} = \frac{k_3E_{tot}\frac{k_1[S]}{k_2+k_3} - k_2E_{tot}\frac{k_4[P]}{k_2+k_3}}{1 + \frac{k_1[S]}{k_2+k_3} + \frac{k_4[P]}{k_2+k_3}} = \frac{\frac{V_{fw}}{K_{mS}}[S] - \frac{V_{bw}}{K_{mP}}[P]}{1 + \frac{[S]}{K_{mS}} + \frac{[P]}{K_{mP}}},$$

where $K_{mS} = (k_2 + k_3)/k_1$ and $K_{mP} = (k_2 + k_3)/k_4$ are the Michaelis-Menten constants (i.e. for half-maximal forward and backward rate) for the substrate and product, respectively, and $V_{fw}(V_{bw})$ denotes the maximal rate in forward (backward) direction. An exact solution to this equation can be found in [44]. For details on the reversible Michaelis-Menten kinetics, we refer the reader to [22].

Other kinetic laws. Mass action and Michaelis-Menten are not the only existing kinetics. Some enzymatic reaction can follow Hill kinetics, Goldbeter-Koshland kinetics, or be subject to inhibition. We only introduce them briefly, discussing the types of reactions that are typically modeled in this way, and skipping the derivation of their mathematical formulations.

Goldbeter-Koshland kinetics, introduced in [18], applies to reversible reactions from substrate to product and back, catalyzed by different enzymes (e.g. phosphorylation and

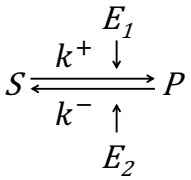


Fig. 3. Goldbeter-Koshland kinetics. P is produced from S in presence of enzyme E_1 and S is produced from P in presence of enzyme E_2.

dephosphorylation of proteins). The forward and backward reactions have Michaelis-Menten kinetics. The general form of such reactions is shown in Figure 3.

Hill kinetics, introduced in [29], are suitable for reactions where the enzyme can bind more molecules from the substrate S. Usually, the binding of the first S molecule changes the binding rate of the second molecule. The rate can either increase (called *positive cooperativity*), or decrease (called *negative cooperativity*). A general form of such reactions is the following:

$$E + S_1 \underset{k_1^-}{\overset{k_1^+}{\rightleftarrows}} ES_1;$$

$$ES_1 \overset{k_2}{\to} P + E;$$

$$ES_1 + S_2 \underset{k_3^-}{\overset{k_3^+}{\rightleftarrows}} ES_1S_2;$$

$$ES_1S_2 \overset{k_4}{\to} P + E.$$

Inhibition in a system with Michaelis-Menten kinetics (see (16)) can occur at different levels. An inhibitor I can bind to an enzyme in different states of the enzyme. When it binds (in a reversible reaction) to the free enzyme, the inhibition is called *competitive*, as both the substrate and the inhibitor are competing for binding the enzyme. When I binds reversibly to the enzyme-substrate complex, the reaction is called *uncompetitive inhibition*, as the enzyme is already bound to the substrate. When the inhibitor binds both to the free enzyme and to the enzyme-substrate complex, the inhibition is called *noncompetitive*. For a more detailed description of enzyme inhibition reactions, we refer the reader to [37].

4 Parameter Estimation

We discuss in this section the parameter estimation problem, including aspects of model identifiability, quantitative measures for model fit quality, model validation, and methods for model fitting.

4.1 Generalities: Relating the Mathematical Model to the Experimental Data

Relating the mathematical model to the experimental data is an essential step in the process of model building. This includes the validation of the model in terms of how

well it can explain existing (quantitative or qualitative) experimental data and how well its predictions correspond to existing (non-quantitative) knowledge. There are several ways to approach this problem.

1. The modeler might have no a-priori hypothesis regarding the mathematical form and is instead strongly guided by data. The focus here is to capture the trend of the data and to predict the behavior in-between the data points and the emphasis is on the data. This approach is called *interpolation*.
2. The modeler has a clear hypothesis regarding the mathematical form she is building. For example, she might start from a reaction-based model and then associate to it an ODE-based model with mass-action kinetics as discussed in Section 3. The focus here is on finding values for all model parameters and the emphasis is on the model. This approach is called *model fitting*.
3. The modeler might replace a fitted model with an interpolating curve because of a need for better mathematical properties in further manipulations/analsysis of the model. This approach is called sometimes *model approximation*.

We only focus in this section on aspects related to model fitting. For a basic introduction to other approaches we refer to [17].

The main focus in model fitting is on the estimation of the unknown kinetic parameters of the model so that its predictions are consistent with a given set of data, usually presented in terms of time series. This step is often followed by a model validation step, where the model is compared with another set of data, that was not used in the fitting stage. In both cases, the task can be formulated as a mathematical optimization problem to minimize a cost function that quantifies the differences between the model predictions and the experimental measurements. The cost function can be seen as a distance measure between two vectors with non-negative real numbers as entries, one holding the experimental data, the other the model prediction for the time points where the data was collected. Some of the most widely used cost measures in this context are based on the *Chebyshev criterion*, *sums of absolute deviations*, and *least-squares*. We introduce briefly each of them in the following. In all cases, we are given a data set (x_i, y_i), $1 \leq i \leq m$ and a model $y = f(k,x)$, where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ and $k \in \mathbb{R}^n$ is the vector of parameters, often with non-negative values.

In the Chebyshev criterion, the goal is to find $k \in \mathbb{R}^n$ that minimizes $\max\{|y_i - f(k,x_i)|, 1 \leq i \leq m\}$. In other words, the goal is to *minimize the largest absolute deviation* of a model value from the corresponding experimental value. The effect is that more weight is given to the worst outlier.

Another approach is to find $k \in \mathbb{R}^n$ that minimizes $\sum_{1 \leq i \leq m} |y_i - f(k,x_i)|$. In other words, the goal is to *minimize the sum of absolute deviations*. The effect is to treat each data point equally and to average the deviations over all experimental points.

In the third approach we mention here, the goal is to find $k \in \mathbb{R}^n$ that minimizes $\sum_{1 \leq i \leq m} |y_i - f(k,x_i)|^2$. This is the most widely used criterion in model fitting because the resulting optimization problem can be approached using calculus if f is a differentiable function (such as those obtained through the methods in Section 3).

The problem of estimating the parameters of kinetic models in systems biology is computationally difficult, see e.g., [4,42,45]. Regardless of which fitting criterion (score

function) is used, the high number of variables in a typical biomodel makes an exact solution to the problem unfeasible in practice. There are however many approximation methods. Some of them are based on local approximation algorithms; they are faster in practice, but tend to converge to local optima. Others are based on global optimization algorithms; they are in general slower, but tend to converge to a global optimum. The global optimization methods can be based on deterministic searches [19, 33] or on stochastic ones [2, 6]. Even though the deterministic methods guaranty the convergence to a global optimum, the speed of the convergence is typically a major concern and in general, these methods cannot ensure the termination of the algorithm within a given finite time interval [45]. On the other hand, the intrinsic randomness of the stochastic approaches does not guarantee their convergence to an optimum [45]. However, many stochastic methods exhibit a good performance in practice – they are often capable of efficiently identifying a point in the vicinity of global solutions, see [45].

There are many modeling software environments, some commercial, others offering free access, that are used for model fitting. In most of our projects we chose COPASI [31] as a computational environment for parameter estimation. This software is a widely used tool in computational systems biology, having a documented good performance, see [4, 42, 45]. It includes a suite of various local and global, deterministic and stochastic parameter estimation algorithms, such as simulated annealing, genetic algorithms, evolution strategy using stochastic ranking, and particle swarm.

4.2 Alternative Model Fits and Model Identifiability

The problem of *model identifiability* adds to the difficulty of model fitting; it has to do with a model having several (sometimes very) different sets of parameter values, all yielding good model fits. The problem is that some numerical properties of the model, such as sensitivity coefficients, might be drastically different in different numerical setups, even if they all fit well the available data. This implies that there exist several models (or model setups) offering equally good, but different explanations for the available data. In such a situation, additional data is needed, focusing on the domains where the candidate models exhibit different behavior.

Even when only one model fit has been achieved, the modeler should evaluate the uniqueness of the parameter set. One way of doing this is to repeat the parameter estimation procedure, using some other available algorithms but the same data set. Such a procedure can in principle yield several different results, as demonstrated e.g. in [8, 53].

When searching for alternative numerical model fits, one can sample the distribution of the score functions measuring the distance between the model predictions through a simultaneous sampling on the range of all parameter values. For each parameter, one can generate a large sample, e.g. through partitioning its value range into a large number of equal sized subintervals (say, on the scale of tens of thousands) and randomly select a value from each of them. For all combinations of values for all parameters, one can then calculate the score of the model fit and thus sample the distribution of the score function. However, the direct implementation of this idea is clearly intractable for models with more than a few parameters due to the combinatorial explosion of the number of model simulations that need to be run. A fast, practical solution to this problem is the *Latin Hypercube Sampling* method (LHS) of [41]. This is a method to generate samples

which are uniformly distributed over each parameter space, with the number of samples being independent of the number of parameters, see [26,27,50] for several applications. Let p be the number of parameters. The first step is to choose the size of the sample, N; this will also serve as the number of samples for each parameter. The range of each parameter is then partitioned into N intervals, with the length of each interval proportional to the probability of the parameter's value to fall in that interval; in particular, if the parameter is uniformly distributed in its range, then all subintervals are equal-sized. We then randomly select a value from each subinterval to generate a sample of N values for each parameter. The N values for parameter i are then stored on the i-th column of an $N \times p$ matrix. Finally, we randomly shuffle the values on all the columns of the matrix. The result is read from the matrix row-by-row, giving a sample of N combinations of parameter values. For a detailed description of this sampling scheme we refer to [41]. We discuss this method in the case of the heat shock response model in Section 6.

4.3 Fit-Preserving Model Refinement

Altering an already-fitted model, for example by adding a new component to it, replacing a module with another one, or adding new variables and reactions to it, will lead to losing its numerical fit. The problem is especially difficult in cases where the number of parameters in the new model is much larger than in the starting model. Rather then attempting to re-estimate all parameters, including those that were already fitted in the starting model, a computationally more efficient way is to build the larger model in an iterative way, ensuring in each step that its quantitative model fit is preserved. This method is called *quantitative model refinement* and has already been investigated in several different setups in [3, 34, 46]. We follow here the presentation of [34].

A given reaction-based model can undergo several types of refinement, for instance depending whether the focus lies on the reactants or reactions of the model. If one's focus lies on model's data, then the model could be refined so as to include more details regarding a species by having it substituted for several of its subspecies. The main interest in this type of refinement originates in the analysis of the possible behavioral intricacies the model refined as such would depict. This type of refinement is called *data refinement* and it consists in refining a set of variables so as to include more details about their internal states, attributes, etc. If the interest lies on the reactions of the model, one could refine the model by replacing for instance a reaction in the model describing a certain process by a set of reactions detailing on some intermediate steps of the process. This type of refinement is called *process refinement*.

Formal refinement arose from the field of software engineering as a necessity to embed an elementary set of specifications in a system's final implementation. The problem of quantitative model refinement has been addressed before in systems biology in particular related to rule-based modeling, which integrates *data refinement* through the notion of agent resolution ([23]). The focus lies here on rule refinement, a method designed to refine rules ensuring model fit preservation. Nevertheless, the refinement technique must preserve the quantitative systemic properties of the model, such as numerical fit and validation, see [46].

A model M consisting of a set of reactions of the form (1) could be formalized through a discrete or continuous approach, a deterministic or non-deterministic

evolution, etc. This discussion focuses on a continuous mass-action formulation. To each variable $S_i, 1 \leq i \leq m$ we associate a time-dependent function $[S_i] : \mathbb{R}_+ \to \mathbb{R}_+$, which denotes the concentration of the species over time. According to the principle of mass action, see [37], the time evolution of the system may be specified by a system of ODEs as follows:

$$\frac{d[S_i]}{dt} = -\sum_{j=1}^{n}\left(k_j c_{i,j}\prod_{l=1}^{m}[S_l]^{c_{l,j}}\right) + \sum_{j=1}^{n}\left(k_j c'_{i,j}\prod_{l=1}^{m}[S_l]^{c'_{l,j}}\right), \quad 1 \leq i \leq m. \tag{23}$$

Assume now model M is refined discerning among various subspecies of S_1. The distinction among the subspecies of S_1 can be made by either different classes of S_1 or several biochemical configurations of S_1, as a result of various post-translational modifications such as acetylation, phosphorylation, etc. All subspecies characterized as such participate in all reactions S_1 took part in (in model M), with possible variations in the kinetics. Replacing species S_1 in model M by subspecies $V_1, \ldots, V_l$ brings about a new model M_R, whose set of species consists of the new variables $\{S'_2, S'_3, ..., S'_m\} \cup \{V_1, ..., V_l\}$, for some $l \geq 2$, where variables S'_j, $2 \leq j \leq m$ of M_R, match variables S_j of model M and $V_1, ..., V_l$ replace species S_1 in M_R. Moreover, each reaction R_j of model M is substituted for in model M_R by a reaction R'_j as follows:

$$\begin{array}{c} R'_j \ : \ (T_{1,j}V_1 + ... + T_{l,j}V_l) + c_{2,j}S'_2 + ... + c_{m,j}S'_m \xrightarrow{k'_j} \\ (T'_{1,j}V_1 + ... + T'_{l,j}V_l) + c'_{2,j}S'_2 + ... + c'_{m,j}S'_m, \end{array}$$

where k'_j is the kinetic rate constant, and $T_{1,j}, ..., T_{l,j}, T'_{1,j}, ..., T'_{l,j}$ are nonnegative integers so that $T_{1,j} + ... + T_{l,j} = c_{1,j}$ and $T'_{1,j} + ... + T'_{l,j} = c'_{1,j}$.

Model M_R is a *data refinement of model M on variable* S_1 if and only if the subsequent conditions hold, see [34]:

$$[S_j](t) = [S'_j](t), \text{ for all } 2 \leq j \leq m, \tag{24}$$

$$[S_1](t) = [V_1](t) + \ldots + [V_l](t), \text{ for all } t \geq 0. \tag{25}$$

The refined model, M_R, involves a number of $m + l - 1$ species, while model M comprises only m species, M_R evolving linearly in the size of its data set. The number of reactions in M_R substituting for reaction R_j of M is the number of non-negative integer solutions of the subsequent system of equations:

$$\begin{aligned} T_{1,j} + T_{2,j} + ... + T_{l,j} &= c_{1,j}; \\ T'_{1,j} + T'_{2,j} + ... + T'_{l,j} &= c'_{1,j}; \end{aligned}$$

over the independent unknowns $T_{k,j}, T'_{k,j}, 1 \leq k \leq l$. The number of solutions of the first equation is given by the *multinomial coefficient* "l multichooses $c_{1,j}$", see [13]:

$$\left(\!\!\binom{l}{c_{1,j}}\!\!\right) = \binom{l + c_{1,j} - 1}{c_{1,j}} = \frac{(l + c_{1,j} - 1)!}{c_{1,j}!\,(l-1)!}.$$

Some values for the new kinetic parameters of M_R may be attained from the literature or they can be estimated experimentally. The parameters not attained as such require

calculation through computational methods so that conditions (24) and (25) are fulfilled. The reiteration of the parameter estimation process is however computationally expensive. As an alternative, the method proposed in [34] describes an approach for setting the values of the unknown parameters in the refined model so that relations (24) and (25) hold. The approach promotes a choice of parameters symmetrical in $V_1, ..., V_l$.

4.4 Quantitative Measures for the Model Fit Quality

Given parameter estimation may yield several different outputs, depending on the methods that were used in the fitting, it is important to quantify the goodness of a model fit. In this way, the results of different parameter estimation rounds can be compared. Moreover, through a suitable normalization, even the fitting of different models, using different sets of data, may also be compared. Part of the challenge here is to avoid to discriminate against models deviations that may be large in absolute values, but relatively small compared to the experimental data.

We discuss here briefly a notion of model fit quality introduced in [38]. Their fit quality only takes into account one set of experimental data at a time and aims to give a measure of the average deviation of the model from the data, normalized on the scale of the numerical values of the model predictions. For a given experimental data set $\mathcal{E} = \{(x_i, y_i) \mid 1 \leq i \leq n\}$ and a model $M = f(k,x)$, the quality of M's fit with respect to $\mathcal{E}$ is denoted as $q(M, \mathcal{E})$ and is defined as follows:

$$q(M, \mathcal{E}) = \frac{\sqrt{\sum_{i=1}^{n} (f(k,x_i) - y_i)^2 / n}}{\sum_{i=1}^{n} f(k,x_i)/n} \cdot 100\%.$$

It was argued in [38] that a low (say, lower than $15-20\%$) value of $q(M, \mathcal{E})$ could be considered as an indicator of a successful fit. We discuss the quality of the best fit for the heat shock response model in Section 6 and refer to [10] for more details on applying this measure.

5 Analysis of ODE-Based Models

We discuss in this section several computational analysis techniques for ODE-based models. We apply some of these techniques in the next section, on the heat shock response model.

5.1 Steady State Analysis

Steady states (also called *stationary states*, *fixed points*, *equilibrium points*) have the property that when taken as initial values for the model, they yield a constant dynamics; in other words, there is no change in the concentration of any of the species when starting from steady state values. This is one of the basic concepts in dynamical systems theory, extensively employed in modeling biological systems. There are several types of steady states: *stable*, *asymptotically stable*, *unstable* etc.

Consider a dynamical system $dx/dt = f(x(t))$, $x(0) = x_0$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous function with equilibrium point x_e. The equilibrium is *stable* if for every $\varepsilon > 0$ there exists δ_ε such that, if $||x_0 - x_e|| < \delta_\varepsilon$, then $||x(t) - x_e|| < \varepsilon, \forall t \geq 0$. A steady state is called *asymptotically stable* if there exists $\delta > 0$ such that if $||x_0 - x_e|| < \delta$, then $\lim_{t\to\infty} ||x(t) - x_e|| = 0$. A steady state is *unstable* if the conditions for stability are not met.

For a reaction-based model, the steady state behavior is characterized by the equation

$$\frac{d[S]}{dt} = 0,$$

or equivalently, considering Equation (4),

$$Nv = 0. \tag{26}$$

The rate vector v that satisfies the steady state condition (26) can be obtained by solving the corresponding system of algebraic equations with the variables $[S_1], [S_2], \ldots, [S_m]$. The equation has nontrivial solutions (not all variables are zero) only if $\text{rank}(N) < n$, where n is the number of reactions in the system, i.e. matrix N contains at least one pair of linearly dependent columns. The dependencies can be expressed by a so-called kernel matrix K, such that

$$NK = 0, \tag{27}$$

where K has $c = n - \text{rank}(N)$ columns. The columns k_i of matrix K are the vectors that span the null space (also termed kernel) of N, i.e. the subspace of the reaction rates space that contains all solutions to Equation (26), see [24]. Consequently, any vector J of steady-state fluxes can be expressed as a linear combination of K's columns,

$$J = \sum_{i=1}^{c} \alpha_i k_i.$$

The kernel matrix K is not uniquely determined. Another kernel matrix K' could be obtained for example by a multiplication $K' = KQ$, where Q has dimensions $[n - \text{rank}(N)] \times [n - \text{rank}(N)]$. Since K is a solution to Equation (27), so is K'. For details on how to determine the kernel matrix using Gauss's algorithm, we refer the reader to [37].

Example 5. Consider the following system of reactions:

$$\begin{aligned} 2A &\xrightarrow{k_1} B; \\ A + B &\underset{k_2^-}{\overset{k_2^+}{\rightleftarrows}} C. \end{aligned}$$

To compute the steady state, one needs to solve Equation (26), which reads as the following system of algebraic equations:

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}} = \underbrace{\begin{bmatrix} -2 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{\mathbf{N}} \cdot \underbrace{\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}}_{\mathbf{v}}.$$

Considering mass action kinetics and denoting by $[A]_0, [B]_0, [C]_0$ the steady-state concentrations for the species in the model, the system reads:

$$\begin{aligned} &-2k_1[A]_0^2 - k_2^+[A]_0[B]_0 + k_2^-[C]_0 = 0,\\ &k_1[A]_0^2 - k_2^+[A]_0[B]_0 + k_2^-[C]_0 = 0,\\ &k_2^+[A]_0[B]_0 - k_2^-[C]_0 = 0. \end{aligned}$$

Solving the steady-state system of equations gives the solution $[A]_0 = 0, [B]_0 = \alpha, [C]_0 = 0$, where $\alpha > 0$ is arbitrary.

Example 6. Let us consider the Lotka-Volterra model expressed in Table 1. The ODEs characterising the system's dynamics are expressed in Equation (5). The steady state analysis leads to the system

$$\begin{aligned} &k_1[Prey] - k_2[Prey][Predator] = 0,\\ &k_2[Prey][Predator] - k_3[Predator] = 0. \end{aligned}$$

Solving this system of two equations gives the steady state points

$$([Prey]_s, [Predator]_s) \in \{(0,0), (k_3/k_2, k_1/k_2)\}.$$

To study the behavior of the Lotka-Volterra model around the steady states, one needs to examine the behavior of the concentrations around each equilibrium point, i.e. their tendency to increase or decrease. To do that, one studies the sign of the derivatives:

$$\begin{aligned} &\tfrac{d[Prey]}{dt} \geq 0 \Rightarrow k_1 - k_2[Predator] \geq 0 \Rightarrow [Predator] \leq \tfrac{k_1}{k_2};\\ &\tfrac{d[Predator]}{dt} \geq 0 \Rightarrow k_2[Prey] - k_3 \geq 0 \Rightarrow [Prey] \geq \tfrac{k_3}{k_2}. \end{aligned} \tag{28}$$

The behavior around the steady states is depicted in Figure 4.

5.2 Mass Conservation Relations

In this section we introduce mass conservation relations and their importance in modeling reaction-based systems. For a more detailed presentation and additional examples we refer to [24].

Identifying the mass conservation relations in a given model is one of the first analyzes that a modeler typically performs. It gives an insight into the dynamics of the model, but at the same time it reduces the number of free variables in the model. Mathematically, a mass conservation relation is a linear combination of concentrations of species that is constant in time:

$$g^T S = C, \tag{29}$$

where g is a vector with some constant entries, S is the species concentrations vector, and C is some constant. An implication of mass conservation relations is that some of the stoichiometric matrix rows are linearly dependent, i.e.

$$g^T N = 0^T. \tag{30}$$

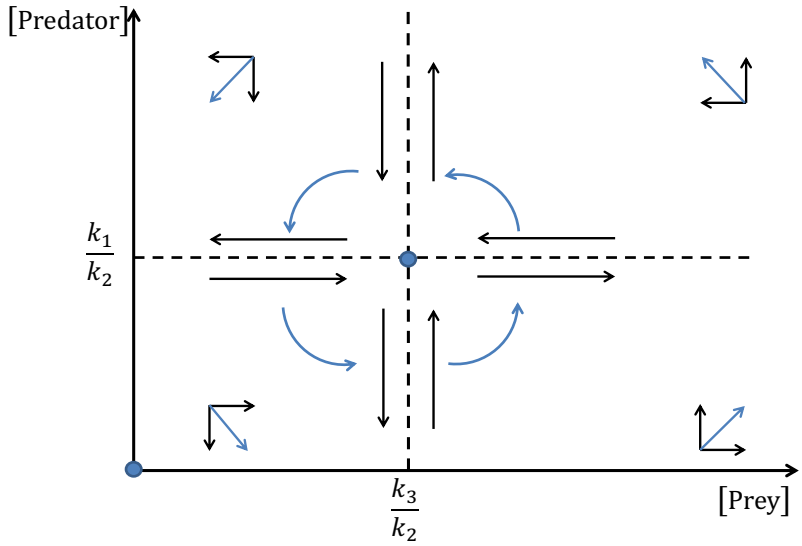


Fig. 4. Steady-state analysis of the Lotka-Volterra system. The blue dots are the two steady states of the system. The black arrows indicate how the concentration of each species increases or decreases (as established in (28)). The blue arrows are the combination of *Predator* and *Prey* concentrations tendencies, and they show how the dynamics of the system changes between the four areas delimited by the dotted lines. The behavior around the $(k_3/k_2, k_1/k_2)$ point suggests periodicity; this is confirmed by Figure 1. Both equilibrium points are unstable, as indicated by the blue arrows.

Equations (29) and (30) are equivalent. Derivating the former equation and taking into account Equation (4) yields

$$(g^T S)' = g^T \dot{S} = g^T N v = 0.$$

There may be more linearly independent vectors g that satisfy Equation (30), each denoting a different mass conservation relation. The number of mass conservation relations is given by $m-$ rank(N), where m is the number of species in the system. The full set of vectors g describing these mass conservation relations form a so-called *conservation matrix* G , see [24], with the property

$$GN = 0.$$

Consequently, G^T is a kernel matrix for N^T. A conservation matrix G can be determined using the Gauss algorithm, and it is not unique (any other matrix $G' = PG$, where P is any nonsingular matrix of appropriate dimensions, is a valid conservation matrix).

Example 7. Consider the following system of biochemical reactions:

$$\begin{aligned}
&2A \rightleftarrows A_2;\\
&A_2 + B \rightleftarrows A_2 : B;\\
&A_2 : B \rightarrow C + A_2 : B;\\
&C \rightarrow \emptyset.
\end{aligned}$$

The species vector, stoichiometric coefficients matrix and the conservation matrix read:

$$S = \begin{bmatrix} A \\ A_2 \\ B \\ A_2 : B \\ C \end{bmatrix}, \quad N = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

The mass conservation relations induced by G are:

$$\begin{aligned} &[A] + 2[A_2] + 2[A_2 : B] = C_1, \\ &[B] + [A_2 : B] = C_2, \end{aligned} \tag{31}$$

for some constants C_1, C_2.

The mass conservation relations are important for reducing the system of differential equations $\dot{S} = Nv$ that describe the dynamics of the model. Each mass conservation relation introduces one dependent variable, which can be expressed in terms of the independent variables, and thus eliminated from the system of ODEs. The two mass conservation relations in Equation (31) could be used to express the dependency between $[A], [B]$ and the rest of the species concentrations:

$$\begin{aligned} [A] &= C_1 - 2[A_2] - 2[A_2 : B], \\ [B] &= C_2 - [A_2 : B]. \end{aligned}$$

This reduces the initial system of ODEs from 5 to 3 equations.

5.3 Sensitivity Analysis

Sensitivity analysis is a method of estimating the changes that small perturbations in the parameters of a model induce in the system. With this type of analysis, one can estimate the robustness of a model against small changes, and also identify ways of inducing a desired change into the model. There exist many methods for sensitivity analysis, some suitable for spatially homogeneous constant-parameter reaction-based models, others suitable for systems with space- and time-dependent parameters, or stochastic models. For a review of multiple methods, we refer the reader to [59, 62]. One of the questions often encountered in biochemical systems is what changes should the system undergo such that the new steady state satisfies certain properties.

There are two types of sensitivity analysis: *local sensitivity analysis*, and *global sensitivity analysis*. In the global approach, all parameters are varied at once, and the sensitivity is measured over the entire range of each parameter. In the local analysis, only one parameter is varied at a time, within a small interval around some nominal value. Generally, it is assumed that input-output relationships are linear. We only focus here on local sensitivity analysis.

We consider the system of ODEs describing a system to be expressed as a function of the concentrations of all species and all the parameter values:

$$\frac{d[S_i]}{dt} = f_i([S_1], [S_2], ..., [S_m], \kappa), \tag{32}$$

where $\kappa = (k_1, k_2, ..., k_n)^T$ is the rate constants vector (assuming without loss of generality that the system comprises n irreversible reactions). Let $\mathcal{S}(t,\kappa) = ([S_1](t,\kappa), [S_2](t,\kappa), ..., [S_m](t,\kappa))^T$ be the solution of Equation (32) with respect to κ, also called *sensitivity matrix*. The elements of the matrix are the partial derivatives $\partial[S_i]/\partial k_j$, also called *first-order local sensitivity coefficients*.

There are many ways of determining the local sensitivity of the concentrations. The simplest method is the *brute force method* (also called *indirect method*, or *finite-difference method*), that uses the finite difference approximation. The j-th parameter, k_j, changes with the amount δk_j at time point t_1, and all other parameters remain unchanged. One can compute the new matrix $[S]$ using the change between the initial and the perturbed solution, see Equation (33). The method requires $n+1$ runs, one for the initial values of the parameters and n modifying each of the parameters at a time.

$$\frac{\partial[S](t_2)}{\partial k_j(t_1)} = \frac{[S](t2, k_j + \delta k_j) - [S](t_2, k_j)}{\delta k_j}, 1 \leq j \leq n. \tag{33}$$

This method is widely used because of its simplicity, but other more efficient methods exist, e.g. the *direct method*. This method solves the differential equations for the sensitivity coefficients $\partial[S_i]/\partial k_j$, by differentiating Equation (32). This results in the following set of sensitivity equations:

$$\frac{d}{dt}\frac{\partial[S]}{\partial k_j} = \mathcal{J}\frac{\partial[S]}{\delta k_j} + \frac{\partial f}{\partial k_j}, 1 \leq j \leq n,$$

where $\mathcal{J}$ is the Jacobian for Equation (32). For a complete mathematical derivation of this result, see [62].

Perturbations should be small enough to yield small errors in the indirect method, and large enough to surpass the simulation inaccuracies of ODE solvers, for the direct method, see [62]. Other methods of computing the sensitivity of a model to parameter changes exist, e.g. the Green function method, polynomial approximation method, AIM method, detailed in [54, 59].

Very often, sensitivity analysis is focused on the steady states, when concentrations are constant. In this case, the sensitivity coefficients are computed as solutions to the system

$$\frac{d}{dt}\frac{\partial[S]}{\partial k_j} = 0,$$

and reflect the dependency of the steady state on the parameters. If the steady state is asymptotically stable, then one can consider the limit $\lim_{t\to\infty}(\partial[S]/\partial k_j)(t), 1 \leq j \leq n$, called *stationary sensitivity coefficients*. The system can be written as

$$\frac{\partial[S]}{\partial k_j} = -\mathcal{J}F_j, 1 \leq j \leq n,$$

where $\mathcal{J}$ is the value of the jacobian at steady state, and F_j is the j-th column in the matrix $F = (\partial f_r/\partial k_s)_{m\times n}$ computed at steady state. Sensitivity coefficients can be computed in many software applications, e.g. in COPASI [31].

6 The Heat Shock Response Model

We consider in this section a larger modeling case-study to which we apply some of the techniques discussed in the chapter. The eukaryotic heat shock response is an evolutionarily conserved bio-regulatory network, crucial to cell survival. It acts as a defence mechanism that regulates the cellular response to proteotoxicity induced by diverse physiological and environmental stressors, such as elevated temperatures. Exposure of proteins to elevated temperatures causes protein misfolding, which results in the constitution of large aggregates that eventually induce apoptosis (controlled cell death). Protein homeostasis is promoted by augmenting the level of molecular chaperons.

6.1 The Reaction-Based Model

We consider here the basic molecular model for the heat shock response introduced in [53]. Elevated temperatures cause protein misfolding and accumulation of misfolded proteins in large conglomerates that induce cell death. The key role in homeostasis restoration is played by *heat shock proteins* (hsp), which chaperone the misfolded proteins, promoting the folding of proteins. The transactivation of hsp-encoding genes regulates the heat shock response. Heat shock factors (hsf) activate gene transcription. In the absence of stress, heat shock factors are present in a monomeric conformation and they are bound to a great extent to heat shock proteins. However, heat stress actuates the dimerization (hsf_2) and consequently trimerization (hsf_3) of heat shock factors, a DNA binding-competent conformation. Due to their high affinity toward the heat shock element (hse), hsf trimers bind to the heat shock elements, promoting the transcription and translation of the gene. Consequently, DNA binding activates hsp synthesis, see [53, 55].

Once the heat stress is removed, hsp synthesis is turned off as follows: hsp's sequestrate free hsf's (residing in the constitution of hsp:hsf complexes), break hsf_2 and hsf_3 and induce DNA unbinding, see [53, 55]. Subsequently, DNA transcription is turned off and the formation of new hsf trimers repressed. The heat shock response mechanism is switched back on when the temperature is again elevated, impelling the proteins in the cell (prot) to misfold and hsp:hsf complexes to break down. The reactions of the molecular model in [53] are shown in Table 2.

Table 2. The molecular model for the eukaryotic heat shock response proposed in [53]

$2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$	$\mathsf{hsp} + \mathsf{hsf}_3 \to \mathsf{hsp{:}hsf} + 2\,\mathsf{hsf}$
$\mathsf{hsf} + \mathsf{hsf}_2 \rightleftarrows \mathsf{hsf}_3$	$\mathsf{hsp} + \mathsf{hsf}_3{:}\mathsf{hse} \to \mathsf{hsp{:}hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$
$\mathsf{hsf}_3 + \mathsf{hse} \rightleftarrows \mathsf{hsf}_3{:}\mathsf{hse}$	$\mathsf{hsp} \to \emptyset$
$\mathsf{hsf}_3{:}\mathsf{hse} \to \mathsf{hsf}_3{:}\mathsf{hse} + \mathsf{hsp}$	$\mathsf{prot} \to \mathsf{mfp}$
$\mathsf{hsp} + \mathsf{hsf} \rightleftarrows \mathsf{hsp{:}hsf}$	$\mathsf{hsp} + \mathsf{mfp} \rightleftarrows \mathsf{hsp{:}mfp}$
$\mathsf{hsp} + \mathsf{hsf}_2 \to \mathsf{hsp{:}hsf} + \mathsf{hsf}$	$\mathsf{hsp{:}mfp} \to \mathsf{hsp} + \mathsf{prot}$

This molecular model is clearly on a high level of abstraction, for the sake of easing its analysis. For example, note that the eukaryotic cell presents various classes of heat shock proteins, denominated according to their molecular weight, e.g., Hsp60, Hsp70, Hsp90. However, in this molecular model, they are all referred to as belonging to the same class, with Hsp70 as common denominator. The same assumptions are made for hsf and hse. Furthermore, the model considers all proteins uniformly, distinguishing only between the ones that are correctly folded (prot) and the misfolded ones (mfp). The model contains also simplified representations of some cellular mechanisms, e.g., protein synthesis and degradation, see [53] for more details.

The molecular model in [53] satisfies the following three mass-conservation relations, for the total amount of hsf, the total amount of proteins (excluding hsp and hsf) and for the total amount of hse:

- $[\mathsf{hsf}] + 2[\mathsf{hsf_2}] + 3[\mathsf{hsf_3}] + 3[\mathsf{hsf_3}\colon\mathsf{hse}] + [\mathsf{hsp}\colon\mathsf{hsf}] = C_1$,
- $[\mathsf{prot}] + [\mathsf{mfp}] + [\mathsf{hsp}\colon\mathsf{mfp}] = C_2$,
- $[\mathsf{hse}] + [\mathsf{hsf_3}\colon\mathsf{hse}] = C_3$,

where C_1, C_2 and C_3 are constants.

6.2 The Mathematical Model

Given the molecular model in Table 2, we consider a mathematical model derived through the principle of mass action, formulated as a system of ordinary differential equations ([37]). The rate coefficient for protein misfolding ($\mathsf{prot} \rightarrow \mathsf{mfp}$) is described by the following formula:

$$\varphi(T) = (1 - \frac{0.4}{e^{T-37}}) \cdot 1.4^{T-37} \cdot 1.45 \cdot 10^{-5} s^{-1},$$

where T is the temperature of the environment, expressed in $^\circ C$, in accordance to [52]. Each species X in the molecular model is associated to a continuous, time-dependent function $[X](t)$, expressing the concentration of the respective reactant. The dynamics of the system is described through the system of differential equations in Table 3.

The initial values of all species and the kinetic rate constants were estimated in [53], by imposing the following three conditions:

(i) At $37^\circ C$ the system is in a steady state, since the model should not reveal any response in the absence of the heat stress;
(ii) At $42^\circ C$, the numerical predictions for DNA binding ($[\mathsf{hsf_3}\colon\mathsf{hse}](t)$) should be in accordance with the experimental data reported in [36];
(iii) At $42^\circ C$, the numerical prediction of the model for $[\mathsf{hsp}](t)$ should confirm the data obtained in [53] through a de-novo fluorescent reporter-based experiment.

The numerical setup obtained in [53] for the heat shock response model is shown in Table 4.

The estimation of parameters was based on the experimental data in [36] on DNA binding in HeLa cells for a temperature of $42^\circ C$. Moreover, the model should also be in a steady state at $37^\circ C$. Hence, seven more independent algebraic relations on the set of

Table 3. The system of ODE's associated with the biochemical model proposed in [53]

$$
\begin{aligned}
d[\mathsf{hsf}]/dt &= -2k_1^+[\mathsf{hsf}]^2 + 2k_1^-[\mathsf{hsf}_2] - k_2^+[\mathsf{hsf}][\mathsf{hsf}_2] + k_2^-[\mathsf{hsf}_3] \\
&\quad - k_5^+[\mathsf{hsf}][\mathsf{hsp}] + k_5^-[\mathsf{hsp{:}\,hsf}] + k_6[\mathsf{hsf}_2][\mathsf{hsp}] \\
&\quad + 2k_7[\mathsf{hsf}_3][\mathsf{hsp}] + 2k_8[\mathsf{hsf}_3{:}\,\mathsf{hse}][\mathsf{hsp}]; \\
d[\mathsf{hsf}_2]/dt &= k_1^+[\mathsf{hsf}]^2 - k_1^-[\mathsf{hsf}_2] - k_2^+[\mathsf{hsf}][\mathsf{hsf}_2] + k_2^-[\mathsf{hsf}_3] \\
&\quad - k_6[\mathsf{hsf}_2][\mathsf{hsp}]; \\
d[\mathsf{hsf}_3]/dt &= k_2^+[\mathsf{hsf}][\mathsf{hsf}_2] - k_2^-[\mathsf{hsf}_3] - k_3^+[\mathsf{hsf}_3][\mathsf{hse}] + k_3^-[\mathsf{hsf}_3{:}\,\mathsf{hse}] \\
&\quad - k_7[\mathsf{hsf}_3][\mathsf{hsp}]; \\
d[\mathsf{hse}]/dt &= -k_3^+[\mathsf{hsf}_3][\mathsf{hse}] + k_3^-[\mathsf{hsf}_3{:}\,\mathsf{hse}] + k_8[\mathsf{hsf}_3{:}\,\mathsf{hse}][\mathsf{hsp}]; \\
d[\mathsf{hsf}_3{:}\,\mathsf{hse}]/dt &= k_3^+[\mathsf{hsf}_3][\mathsf{hse}] - k_3^-[\mathsf{hsf}_3{:}\,\mathsf{hse}] - k_8[\mathsf{hsf}_3{:}\,\mathsf{hse}][\mathsf{hsp}]; \\
d[\mathsf{hsp}]/dt &= k_4[\mathsf{hsf}_3{:}\,\mathsf{hse}] - k_5^+[\mathsf{hsf}][\mathsf{hsp}] + k_5^-[\mathsf{hsp{:}\,hsf}] - k_6[\mathsf{hsf}_2][\mathsf{hsp}] \\
&\quad - k_7[\mathsf{hsf}_3][\mathsf{hsp}] - k_8[\mathsf{hsf}_3{:}\,\mathsf{hse}][\mathsf{hsp}] - k_{11}^+[\mathsf{hsp}][\mathsf{mfp}] \\
&\quad + (k_{11}^- + k_{12})[\mathsf{hsp{:}\,mfp}] - k_9[\mathsf{hsp}]; \\
d[\mathsf{hsp{:}\,hsf}]/dt &= k_5^+[\mathsf{hsf}][\mathsf{hsp}] - k_5^-[\mathsf{hsp{:}\,hsf}] + k_6[\mathsf{hsf}_2][\mathsf{hsp}] \\
&\quad + k_7[\mathsf{hsf}_3][\mathsf{hsp}] + k_8[\mathsf{hsf}_3{:}\,\mathsf{hse}][\mathsf{hsp}]; \\
d[\mathsf{mfp}]/dt &= \varphi(T)[\mathsf{prot}] - k_{11}^+[\mathsf{hsp}][\mathsf{mfp}] + k_{11}^-[\mathsf{hsp{:}\,mfp}]; \\
d[\mathsf{hsp{:}\,mfp}]/dt &= k_{11}^+[\mathsf{hsp}][\mathsf{mfp}] - (k_{11}^- + k_{12})[\mathsf{hsp{:}\,mfp}]; \\
d[\mathsf{prot}]/dt &= -\varphi(T)[\mathsf{prot}] + k_{12}[\mathsf{hsp{:}\,mfp}].
\end{aligned}
$$

parameters and initial values are derived. Therefore, the model comprises 17 independent values that require estimation. The above-mentioned conditions are satisfied by the values in Table 4. These values have been attained by means of parameter estimation in COPASI [31]. The model is fit with regard to the DNA binding experimental data in [36]. The model predictions regarding $\mathsf{hsf}_3{:}\,\mathsf{hse}$ compared with the experimental data of [36] are shown in Figure 5.

6.3 Model Validation

The model exhibits a very low rate for protein misfolding for a temperature of $37^{\circ}C$ and a high rate for protein folding, in compliance with [5] and [35]. The model also predicts a transient increase in the level of hsf trimers, in accordance with [30]. The model confirms that dimers are only a transient form between monomers and trimers, and that the level of dimers is low throughout the simulation, regardless of the temperature.

Another validation test consisted in applying the heat shock response twice subsequently. The second heat shock was applied after the heat shock proteins had attained a maximal level. The model in [53] predicted the response to the second heat shock to be

Table 4. The numerical values of the parameters (A) and the initial values of the variables (B) of the heat shock response model proposed in [53]

A

Param.	Value	Units
k_1^+	3.49	$\frac{ml}{\#\cdot s}$
k_1^-	0.19	s^{-1}
k_2^+	1.07	$\frac{ml}{\#\cdot s}$
k_2^-	10^{-9}	s^{-1}
k_3^+	0.17	$\frac{ml}{\#\cdot s}$
k_3^-	$1.21 \cdot 10^{-6}$	s^{-1}
k_4	$8.3 \cdot 10^{-3}$	s^{-1}
k_5^+	9.74	$\frac{ml}{\#\cdot s}$
k_5^-	3.56	s^{-1}
k_6	2.33	$\frac{ml}{\#\cdot s}$
k_7	$4.31 \cdot 10^{-5}$	$\frac{ml}{\#\cdot s}$
k_8	$2.73 \cdot 10^{-7}$	$\frac{ml}{\#\cdot s}$
k_9	$3.2 \cdot 10^{-5}$	s^{-1}
k_{11}^+	$3.32 \cdot 10^{-3}$	$\frac{ml}{\#\cdot s}$
k_{11}^-	4.44	s^{-1}
k_{12}	13.94	s^{-1}

B

Variable	Initial conc.
[hsf]	0.67
[hsf_2]	$8.7 \cdot 10^{-4}$
[hsf_3]	$1.2 \cdot 10^{-4}$
[hse]	29.73
[hsf_3:hse]	2.96
[hsp]	766.88
[hsp:hsf]	1403.13
[mfp]	517.352
[hsp:mfp]	71.65
[prot]	1.15×10^8

greatly diminished in intensity. Indeed, a diminished response for the second heat shock could be anticipated since the level of heat shock proteins (hsp's) is already elevated as a consequence of the first heat shock. A similar result was reported in [52].

Another validation method consisted in simulating the model for a temperature of $43°C$ and comparing the results with those of [55]. The model in [53] predicts a prolonged transactivation for DNA binding, as opposed to the model in [55], but it is consistent with the experimental data in [1]. An experiment consisting in the removal of the heat shock at $42°C$ at the peak of the response exhibited an accelerated attenuation phase, complying with the results reported by [55].

An alternative verification scenario focused on the prediction of the evolution of heat shock proteins (hsp's) over time. This method required the use of a quantitative reporter system founded on *yellow fluorescent proteins* (yfp's). This method was based on the assumption that fluorescence intensity is virtually linear reported to the level of yfp's. As yfp's transactivation is regulated by their own heat shock elements, denoted in [53] by hse′, transcription and degradation kinetics ($k_4{}'$ and $k_9{}'$ respectively), their evolution in time may be described by the following differential equation:

$$d[\mathsf{yfp}]/dt = k_4'[\mathsf{hsf_3{:}hse'}] - k_9'[\mathsf{yfp}], \tag{34}$$

for some positive constants k_4', k_9' accounting for the kinetic rate constants of yfp synthesis and of yfp degradation. The extended model, including equation (34), takes into

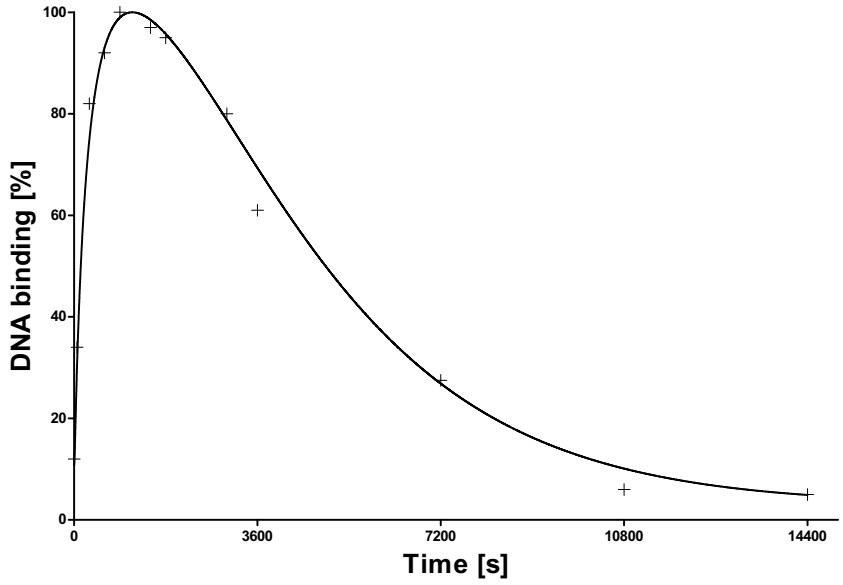


Fig. 5. The dynamic behavior of $\mathsf{hsf_3{:}hse}$ in the best fitted model. The continuous line is the model prediction and the crossed points indicate the experimental data of [36].

account all numerical values from the basic model, described in Table 2, and the numerical values for the new rate constants k_4' and k_9' were estimated so that the fit for yfp's complies with the experimental data.

6.4 Model Analysis

Sensitivity analysis. The first analysis approach consisted in estimating the scaled steady state sensitivity coefficients, see [59], of all variables against reaction rate constants and initial concentrations. Given a variable X and a parameter p, the *scaled steady state sensitivity coefficient* of variable X against parameter p is defined by:

$$\lim_{t\to\infty} \frac{\partial ln(X)}{\partial \ln(p)(t)}.$$

The coefficients described above represent the relative variance of the steady state when the model undergoes infinitesimal changes in parameter p. The sensitivity coefficients of all variables against reaction rate constants $k_1{}^-, k_2{}^-, k_3{}^-, k_7$ proved to be all insignificant, suggesting that the reactions corresponding to those rate constants may not be crucial to the global behavior of the model. For this aim, the model was altered so as to exclude the reactions corresponding to the aforementioned kinetic rate constants, namely the backward reactions for dimerization, trimerization, DNA binding and DNA unbinding. The new model attained as such satisfies the validation tests described in Section 6.3. This suggests that hsf dimers and trimers are steady configurations and that non-hsp-mediated DNA unbinding is negligible. While the breaking of trimers

(by hsp) does not affect greatly the overall behavior of the model, the reaction describing the breaking of trimers (by hsp) proved to have a substantial impact on the evolution of hsp and mfp.

The variation between the steady state levels of hsp and mfp are correlated, see [53], which is consistent with the biological knowledge that hsp's have a major role in chaperoning mfp's. Table 5 shows the largest sensitivity coefficients for hsp and mfp. The coefficients with respect to ${k_5}^+$ and ${k_5}^-$ are the highest, suggesting that the reaction describing hsf sequestration (forward)/dissipation of hsp:hsf ($\mathsf{hsp}+\mathsf{hsf} \leftrightarrows \mathsf{hsp{:}hsf}$) is the main feedback loop. The forward direction, hsf sequestration, hereafter compels the ceasing of transcription, inducing an augmentation in the level of mfp and a decrease in that of hsp. The backward direction (dissipation of hsp:hsf), however, actuates an increase in the level of hsp and hsf and a reduction of mfp. Considering the coefficients in Table 5 in descending order, the next set of coefficients to discuss consists of ${k_1}^+$, ${k_2}^+$ and k_4, corresponding to the forward directions of dimer(trimer) formation and DNA binding respectively, suggesting the augmentation of the transcription level and thereupon the level of hsp. On the contrary, the reactions describing the breaking of dimers, hsp degradation and protein misfolding, diminish the transcription level. The reactions influencing the level of mfp alone are the reactions corresponding to the sequestration of mfp's/dissipation of hsp:mfp (see coefficients corresponding to ${k_{11}}^+$ and ${k_{11}}^-$ in Table 5) and protein refolding (same for k_{12}).

Among the sensitivity coefficients of hsp and mfp with respect to the initial concentrations, the one dependent on the initial level of hsp:hsf ($\mathsf{hsp{:}hsf}(0)$) was the most relevant. On the other hand, the sensitivity coefficients of hsp and mfp with respect to the level of any of the hsf species (monomers, dimers or trimers) were insignificant. This is to be expected since initially the majority of hsf's is sequestrated by hsp's and the initial levels of dimers and trimers are reduced, which is consistent with [30]. Consequently, the sensitivity coefficient with respect to $\mathsf{hsp{:}hsf}(0)$ should be conceived as describing a dependency over the total initial amount of hsf.

The sensitivity coefficients with respect to the initial amount of hse were insignificant, which is justified by the consideration of the sensitivity coefficients around the steady state. For instance, for a lower initial amount of hse, the response reaches hereafter the same steady state. A higher level of $\mathsf{hsf}(0)$ brings no change in the evolution of the response. The sensitivity coefficients of hsp and mfp with respect to $\mathsf{hsp}(0)$ were also insignificant.

Model identifiability. Looking into the model identifiability problem, alternative good numerical fits were searched for, using the same fitting data as in the model fitting procedure described above. Several were found, but none of them passed the additional validation tests described in the previous section. Then the *Latin Hypercube Sampling* method was applied to sample the distribution of the fitting score function. The first step was to generate a sample of $N = 100000$ combinations of parameter values, as described in Section 4. For each of them, the initial values were chosen so that they are a steady state of the model at 37°C. Out of these, the analysis was continued only for those combinations that were "responsive", where a model was declared responsive if $\mathsf{hsf_3{:}hse}(900) \geq 20$ (note that the experimental data indicated that the peek of the

Table 5. The largest scaled steady state sensitivity coefficients of hsp and mfp. The coefficients are identical for both $37°C$ and $42°C$ [53]

Parameter description	p	$\frac{\partial ln(\mathsf{hsp})}{\partial ln(p)}\|_{t\to\infty}$	$\frac{\partial ln(\mathsf{mfp})}{\partial ln(p)}\|_{t\to\infty}$
Sequestration of hsf	${k_5}^+$	-0.50	0.50
Dissipation of hsp:hsf	${k_5}^-$	0.50	-0.50
Formation of dimers	${k_1}^+$	0.17	-0.17
Formation of trimers	${k_2}^+$	0.17	-0.17
Transcription, translation	k_4	0.17	-0.17
Affinity of hsp for hsf_2	k_6	-0.17	0.17
Affinity of hsp for hsf_3:hse	k_8	-0.17	0.17
Degradation of hsp	k_9	-0.17	0.17
Affinity of hsp for mfp	${k_{11}}^+$	0.00	-1.00
Dissipation of hsp:mfp	${k_{11}}^-$	0.00	0.24
Protein refolding	k_{12}	0.00	-0.24
Initial level of hsp:hsf	$\mathsf{hsp}:\mathsf{hsf}(0)$	0.50	-0.50

response is reached after 900 time units). The result was interesting: there were only 31506 models satisfying the constraint, already suggesting that finding suitable alternative model fits is a difficult problem. For each of these models we calculated the fit quality as discussed in Section 4; the result is plotted in Figure 6, showing clearly our best fit as an outlier in the fit quality distribution. More details on the identifiability of the heat shock response model can be found in [53]. This suggests that fitting the simple heat shock response model in Table 2 to the experimental data in [36] and to the steady-state condition for the initial values is indeed a difficult numerical problem.

7 Discussion

The focus of our chapter has been on the practical use of modeling with ordinary differential equations in biology. Our choice of topics to discuss has been driven by targeting primarily the computer science community and by the space limitations. This chapter should only be seen as a "teaser" for modeling with ODEs in biology; for a more comprehensive reading on this topic, many excellent textbooks exist, such as [11, 32, 47, 48, 56, 58]. We only considered in this chapter reaction-based models and started by discussing how to associate to them an ODE-based model; we presented briefly several laws for biochemical kinetics: mass-action, Michaelis-Menten, Goldbeter-Koshland, Hill, and inhibition. One should note that many other types of models exist, see, e.g., [9]. We then discussed the parameter estimation problem, including model identifiability, measures for fit quality, and fit-preserving model refinement. We then introduced several analysis methods for ODE-based models: steady state analysis, mass conservation, and sensitivity analysis. In addition to some smaller examples discussed throughout the chapter, we dedicated a separate section to a larger case-study on the eukaryotic heat shock response.

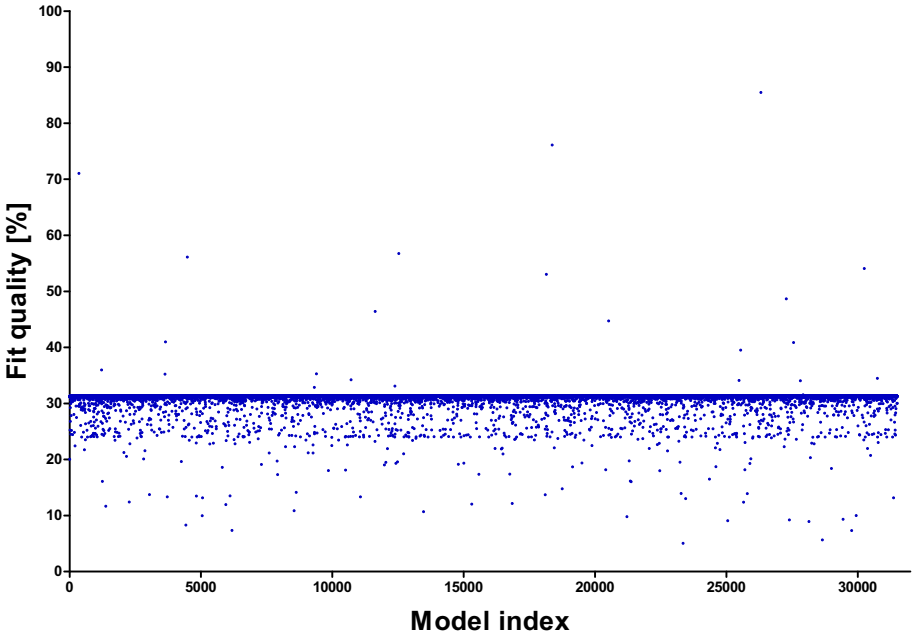


Fig. 6. The distribution of the model fit quality among 31506 model variants obtained through the Latin hypercube sampling method. Most models exhibit a constant level of $\mathsf{hsf}_3\mathsf{:hse}$, very different from the dynamic behavior in Figure 5; these models yield a numerical value for the fit quality around 30%. The quality of our best fit is around 10^{-30}.

There are many computational benefits that modeling with ODEs brings, including fast numerical simulations, many methods for parameter estimation, several highly useful static and dynamic analysis techniques, such as mass conservation, steady state analysis, flux-balance analysis, metabolic control analysis, sensitivity analysis, etc. At the same time, the ODE-based approach also suffers from several difficulties. The one that is most discussed is the inability to account for stochastic noise in a system, which might be problematic especially in cases where there are relatively small species; a detailed discussion about the physical limitations of the ODE-based approach is in [14, 15]. Another difficulty is in the need for knowing a potentially large number of kinetic parameters; measuring them experimentally is sometimes impossible, while estimating them computationally suffers from model identifiability issues. A partial solution here is the approach based on quantitative model refinement, see [34]. Another partial solution is in terms of static, rather than dynamic analysis, often performed around the steady states; such an approach is modeling based on flux balance analysis, see [51].

The stochastic approach, either in terms of continuous time Markov chains (CTMC) and the chemical master equation, or in terms of higher-level formalisms (such as Petri nets or process algebra) based on a CTMC semantic, is often offered as a solution to the physical limitations of the ODE-based approach. It is important however to understand the limitations of both approaches so that we can take advantage of the benefits of either one, whenever they are applicable. In Table 6 we summarized several aspects about modeling with ODEs and with CTMCs, and placed them in mirror for an easy comparison. It is also important to point out that in the case of very large models, both approaches are insufficient, see Figure 7.

Table 6. Some of the differences between the deterministic and the stochastic modeling approaches

	Deterministic approach	*Stochastic approach*
Fundamental assumptions	the system is well-stirred and at thermodynamical equilibrium	the system is well-stirred and at thermodynamical equilibrium
Modeling goal	it models the average behavior of the system	it models individual runs of the system
Concept	based on the concept of diffusion-like reactions	based on the concept of reactive molecular collisions
Type of model	the time evolution of the model is a continuous process	the time evolution of the model is a random-walk process through the possible states
Math model	governed by a set of ODEs	governed by a single ODE: the chemical master equation
Analytic solution	the system of ODEs is often impossible to solve analytically	the chemical master equation is often impossible to solve
Small populations	conceptual difficulties when small populations are involved	no difficulties with small populations
Numerical simulations	fast	Gillespies algorithm is slow; many runs are needed

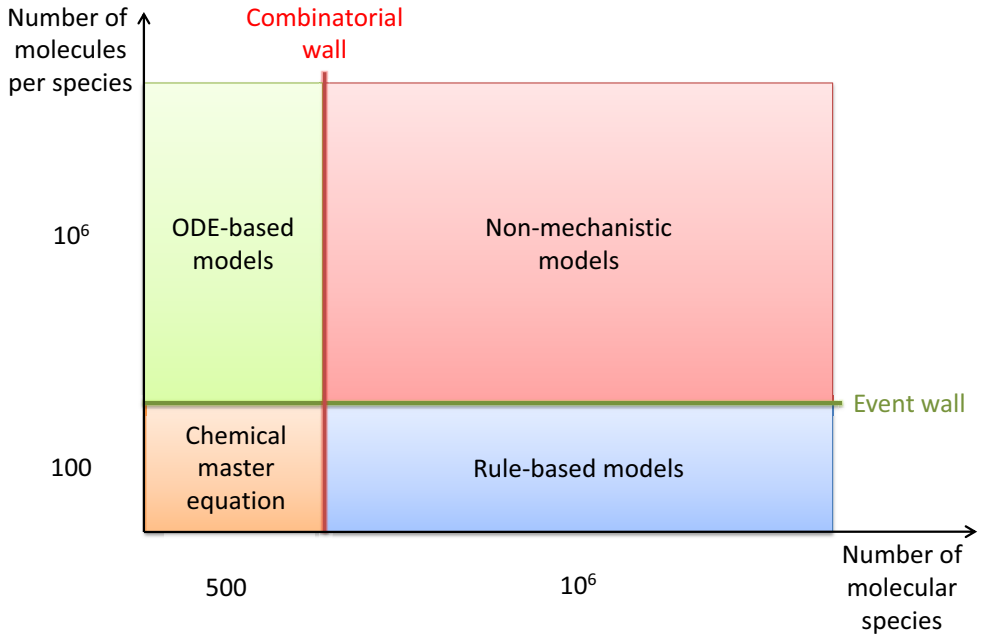


Fig. 7. Modeling limitations depending on the size of the model. Adapted from Walter Fontana `http://fontana.med.harvard.edu/`

The ODE-based approach to computational modeling is (still) arguably the standard choice for biomodelers, especially on the biological side of the community. There are many advantages that it brings, as there are clear limitations. Even in cases where another modeling approach is taken, the corresponding ODE-based model is often also built to serve as comparison to related (ODE-based) models and to make available tools such as parameter estimation or steady state analysis. Moreover, on top of the ODE-based semantic there are many other discrete techniques that can be added to give further insight into the model: Petri net tools, control analysis, network motif identification, etc. In the continuing debate of 'discrete vs. continuous biomodeling' we argue that it is good to retain the advantages of both worlds and use them to their full potential whenever applicable.

References

1. Abravaya, K., Phillips, B., Morimoto, R.I.: Attenuation of the heat shock response in hela cells is mediated by the release of bound heat shock transcription factor and is modulated by changes in growth and in heat shock temperatures. Genes & Development 5(11), 2117–2127 (1991)
2. Ali, M.M., Storey, C., Törn, A.: Application of stochastic global optimization algorithms to practical problems. Journal of Optimization Theory and Applications 95(3), 545–563 (1997)
3. Azimi, S., Gratie, D.-E., Iancu, B., Petre, I.: Three approaches to quantitative model refinement with applications to the heat shock response. Technical report, Turku Centre for Computer Science (2013)
4. Baker, S.M., Schallau, K., Junker, B.H.: Comparison of different algorithms for simultaneous estimation of multiple parameters in kinetic metabolic models. Journal of Integrative Bioinformatics 7(3), 1–9 (2010)
5. Ballew, R.M., Sabelko, J., Gruebele, M.: Direct observation of fast protein folding: the initial collapse of apomyoglobin. Proceedings of the National Academy of Sciences 93(12), 5759–5764 (1996)
6. Boender, G., Romeijn, E.: Stochastic methods. In: Handbook of Global Optimization: nonconvex optimization and its applications, pp. 829–869. Kluwer Academic Publishers (1995)
7. Briggs, G.E., Haldane, J.B.S.: A note on the kinetics of enzyme action. Biochemical Journal 19(2), 338–339 (1925)
8. Chen, W.W., Schoeberl, B., Jasper, P.J., Niepel, M., Nielsen, U.B., Lauffenburger, D.A., Sorger, P.K.: Input-output behavior of erbb signaling pathways as revealed by a mass action model trained against dynamic data. Molecular Systems Biology 5, 239 (2009)
9. Ciobanu, G., Rozenberg, G. (eds.): Modeling in Molecular Biology. Springer (2004)
10. Czeizler, E., Rogojin, V., Petre, I.: The phosphorylation of the heat shock factor as a modulator for the heat shock response. IEEE-ACM Trans. Comp. Biol. Bioinf. 9(5), 1326–1337 (2012)
11. Edelstein-Keshet, L.: Mathematical Models in Biology. McGraw-Hill, New York (1988)
12. Fisher, J., Henzinger, T.A.: Executable cell biology. Nature Biotechnology 25, 1239–1249 (2007)
13. Gessel, I.M., Stanley, R.P.: Algebraic enumeration. Handbook of Combinatorics 2, 1021–1061 (1995)
14. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics 22(4), 403–434 (1976)
15. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry 81(25), 2340–2361 (1977)

16. Gillespie, D.T.: A rigorous derivation of the chemical master equation. Physica A: Statistical Mechanics and its Applications 188(1), 404–425 (1992)
17. Giordano, F., Weir, M., Fox, W.: A first course in mathematical modeling, 3rd edn. Thomson (2003)
18. Goldbeter, A., Koshland, D.E.: An amplified sensitivity arising from covalent modification in biological systems. Proceedings of the National Academy of Sciences 78(11), 6840–6844 (1981)
19. Grossmann, I.E.: Global optimization in engineering design. Kluwer Academic Publishers (1996)
20. Guldberg, C.M., Waage, P.: Studies concerning affinity. CM Forhandlinger: Videnskabs-Selskabet i Christiana 35, 92–111 (1864)
21. Guldberg, C.M., Waage, P.: Etudes sur les affinités chimiques. Brøgger & Christie (1867)
22. Haldane, J.B.S.: Enzymes (1930, 1965)
23. Harmer, R.: Rule-based modelling and tunable resolution. EPTCS 9, 65–72 (2009)
24. Heinrich, R., Schuster, S.: The regulation of cellular systems, vol. 416. Chapman & Hall, New York (1996)
25. Heinrich, R., Schuster, S.: The modelling of metabolic systems. structure, control and optimality. Biosystems 47(1), 61–77 (1998)
26. Helton, J.C., Davis, F.J.: Illustration of sampling-based methods for uncertainty and sensitivity analysis. Risk Analysis 22(3), 591–622 (2002)
27. Helton, J.C., Davis, F.J.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliability Engineering and System Safety 81, 23–69 (2003)
28. Henri, V.: Lois générales de l'action des diastases. Librairie Scientifique A. Hermann (1903)
29. Hill, A.V.: A new mathematical treatment of changes of ionic concentration in muscle and nerve under the action of electric currents, with a theory as to their mode of excitation. The Journal of Physiology 40(3), 190–224 (1910)
30. Holmberg, C.I., Tran, S.E.F., Eriksson, J.E., Sistonen, L.: Multisite phosphorylation provides sophisticated regulation of transcription factors. Trends in Biochemical Sciences 27(12), 619–627 (2002)
31. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: Copasi-a complex pathway simulator. Bioinformatics 22(24), 3067–3074 (2006)
32. Hoppenstaedt, F.C., Peskin, C.S.: Modeling and Simulation in Medicine and the Life Sciences. Springer, New York (2002)
33. Horst, R., Tuy, H.: Global optimization: Deterministic approaches. Springer, Berlin (1990)
34. Iancu, B., Czeizler, E., Czeizler, E., Petre, I.: Quantitative refinement of reaction models. International Journal of Unconventional Computing (page to appear, 2013)
35. Jones, C.M., Henry, E.R., Hu, Y., Chan, C.-K., Luck, S.D., Bhuyan, A., Roder, H., Hofrichter, J., Eaton, W.A.: Fast events in protein folding initiated by nanosecond laser photolysis. Proceedings of the National Academy of Sciences 90(24), 11860–11864 (1993)
36. Kline, M.P., Morimoto, R.I.: Repression of the heat shock factor 1 transcriptional activation domain is modulated by constitutive phosphorylation. Molecular and Cellular Biology 17(4), 2107–2115 (1997)
37. Klipp, E., Herwig, R., Kowald, A., Wierling, C., Lehrach, H.: Systems biology in practice: concepts, implementation and application. Wiley-Vch (2005)
38. Kühnel, M., Mayorga, L.S., Dandekar, T., Thakar, J., Schwarz, R., Anes, E., Griffiths, G., Reich, J.: Modelling phagosomal lipid networks that regulate actin assembly. BMC Systems Biology 2, 107–121 (2008)
39. Lotka, A.J.: Elements of Physical Biology. Williams & Wilkins Company (1925)
40. Malthus, T.R.: An Essay on the Principle of Population. 1798

41. McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2), 239–245 (1979)
42. Mendes, P., Kell, D.: Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14(10), 869–883 (1998)
43. Menten, L., Michaelis, M.: Die kinetik der invertinwirkung. Biochem. Z. 49, 333–369 (1913)
44. Miller, W.G., Alberty, R.A.: Kinetics of the reversible michaelis-menten mechanism and the applicability of the steady-state approximation. Journal of the American Chemical Society 80(19), 5146–5151 (1958)
45. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Research 13(11), 2467–2474 (2003)
46. Murphy, E., Danos, V., Féret, J., Krivine, J., Harmer, R.: Rule-based modeling and model refinement. Elements of Computational Systems Biology, 83–114 (2009)
47. Murray, J.D.: Mathematical Biology I: An Introduction. Springer, New York (2002)
48. Murray, J.D.: Mathematical Biology II: Spatial Models and Biomedical Applications. Springer, New York (2002)
49. Nelson, D.L., Cox, M.M.: Lehninger principles of biochemistry. Worth Publishers (2000)
50. Oberguggenberger, M., King, J., Schmelzer, B.: Classical and imprecise probability methods for sensitivity analysis in engineering: A case study. International Journal of Approximate Reasoning 50, 680–693 (2009)
51. Orth, J.D., Thiele, I., Palsson, B.: What is flux balance analysis? Nature Biotechnology 28, 245–248 (2010)
52. Peper, A., Grimbergen, C.A., Spaan, J.A.E., Souren, J.E.M., Van Wijk, R.: A mathematical model of the hsp70 regulation in the cell. International Journal of Hyperthermia 14(1), 97–124 (1998)
53. Petre, I., Mizera, A., Hyder, C.L., Meinander, A., Mikhailov, A., Morimoto, R.I., Sistonen, L., Eriksson, J.E., Back, R.-J.: A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. Natural Computing 10(1), 595–612 (2011)
54. Rabitz, H., Kramer, M., Dacol, D.: Sensitivity analysis in chemical kinetics. Annual Review of Physical Chemistry 34(1), 419–461 (1983)
55. Rieger, T.R., Morimoto, R.I., Hatzimanikatis, V.: Mathematical modeling of the eukaryotic heat-shock response: Dynamics of the hsp70 promoter. Biophysical Journal 88(3), 1646 (2005)
56. Rubinow, S.I.: Introduction to Mathematical Biology. John Wiley, New York (1975)
57. Stewart, W.J.: Probability, Markov chains, queues, and simulation. The mathematical basis of performance modeling. Princeton University Press, Princeton (2009)
58. Taubes, C.: Modeilng Differential Equations in Biology. Prentice Hall, Upper Saddle River (2001)
59. Turányi, T.: Sensitivity analysis of complex kinetic systems. tools and applications. Journal of Mathematical Chemistry 5(3), 203–248 (1990)
60. Volterra, V.: Animal ecology. In: Chapman, R.N. (ed.), pp. 409–448. McGraw-Hill, New York (1926)
61. Wilkinson, D.J.: Stochastic modelling for systems biology. Chapman & Hall/CRC Mathematical Biology and Medicine Series (2006)
62. Zi, Z.: Sensitivity analysis approaches applied to systems biology models. Systems Biology, IET 5(6), 336–346 (2011)

Paper II

On the implementation of quantitative model refinement

Bogdan Iancu, Diana-Elena Gratie, Sepinoud Azimi, and Ion Petre

Originally published in Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe (Eds.), *Algorithms for Computational Biology*. Vol. 8542 of Lecture Notes in Computer Science, pp. 95–106. Springer International Publishing, 2014. The publication is available at Springer via `http://dx.doi.org/10.1007/978-3-319-07953-0_8`.

On the Implementation of Quantitative Model Refinement

Bogdan Iancu[1,2,3], Diana-Elena Gratie[1,2,3], Sepinoud Azimi[1,2,3], and Ion Petre[1,2,3]

[1] Computational Biomodeling Laboratory
[2] Turku Centre for Computer Science
[3] Department of IT, Åbo Akademi University
Joukahainengatan 3-5, 20520 Åbo, Finland
{biancu,dgratie,sazimi,ipetre}@abo.fi

Abstract. The iterative process of adding details to a model while preserving its numerical behavior is called *quantitative model refinement*, and it has been previously discussed for ODE-based models and for *kappa*-based models. In this paper, we investigate and compare this approach in three different modeling frameworks: rule-based modeling, Petri nets and guarded command languages. As case study we use a model for the eukaryotic heat shock response that we refine to include the acetylation of the heat shock factor. We discuss how to perform the refinement in each of these frameworks in order to avoid the combinatorial state explosion of the refined model. We conclude that Bionetgen (and rule-based modeling in general) is well-suited for a compact representation of the refined model, Petri nets offer a good solution through the use of colors, while the PRISM refined model may be much larger than the basic model.

Keywords: Quantitative model refinement, heat shock response, acetylation, rule-based modeling, Petri nets, model checking.

1 Introduction

Systems biology aims to holistically characterize highly complex biological systems. A hierarchical system-level representation is very adequate in this context. Formal frameworks turn out to be fundamental in the effort of understanding the behavior of such complex systems, see [21,12]. The abstractions that lie at the core of these formalisms need to be refined to incorporate more details.

We focus in this paper on the implementation of model refinement. Within the model development process, we examine *data refinement* through three different frameworks – *rule-based modeling, Petri nets* and *guarded command languages* – and discuss their capabilities for the efficient construction of a refined model. For rule-based modeling we used the Bionetgen framework and RuleBender, for Petri nets we chose Snoopy and Charlie as modeling tools, while for modeling with guarded command languages we used PRISM. Data refinement, as described in [3] and [10], assumes the replacement of one species in the model with several of its variants, called subspecies. This type of refinement is adequate for representing post-translational modifications of proteins, e.g., acetylation, phosphorylation, etc. Given a protein P, one can indicate its state regarding

A.-H. Dediu, C. Martín-Vide, and B. Truthe (Eds.): AlCoB 2014, LNBI 8542, pp. 95–106, 2014.

post-translational modifications by replacing it with its variants. This substitution also implies a refinement of all complexes involving protein P and of all reactions involving either P or any such complex, see [10]. This might induce a combinatorial state explosion of the refined model, as in the case of ODE-based models, see [10]. The main question we are answering is whether one can avoid this problem in the other three frameworks we investigate in this paper and build a compact representation of the refined model.

We consider as a case study for our analysis the heat shock response mechanism, as described in [20] and [10]. Throughout the paper, the model in [20] will be referred to as the *basic* heat shock response model, while the model in [10] will be referred to as the *refined model*.

All models developed in this paper are available for download at [11]. Due to space restrictions, some of the details of this work were omitted. For full details, we refer the reader to [6].

2 The Heat Shock Response (HSR)

The eukaryotic *heat shock response* is a highly conserved bio-regulatory network that controls cellular function impairment produced by protein misfolding as a result of high temperatures. Elevated temperatures have proteotoxic effects on proteins, inducing protein misfolding and leading to the formation of large aggregates that thereafter trigger apoptosis (controlled cell death). Cell survival is promoted by a defense mechanism, which consists in restoring protein homeostasis by augmenting the level of molecular chaperones, see [22].

We consider the basic molecular model for the eukaryotic heat shock response proposed in [20]. *Heat shock proteins* (hsp's) play a key role in the heat shock response mechanism by chaperoning the *misfolded proteins* (mfp's). Due to their affinity to mfp's, hsp's form hsp:mfp complexes and help the misfolded proteins refold. The heat shock response is regulated by the transactivation of the hsp-encoding genes. In eukaryotes, some specific proteins, called *heat shock factors* (hsf's), promote gene transcription. In the absence of environmental stressors, heat shock factors are predominantly found in a monomeric state, extensively bound to heat shock proteins. Raising the temperature causes the correctly folded proteins (prot) to misfold and hsp:hsf complexes to break down. This switches on the heat shock response by releasing hsf's, which quickly reach a DNA binding competent state, see [20,23].

Heat stress induces dimerization (hsf_2) and, subsequently, trimerization (hsf_3) of hsf's, enabling the binding of the hsf trimers to the promoter site of the hsp-encoding gene, called *heat shock element* (hse). Subsequently, DNA binding triggers the transcription and translation of the hsp-encoding gene, inducing hsp synthesis, see [20,22]. Once the level of heat shock proteins is sufficiently elevated for the cell to withstand thermal stress, hsp synthesis is turned off. Heat shock proteins sequestrate heat shock factors and break hsf dimers and trimers, constituting hsp:hsf complexes. The explicit molecular reactions constituting the model can be found in [20].

The numerical setup of the basic model (in terms of initial concentrations and kinetic constants) can be found in [20]. Acetylation has been shown to have an extensive influ-

ence in regulating the heat shock response, we refer the reader to [24]. To this end, we consider the acetylation of heat shock factors implemented through data refinement.

3 Quantitative Model Refinement

Quantitative model refinement was investigated in [19,4] regarding rule based modeling and applied to two ampler ODE-based models in [18,10].

3.1 Quantitative Model Refinement

A reaction-based model can be refined to incorporate more information regarding its reactants and/or reactions. There are two types of refinement, either of the *data* (data refinement) or of the *reactions* (process refinement). In this study, we focus on the first refinement type. Considering that one's interest lies especially on data, a species in a model could be refined by replacing it with several of its subspecies, a routine called *data refinement*. When the interest is focused on reactions, the model can be refined by replacing a collective reaction, accounting for a specific process, by a set of reactions depicting the transitional steps of the process. The last type of refinement is called *process refinement*, see [10].

The notion of quantitative model refinement has been previously addressed in systems biology in the context of rule based modelling, see [19,4,7,5]. The rule based modelling framework embodies the concept of *data refinement*, as previously introduced, implementing agent resolution as a fundamental constituent, [7]. The key refinement method in this context is rule refinement, an approach that requires the refinement of the set of rules ensuring the preservation of the dynamic behavior of the system, see [19].

We present here the *quantitative model refinement* of reaction models following the discussion in [10]. Consider a model M, comprising a number m of species $\Sigma = \{A_1, A_2, \ldots, A_m\}$ and n of reactions r_i, $1 \leq i \leq n$, as follows:

$$r_i \ : \ S_{i,1}A_1 + S_{i,2}A_2 + \ldots + S_{i,m}A_m \xrightarrow{k_i} S'_{i,1}A_1 + S'_{i,2}A_2 + \ldots + S'_{i,m}A_m,$$

where $S_{i,1}, \ldots, S_{i,m}, S'_{i,1}, \ldots, S'_{i,m} \geq 0$ are the stoichiometric coefficients of r_i and $k_i \geq 0$ is the kinetic rate constant of r_i. We discuss here a continuous, mass-action formulation of the model based on ODEs. For some details on this approach we refer to [14].

Model M can be refined to distinguish between various subspecies of any species in the model, for example, A_1. The distinction between the subspecies is very often drawn by post-translational modifications such as acetylation, phosphorylation, sumoylation, etc. All previously mentioned subspecies of A_1 take part in all reactions A_1 engaged in, conceivably obeying a different kinetic setup. Given model M and species A_1, substituting subspecies $B_1, \ldots, B_l$ for species A_1 in M leads to attaining a new model M_R, comprising species $\{A'_2, A'_3, \ldots, A'_m\} \cup \{B_1, \ldots, B_l\}$, for some $l \geq 2$, where variables A'_i, $2 \leq i \leq m$ from M_R, coincide with A_i from model M and $B_1, \ldots, B_l$ substitute for species A_1 in M_R. Furthermore, each reaction r_i of M is replaced in the new model M_R by all possible reactions $r_{i,j}$ of the following form:

$$r_{i,j} \ : \ (T^j_{i,1}B_1 + \ldots + T^j_{i,l}B_l) + S_{i,2}A'_2 + \ldots + S_{i,m}A'_m \xrightarrow{k_{i,j}} (T'^j_{i,1}B_1 + \ldots + T'^j_{i,l}B_l) + S'_{i,2}A'_2 + \ldots + S'_{i,m}A'_m,$$

where $k_{i,j}$ is the kinetic rate constant of $r_{i,j}$ and $(T^j_{i,1}, \ldots, T^j_{i,l}, T'^j_{i,1}, \ldots, T'^j_{i,l})$ are all possible nonnegative integers so that $T^j_{i,1} + \ldots + T^j_{i,l} = S_{i,1}$ and $T'^j_{i,1} + \ldots + T'^j_{i,l} = S'_{i,1}$. Model M_R is said to be a *data refinement of model M on variable* A_1 if and only if the following conditions are fulfilled:

$$[A_i](t) = [A'_i](t), \tag{1}$$

$$[A_1](t) = [B_1](t) + \ldots + [B_l](t), \tag{2}$$

for all $2 \leq i \leq m$, $t \geq 0$. Fulfilling these conditions depends on the numerical setup of model M_R, i.e., on the kinetic constants of its reactions (both those adopted from the basic model, as well as those newly introduced in the construction) and on the initial concentrations of its species.

3.2 Adding the Acetylation Details to the HSR Model through Data Refinement

We start from the basic model of the heat shock response, introduced in [20], where no post-translational modification of hsf is taken into account, and we refine all species and complexes that involve hsf taking into consideration one acetylation site for every hsf molecule. We follow here the discussion in [10]. The aim is to refine the basic model and preserve its numerical properties. For hsf_2, hsf_3, hsf_3:hse and hsp:hsf, the refinement is performed conforming to the number of hsf constituents respectively. This leads to the following data refinements: $\mathsf{hsf} \rightarrow \{\mathsf{rhsf}, \mathsf{rhsf}^{(1)}\}$; $\mathsf{hsf}_2 \rightarrow \{\mathsf{rhsf}_2, \mathsf{rhsf}_2{}^{(1)}, \mathsf{rhsf}_2{}^{(2)}\}$; $\mathsf{hsp{:}hsf} \rightarrow \{\mathsf{rhsp{:}rhsf}, \mathsf{rhsp{:}rhsf}^{(1)}\}$; $\mathsf{hsf}_3 \rightarrow \{\mathsf{rhsf}_3, \mathsf{rhsf}_3{}^{(1)}, \mathsf{rhsf}_3{}^{(2)}, \mathsf{rhsf}_3{}^{(3)}\}$; $\mathsf{hsf}_3\mathsf{:hse} \rightarrow \{\mathsf{rhsf}_3\mathsf{:rhse}, \mathsf{rhsf}_3{}^{(1)}\mathsf{:rhse}, \mathsf{rhsf}_3{}^{(2)}\mathsf{:rhse}, \mathsf{rhsf}_3{}^{(3)}\mathsf{:rhse}\}$. The refinement based on the above data refinements involves substantial changes in the list of reactions. For example, the reversible reaction of dimerization $2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$ in the basic model is replaced by three reactions as follows: $2\,\mathsf{rhsf} \rightleftarrows \mathsf{rhsf}_2$; $\mathsf{rhsf} + \mathsf{rhsf}^{(1)} \rightleftarrows \mathsf{rhsf}_2{}^{(1)}$; $2\mathsf{rhsf}^{(1)} \rightleftarrows \mathsf{rhsf}_2{}^{(2)}$.

The refined model of [10] consists of 20 species and 55 irreversible reactions, compared to 10 species and 17 irreversible reactions in the basic model of [20].

4 Quantitative Refinement in Rule-Based Models

4.1 A RuleBender Implementation of the Basic HSR Model

This section focuses on the RuleBender implementation of the basic heat shock response model, as introduced in Section 2. We model all reactions to follow the principle of mass action. Conforming to the implementation presented here, Bionetgen source code comprises a set of twelve rules, which generate a total number of seventeen irreversible reactions. Due to the symmetry that some of the species exhibit, the collision frequency (e.g. in our case dimerization, trimerization, etc) and the existence of multiple paths from substrates to products in some reactions (e.g. for the heat shock response

model, the unbinding of trimers), kinetic rate constants for those specific reactions are multiplied in Bionetgen by diverse symmetry and/or statistical factors, see [2]. For example, the collision frequency of two different types of reactants A and B, $A + B$, is twice that of identical types of reactants $A + A$. Another example concerns the multiple reaction paths from reactants to products, which may generate statistical factors. Preserving the fit of the heat shock response model attained in [20] required a multiplication of some rate constants by the inverse of the aforementioned factors respectively.

RuleBender generates during the process of model development a contact map which depicts the connectivity between the molecules. The contact map for the basic model of the heat shock response is shown in Figure 1.

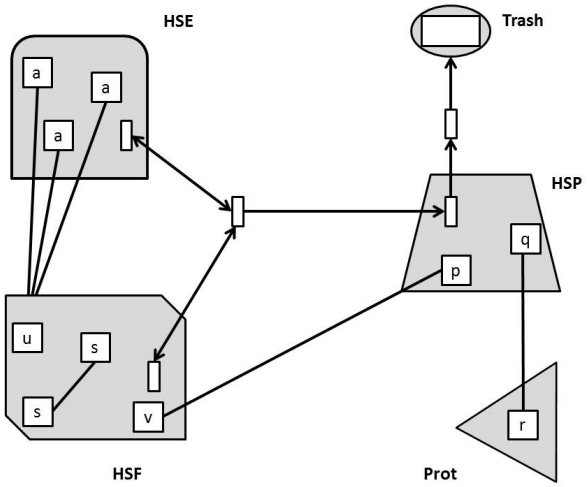


Fig. 1. The RuleBender generated contact map for the basic model of the heat shock response. It depicts the possible interconnections among the model's species.

One can notice in Figure 1 that hsf's have been represented as having 4 sites (s, s, u, v). The two s sites are involved in the generation of dimers and trimers. The other two sites, u and v, are used to illustrate the process of DNA binding/unbinding and hsf sequestratation/dimer (trimer) dissipation. Trimers are considered to be circular structures, each of the 's' site of one hsf being bound to the 's' sites of the consequent hsf's, no two hsf's having both sites 's' bound to the same partner. The promoter, hse, has been represented as having three identical sites (a, a, a), so as to be connected to the trimer in such a way that the symmetry is not affected. Heat shock proteins are modeled to have two sites 'p' and 'q', used for the modelling of unbinding of dimers and trimers and for the sequestration of misfolded proteins. The model takes into account a species called Prot, which has a site with two possible states, one of which accounts for misfolded proteins 'm' and another one 'f', that accounts for folded proteins. A "dummy" component, called Trash, has been introduced to help encode the degradation of heat shock proteins.

The contact map in Figure 1 illustrates the connectivity between the species in the model. The link between the 's' sites of the hsf molecule denotes the formation of dimers and trimers through the agency of these sites. Once trimers are formed, they

can bind to the heat shock element (hse), the connection being illustrated by three links connecting hsf trimers to the heat shock element (one can notice three 'a' sites the heat shock element component exhibits). The middle connector encodes for a number of reactions, such as: DNA unbinding, HSP synthesis and breaking of dimers and trimers. The link between the site 'v' of the hsf component and the site 'p' of the hsp component illustrates hsf sequestration. The link between the hsp component and the prot component encodes the following reactions: protein misfolding, protein refolding and mfp sequestration. By linking the component *Trash* to the hsp component, we encoded for the degradation of hsp's.

We chose a deterministic simulation for the basic model. The simulation results for DNA binding for a temperature of 42°C showed that RuleBender prediction are in accordance with the results reported in [20].

4.2 A RuleBender Implementation of the Acetylation-Refined HSR Model

We focus in this section on the acetylation-refinement of the heat shock response, as described in [10]. There are several changes to do in Rulebender to refine the basic model so as to include the acetylation of hsf's. The syntax of the rules remains, in this case, unchanged, since all reactions, in this model, take place regardless of the acetylation status of the molecules. We brought changes in the definition of hsf's, by introducing one acetylation site, 'w', which can be either acetylated or not, and in the initial concentrations of the molecules. The initial concentrations were set conforming to [10].

As expected from the refinement conditions, the simulation of the refined model for a temperature of 42°C showed that the Rulebender prediction for the refined model and the one for the basic model are the same.

5 Quantitative Refinement in Petri Net Models

5.1 A Petri Net for the Basic HSR Model

A standard Petri net model for the heat shock response was previously reported in [1]. We focus here on a Snoopy continuous Petri net implementation of the basic heat shock response model, shown in Figure 2. The network has 10 places and 17 transitions, encoding the 10 species and 17 irreversible reactions in the basic model definition of [20]. Verifying the model required the analysis of several properties. For instance, the model is covered by T-invariants; also, the P-invariants reported by Charlie encode all mass conservation relations reported in the ODE-based model of [20]. Moreover, all places except HSP are covered by P-invariants, which means that they are bounded. The three mass conservation relations yield three constants (accounting for the total amount of HSF, HSE and protein molecules in the system, respectively), that have been used in the PRISM implementation of the model.

5.2 Petri Nets for the Acetylation-Refined HSR Model

For the refined heat shock response that includes two types of hsf's (acetylated and non-acetylated [10]), we chose an implementation based on colored continuous Petri nets.

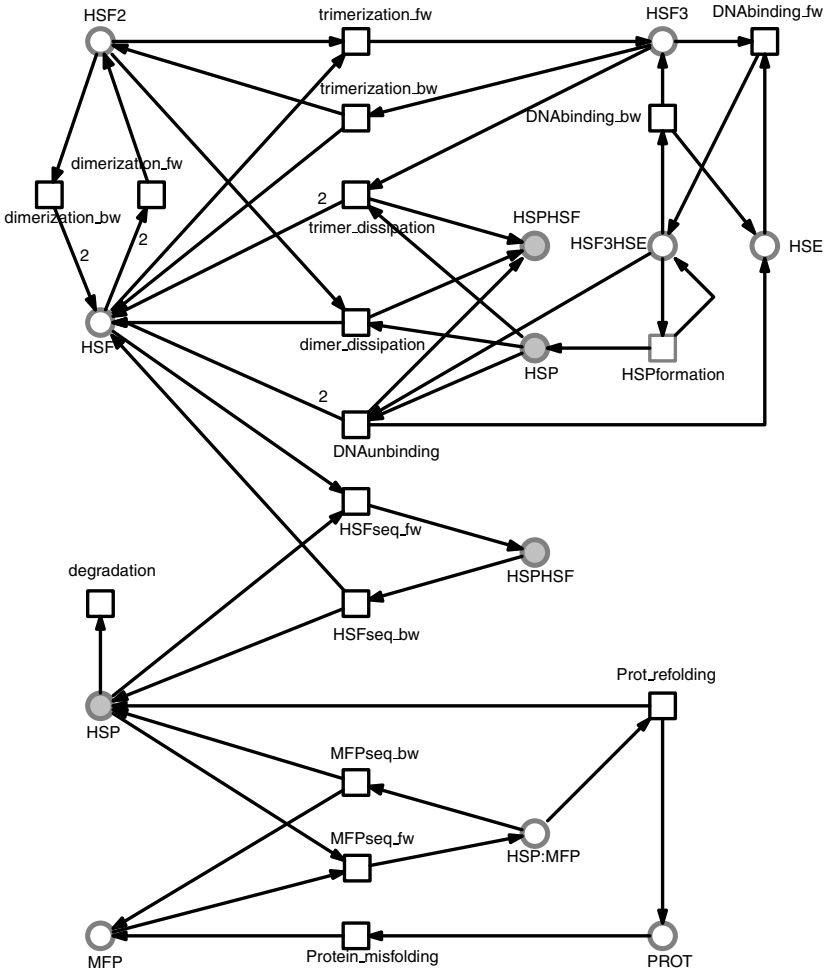


Fig. 2. Snoopy implementation of the basic heat shock response model. The text next to a place (transition) denotes the identifier of that particular place (transition). Arc multiplicities greater than 1 are included in the picture. The dashed gray circles are logical places (they may appear several times, but they represent the same species).

There are several ways of reasoning about refined species within this framework. For example, the dimer of a protein with a site that can be acetylated (1) or non-acetylated (0) can be either seen as an entity with 0, 1, or 2 acetylated sites, or as a compound where the order of the acetylated sites counts (i.e. (0,0), (0,1), (1,0), (1,1)). Depending on the approach one takes, the colored representation will have different color sets, different number of transitions and different kinetic constants.

We modeled the refined heat shock response using two approaches: one focused on keeping the structure of the basic model intact, with the same transitions and kinetic constants (we call this model *transition-focused*). This is the most compact representation. The other approach aimed to minimize the number of colors used in the model (we call this model *color-focused*). This approach uses as few colors as possible, at the cost of a complicated representation, with many conditions in a transition, and also introducing new transitions in the colored representation. Due to space limitations, we present here only the color-focused model. A more detailed description of both approaches can be found in [6].

Several choices had to be made during the modeling process. We detail the modeling options for the dimerization and trimerization of acetylated and non-acetylated hsf's. There are three types of dimers that can be formed: non-acetylated ($\mathsf{hsf_2}^{(0)}$), single-acetylated ($\mathsf{hsf_2}^{(1)}$) and double-acetylated dimer ($\mathsf{hsf_2}^{(2)}$). One way of modeling the dimers is using a color set with three colors of type int (0, 1, and 2 denoting the number of acetylated sites). Another approach is using a cartesian product $\{0,1\} \times \{0,1\}$. When modeling hsf trimers, one could consider, for example, one of the following three color sets: a color set $\mathsf{Tri} = \{0,1,2,3\}$, a compound color set $\mathsf{Compound} = \{0,1\} \times \{0,1,2\}$, or a compound set $\mathsf{Trimer} = \{0,1\} \times \{0,1\} \times \{0,1\}$. For the color-focused refinement, we chose the simple integer color sets.

All reactions involving the decomposition of complexes containing hsf's required additional transitions. For example, the trimer dissipation reaction $\mathsf{hsf_3} + \mathsf{hsp} \rightarrow \mathsf{hsp}: \mathsf{hsf} + 2\,\mathsf{hsf}$ is split into three transitions. One covers the case when all hsf's in the trimer have the same acetylation value (i.e. $\mathsf{hsf_3}$ has color 0 or 3). In this case, there is no distinction between which hsf binds to hsp and which two hsf's become unbound, and the kinetic constant for this transition is the same as the corresponding one in the basic model. When $\mathsf{hsf_3}$ has color 1 or 2, there are two binding possibilities: hsp binds to either a non-acetylated hsf, or to an acetylated hsf. For the two transitions representing these possibilities, the kinetic constant is half of the corresponding one in the basic model (following the reasoning explained in [10]).

When simulating a colored Petri net, Snoopy first unfolds it, in other words it creates an equivalent Petri net. Each place instance (each color) will correspond to a place in the unfolded net, and each transition instance (each binding) will correspond to a transition in the unfolded net; for details on colored Petri nets unfolding, see [17]. The color-focused refined model has 10 places and 25 transitions, and its corresponding flattened Petri net has 20 places and 56 transitions. This representation, although more complex than the transition-focused one, encodes a smaller flattened network. Both the transition- and color-based refinements have been compared with the basic model predictions, and they are all equivalent (data not shown).

6 Quantitative Refinement in PRISM Models

6.1 A PRISM Implementation of the Basic HSR Model

We implemented the basic heat shock response as a CTMC model that defines all possible guards (in this case reactions) within a single module. The PRISM model consists of 10 variables, each of them corresponding to one of the reactants in the model, and 17 guards representing the 17 irreversible reactions of the system. The values for upper bounds of the variables are taken from our Petri net model's P-invariants and mass-conservation relations. Upper bounds are used both for allocating memory and in the guarded commands. For example the guard corresponding to *dna binding* is expressed as follows: $\mathsf{hsf_3} >= 1 \wedge \mathsf{hse} >= 1 \wedge \mathsf{hsf_3}\mathsf{:hse} <= N-1 \rightarrow \mathsf{hsf_3} * \mathsf{hse} * k_5 : (\mathsf{hsf_3}' = \mathsf{hsf_3} - 1) \wedge (\mathsf{hse}' = \mathsf{hse} - 1) \wedge (\mathsf{hsf_3}\mathsf{:hse}' = \mathsf{hsf_3}\mathsf{:hse} + 1)$, where N represents the upper bound for hse in the system.

It is noteworthy to mention that the PRISM model could be obtained from the Petri net model via some format manipulations in Snoopy. However, we decided to write the

model from the very beginning in order to be able to compare the modeling effort in each chosen framework.

6.2 A PRISM Implementation of the Acetylation-Refined HSR Model

The approach we took in Sections 4 and 5 to implement the acetylation-refined heat shock response model was through a compact representation of the acetylated species. Whereas colors of the places and arc expressions were employed to represent the refinement in the Petri net model, in modeling with RuleBender the solution was to introduce a new acetylation site for every hsf molecule. Both methods used structured data types for the species, thus concealing the complexity of the model in a compact representation. In PRISM this requires a method to represent the acetylation details in the definition of hsf, i.e. a composite data type. Since PRISM currently supports only simple data type (e.g. integer, boolean) variables in the model, such a definition is not possible. Alternatively, we implemented the acetylation-refined model through introducing new variables describing all possible acetylation configurations of hsf and hsf complexes. This was similar to the ODE-based approach to quantitative model refinement discussed in [10].

The refined heat shock response model is built based on the refinements given in Section 3.2 by refining all reactants and complexes involving hsf. In this approach, the strategy is to replace each guard involving any refined reactant by the guards considering all possible refined reactions.

One could also use parallel modules to implement the refinement but this approach would not help reducing the complexity of the model.

The complete PRISM implementation of the refinement is not listed here due to space limitations. The numerical setup of this model is based on [10].

6.3 Model Checking of the HSR Models

According to [15], the maximum number of states that PRISM can handle for CTMCs is 10^{10}. In both our models (basic and refined version of the heat shock response), the number of all possible states in the system exceeds this limit. This is a known problem for biological systems in PRISM, see [8]. Several studies have addressed this issue, see e.g., [9,16,8]. One of the investigated approaches is *approximate verification* of probabilistic systems, where a Monte-Carlo algorithm is used to approximate the probability of a temporal formula to be true, see [9]. We used this method to verify the desired properties of the heat shock response model. In this approach a large number of stochastic paths is sampled for the model and based on the defined properties, the result for each run is obtained. The information produced in this way gives an approximate result for the probability that the desired property holds for the model.

We are interested in verifying two properties discussed in [20]. The properties are: (i) the validity of three mass-conservation relations and (ii) the level of DNA binding eventually returns to the basal values, both at $37°C$ and at $42°C$.

In order to check the mass conservation properties, we used the G operator which checks if the property remains true at all states along the path. The three properties we were interested in are listed as follows:

- $p =?$ [G $\mathsf{hsf} + 2\,\mathsf{hsf}_2 + 3\,\mathsf{hsf}_3 + 3\,\mathsf{hsf}_3{:}\,\mathsf{hse} + \mathsf{hsp}{:}\,\mathsf{hsf} = \mathsf{hsf}_{const}$],
- $p =?$ [G $\mathsf{hse} + \mathsf{hsf}_3{:}\,\mathsf{hse} = \mathsf{hse}_{const}$],
- $p =?$ [G $\mathsf{prot} + \mathsf{mfp} + \mathsf{hsp}{:}\,\mathsf{mfp} = \mathsf{prot}_{const}$],

where $\mathsf{hsf}_{const}, \mathsf{hse}_{const}$ and prot_{const} represent the total amounts of $\mathsf{hsf}, \mathsf{hse}$ and prot respectively. These properties check if the mass-conservation relations, corresponding to the level of $\mathsf{hsf}, \mathsf{hse}$ and prot, are valid in all the states. In each case, the value of p was confirmed to be one, which was to be expected, with confidence level 95%, i.e. the mass conservation laws are respected in the model.

For the second property, we verified in PRISM that for time points larger than 14400, the value of $\mathsf{hsf}_3{:}\,\mathsf{hse}$ reactants returns to their initial value. We formulated the following property: $p =? \quad [F >= 14400 \quad \mathsf{hsf}_3{:}\,\mathsf{hse} = 3]$. The probability value calculated by PRISM was one for this property as well, with confidence level 95%.

We also checked if the model confirms the experimental data of [13] on DNA binding. One approach could be to run the simulation for many times and plot the average run. Due to the memory issues of the PRISM, we were not able to follow this approach. Since we are using a stochastic model, our second approach was to check the probability of having a data point within the interval $[0.9 \cdot d, 1.1 \cdot d]$ in the time period $[0.9 \cdot t, 1.1 \cdot t]$, where d is the experimental data point at time t. The confidence interval for all the properties and the number of simulations were 95% and 150 respectively. We interpret the high values we obtained as a result as a confirmation that the two PRISM models are in accordance with the experimental data of [13].

7 Discussion

We focused in this paper on analyzing the capability of three different frameworks to implement the concept of quantitative model refinement: rule based modelling (with Bionetgen), Petri nets (with Snoopy) and guarded command languages (with PRISM). Handling the combinatorial explosion due to accounting for a post-translational modification throughout our refinement proved to be fundamentally different in the approaches we considered. These modeling methods are not restricted to the analysis of our case study solely, but their applicability extends to other reaction-based models. Rule-based modelling tackles the complexity of refinement through a compact model representation based on a partial presentation of the details of the model species, leading to more effective model construction and analysis techniques. Colored Petri nets integrate programmability by including data types (color sets) as an intrinsic property of places. The color set assignation reflects on the structure of the network, affecting the dimensions of the corresponding flattened network. PRISM model checker promotes a low level implementation of data structures and it does not allow the modeler to introduce more complex data structures.

Our study shows that some modeling frameworks are more suitable for model refinement than others, with respect to the compactness of the representation of the refined model. A key ingredient for this is the spectrum of internal data structures supported by the modeling framework. Data structures may encapsulate a large amount of information, and their effective manipulation can substantially reduce the complexity of a

model's representation. RuleBender provides data structures suitable for modeling biological systems: species, sites, links, partial description of species, rendering a straightforward refinement procedure with a very compact representation. In contrast, Petri nets are not primarily a biology-focused framework. Colored Petri nets introduce programmability in this modeling formalism, incorporating data types into the places of the network. New data types can be implemented based on primitive built-in types and composition rules. In refining a Petri net model, one has to define the appropriate data structures, and associate a biological meaning to each of them. The modeling choices affect both the compactness of the representation and the complexity of the corresponding flattened Petri net model. PRISM on the other hand only supports primitive data types. This translates into an explicit detailing of all elements of the refined model.

Our study shows that quantitative model refinement is a potentially viable approach to building a large biomodel. The approach can be used together with a multitude of modeling paradigms, allowing the modeler to increase the level of details of the model, while preserving its numerical behavior. Moreover, on any level of detail one can switch from a modeling paradigm to another, taking full advantage of the various analysis tools made possible in different model formulations, in terms of fast simulations, model checking or compact model representation. While our case-study shows the potential of the quantitative model refinement approach to model building, its scalability remains to be tested on a larger case study.

Acknowledgement. This research was partially supported by Academy of Finland under project 267915.

The authors thank Monika Heiner for her help with issues related to Snoopy and Charlie, James Faeder and Leonard Harris for advice on the Bionetgen implementation of the heat shock response, and Adam Smith for technical support regarding RuleBender.

References

1. Back, R., Ishdorj, T., Petre, I.: A petri net formalization of heat shock response model. In: Petre, I., Rozenberg, G. (eds.) Workshop on Natural Computing and Graph Transformations, pp. 19–28 (2008)
2. Blinov, M.L., Yang, J., Faeder, J.R., Hlavacek, W.S.: Graph theory for of biochemical networks. In: Priami, C., Ingólfsdóttir, A., Mishra, B., Riis Nielson, H. (eds.) Transactions on Computational Systems Biology VII. LNB, vol. 4230, pp. 89–106. Springer, Heidelberg (2006)
3. Czeizler, E., Rogojin, V., Petre, I.: The phosphorylation of the heat shock factor as a modulator for the heat shock response. In: Proceedings of the 9th International Conference on Computational Methods in Systems Biology, pp. 9–23. ACM (2011)
4. Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J.: Rule-based modelling and model perturbation. In: Priami, C., Back, R.-J., Petre, I. (eds.) Transactions on Computational Systems Biology XI. LNCS, vol. 5750, pp. 116–137. Springer, Heidelberg (2009)
5. Faeder, J., Blinov, M., Goldstein, B., Hlavacek, W.: Rule-based modeling of biochemical networks. Complexity 10(4), 22–41 (2005)
6. Gratie, D., Iancu, B., Azimi, S., Petre, I.: Quantitative model refinement in four different frameworks, with applications to the heat shock response. Tech. Rep. 1067, TUCS (2013)

7. Harmer, R.: Rule-based modelling and tunable resolution. EPTCS 9, 65–72 (2009)
8. Heath, J.K., Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O.: Probabilistic model checking of complex biological pathways. In: Priami, C. (ed.) CMSB 2006. LNB, vol. 4210, pp. 32–47. Springer, Heidelberg (2006)
9. Hinton, A., Kwiatkowska, M., Norman, G., Parker, D.: Prism: A tool for automatic verification of probabilistic systems. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 441–444. Springer, Heidelberg (2006)
10. Iancu, B., Czeizler, E., Czeizler, E., Petre, I.: Quantitative refinement of reaction models. IJUC 8(5-6), 529–550 (2012)
11. Iancu, B., Gratie, D., Azimi, S., Petre, I.: Computational modeling of the eukaryotic heat shock response: The bionetgen implementation, the petri net implementation and the prism implementation (2013), http://combio.abo.fi/research/computational-modeling-of-the-eukaryotic-heat-shock-response/
12. Kitano, H.: Systems biology: a brief overview. Science 295(5560), 1662–1664 (2002)
13. Kline, M., Morimoto, R.: Repression of the heat shock factor 1 transcriptional activation domain is modulated by constitutive phosphorylation. Molecular and cellular biology 17(4), 2107–2115 (1997)
14. Klipp, E., Herwig, R., Kowald, A., Wierling, C., Lehrach, H.: Systems biology in practice: Concepts, implementation and application. Wiley-Vch (2005)
15. Kwiatkowska, M., Norman, G., Parker, D.: Quantitative analysis with the probabilistic model checker prism. Electronic Notes in Theoretical Computer Science 153(2), 5–31 (2006)
16. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011)
17. Liu, F., Heiner, M., Yang, M.: An efficient method for unfolding colored petri nets. In: Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., Uhrmacher, A. (eds.) Proceedings of the Winter Simulation Conference, vol. 295. Winter Simulation Conference (2012)
18. Mizera, A., Czeizler, E., Petre, I.: Self-assembly models of variable resolution. In: Priami, C., Petre, I., de Vink, E. (eds.) Transactions on Computational Systems Biology XIV. LNCS, vol. 7625, pp. 181–203. Springer, Heidelberg (2012)
19. Murphy, E., Danos, V., Feret, J., Krivine, J., Harmer, R.: Rule Based Modelling and Model Refinement. In: Elements of Computational Systems Biology, pp. 83–114. Wiley Book Series on Bioinformatics, John Wiley & Sons, Inc. (2010)
20. Petre, I., Mizera, A., Hyder, C., Meinander, A., Mikhailov, A., Morimoto, R., Sistonen, L., Eriksson, J., Back, R.: A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. Natural Computing 10(1), 595–612 (2011)
21. Raman, K., Chandra, N.: Systems biology. Resonance 15(2), 131–153 (2010)
22. Rieger, T., Morimoto, R., Hatzimanikatis, V.: Mathematical modeling of the eukaryotic heat-shock response: Dynamics of the hsp70 promoter. Biophysical Journal 88(3), 1646–1658 (2005)
23. Shi, Y., Mosser, D., Morimoto, R.: Molecular chaperones as hsf1-specific transcriptional repressors. Genes & Development 12(5), 654–666 (1998)
24. Westerheide, S., Anckar, J., Stevens Jr., S., Sistonen, L., Morimoto, R.: Stress-inducible regulation of heat shock factor 1 by the deacetylase sirt1. Science Signalling 323(5917), 1063–1066 (2009)

Paper III

Hiding the combinatorial state space explosion of biomodels through colored Petri nets

Diana-Elena Gratie and Ion Petre

Originally published in *Annals of University of Bucharest.* Vol. LXI, pp. 23–41. Editura Universității din București, 2014.

HIDING THE COMBINATORIAL STATE SPACE EXPLOSION OF BIOMODELS THROUGH COLORED PETRI NETS

DIANA-ELENA GRATIE and ION PETRE

Computational Biomodeling Laboratory,
Turku Centre for Computer Science and
Department of IT, Åbo Akademi University
Joukahainengatan 3-5, FIN-20520 Åbo
{dgratie,ipetre}@abo.fi

Abstract

Model refinement is an important step in the model implementation cycle that deals with adding details to an existing model. Several ways of implementing model refinement have been discussed in the literature, for rule-based models and for ODE models. We focus here on implementing model refinement in the framework of Petri nets, using the programming capabilities of colored Petri nets. We exemplify our strategy on a reaction-based model of the eukaryotic heat shock response. We conclude with an analysis of the initial and refined models, a proof that the two colored Petri net models we have built are bisimilar, and a discussion on how modeling biological systems with colored Petri nets scales with further expansions of the model.

1 Introduction

One of the steps often required in modeling is model refinement, i.e. increasing the level of detail of a model to include more information. This process can be implemented either starting a new model from scratch, and doing all the model fitting steps, or start from an existing fitted model to which details are added in such a way that the model fit is preserved. The latter method is called *data refinement* and has been introduced in [3,14] for rule-based models, and discussed in [6,4] in the context of ODE models.

Throughout this paper we consider *reaction-based models* consisting of a list of reactions of the type $c_1A_1 + c_2A_2 + ... + c_nA_n \rightarrow c'_1B_1 + c'_2B_2 + ... + c'_mB_m$, with $m, n \geq 0$, where A_i, B_j are molecular species representing the reactants (substrate) and products of the reaction, respectively, and c_i, c'_j are the stoichiometric coefficients (multiplicities), with $1 \leq i \leq n, 1 \leq j \leq m$. The mathematical semantic for such a model can be defined both in terms of continuous mathematics or in terms of discrete mathematics.

A reaction-based model can be refined to incorporate more information regarding its reactants, a process called *data refinement*. In this paper we consider the implementation of data refinement as presented in [5]. The species of a model are considered to be either *atomic* or *complex*, where a complex species contains in its structure at least two (possibly identical) atomic species. Refinement can be done on atomic species only, and it implies replacing the species with several of its variants. The change propagates throughout the model to all complex species that contain the atomic species being refined. Depending on the composition of complex species, one small refinement of an atomic species can induce an explosion in the number of species in the refined model, and consequently in the number of reactions. For details on the size of this explosion, see [5].

We focus in this paper on a Petri net approach of refinement, with the goal of obtaining a compact representation of the refinement of a model. To this end, we use colored Petri nets, a variant of high-level Petri nets that are "programmable" by means of data types (color sets), variables and functions.

As proof of concept, we implement the refinement of a model of the eukaryotic heat shock response. We construct a Petri net model that we subsequently refine to include more biological details. The refinement of the model is compact; the structure of the Petri net (in terms of places, transitions and arcs connecting them) remains the same. All details that are added to the network are encoded by means of colors. Our focus is on the ability of the [colored] Petri net framework to scale up with model refinements.

The paper is organized as follows: we start with a short overview of the Petri net formalism and its use in modeling biological systems, in Section 2. We continue with the biological semantics of the eukaryotic heat shock response, our case study, in Section 3. We also present here the molecular model for the heat shock response mechanism proposed in [15], and its refinement that accounts for the acetylation of one of the species, as introduced in [6]. In Section 4 we present our Petri net model for the heat shock response, and in Section 5 our modeling of its refinement as a colored Petri net. We prove that the two networks are bisimilar in Section 6, and draw some conclusions in Section 7.

2 Preliminaries on Modeling with Petri Nets

2.1 The Petri Net Formalism

Petri nets are a sound formalism for representing systems with concurrency and resource sharing. They can also be viewed as a simple, graphical modeling language represented as bipartite graphs. The language was defined by Carl Adam Petri with the purpose of describing chemical processes in [16]. Many extensions of Petri nets have been developed, with colored Petri nets being of particular interest for this paper. We consider the reader is familiar with the concepts of Petri nets and colored Petri nets; for details, we refer to [18,19,8].

Petri nets are represented as directed bipartite graphs, with four main components: places, transitions, arcs and tokens. *Places* are represented as circles, and they stand for the "states" of the system. *Transitions* are depicted as rectangles, and they stand for the transition of the system from one state to another. A transition has several pre-places and several post-places that are connected to it by arcs. *Arcs* represent the connection between places and transitions, and have an associated multiplicity, denoting how many elements of the preceding (following) place are consumed (produced). *Tokens* represent the quantities of species denoted by places (be it the number of particles or the concentration of a species).

Definition 1. *[13] A Petri net is a tuple* $N = (P, T, F, f, M_0)$ *where* P *is the finite set of places,* T *is the finite set of transitions,* $F \subseteq P \times T \cup T \times P$ *is the set of arcs,* $f : F \to \mathbb{N}$ *is the arc function assigning multiplicities to each arc, and* $M_0 : P \to \mathbb{N}$ *is the function assigning an initial marking of the network.*

Labeled Petri nets are Petri nets with a labeling of their transitions: $N = (P, T, F, f, L)$ where $L : T \to A$ assigns labels from the set A to each transition in T.

The colored counterpart of Petri nets has additional elements. *Color sets* are associated to places, and they represent data types by means of *colors*. *Variables* can be used to form complex *arc expressions* (the counterpart of simple arc multiplicities), *functions*, and *guards* (conditions associated to transitions, that restrict the fireability of transitions to particular subsets of the colored tokens flowing from pre-places to transitions). Each place p contains a multiset of colored tokens with colors from the color set of p.

We use the following notations from [9,12]: S_{MS} denotes the set of all multisets over a set S; $EXPR_S$ denotes the set of expressions over a set of typed variables S; the type of the values obtained when evaluating an expression $e \in EXPR_S$ is the type of the expression; b is a binding that maps each variable onto a value $b(v)$ which is of the same type as the variable; $t(b)$ is a transition instance with transition $t \in T$ and binding b; $p(c)$ is a place instance with $p \in P$ and $c \in C$; $I_P(p)$ ($I_T(t)$, resp.) denotes the place (transition, resp.) instances of a place $p \in P$ (transition $t \in T$, resp.); I_P (I_T, resp.) denotes the set of all instances of all places $p \in P$ (transitions $t \in T$, resp.); $f(x,y)\langle b \rangle \langle c \rangle$ denotes the number of tokens with color c that are present when evaluating arc expression $f(x,y)$ in binding b and $M(p)\langle c \rangle$ denotes the number of tokens with color c that are present in place p in marking M.

Definition 2. *[10] A colored Petri net is a tuple* $N = (P, T, F, \Sigma, V, C, G, f, M_0)$ *where* P *is the finite set of places,* T *is the finite set of transitions,* $F \subseteq P \times T \cup T \times P$ *is the set of arcs,* Σ *is the set of color sets,* V *is the set of typed variables with types from* Σ*,* $C : P \to \Sigma$ *is the color function assigning a color set to each place,* G *is the guard function,* $f : F \to EXPR_V$ *is the arc function assigning expressions over* V *to each arc such that the type of the arc expression is* $C(p)_{MS}$ *where* $p \in P$ *is the place connected to the arc, and* $M_0 : P \to EXPR_\emptyset$ *is the function assigning an initialization expression with type* $C(p)_{MS}$ *to each place* $p \in P$*.*

Each colored Petri net can be *unfolded* to a behaviorally equivalent standard Petri net representation ([9,12]). We denote by $N^* = (P^*, T^*, F^*, f^*, M_0^*)$ the Petri net obtained by unfolding a colored Petri net $N = (P, T, F, \Sigma, V, C, G, f, M_0)$.

Definition 3. *[12] Given a colored Petri net* $N = (P, T, F, \Sigma, V, C, G, f, M_0)$*, its unfolded Petri net is denoted by* $N^* = (P^*, T^*, F^*, f^*, M_0^*)$*, where:* $P^* = I_P$*;* $T^* = I_T$*;* $F^* = \{(p(c), t(b)) \in P^* \times T^* \mid (f(p,t)\langle b \rangle)\langle c \rangle > 0\} \cup \{(t(b), p(c)) \in T^* \times P^* \mid (f(t,p)\langle b \rangle)\langle c \rangle > 0\}$*;* $f^*(p(c), t(b)) = (f(p,t)\langle b \rangle)\langle c \rangle, \forall (p(c), t(b)) \in F^*$ *and* $f^*(t(b), p(c)) = f(t,p)\langle b \rangle)\langle c \rangle, \forall (t(b), p(c) \in F^*$*;* $M_0^*(p(c)) = M_0(p)\langle c \rangle$*.*

2.2 Petri Nets in Biomodeling

One of the many applications of Petri nets is modeling biological systems. Such systems are bipartite, i.e. they consist of species and the interactions be-

tween them. Some of the interactions are independent, and could fire in parallel, thus biological systems exhibit concurrent behavior. These characteristics make them suitable for modeling within the Petri nets formalism, as first proposed in [17]. Extensions of Petri nets allow modeling and simulation of both stochastic and continuous systems, integrating quantitative and qualitative analysis techniques, see [2].

The species in a biological reaction-based model can be represented as places in the Petri nets framework, and each reaction can be represented as a transition that has all substrates as pre-places, and all products as post-places, with the arc multiplicities given by the corresponding stoichiometric coefficients. For more details about modeling biological systems in the Petri net framework see [11].

The software we used to model our case study within the colored Petri nets framework is Snoopy [20]. We used the related tool Charlie to validate our implementations against some basic properties of the models.

3 Case Study: the Heat Shock Response

In this section, we briefly describe the regulatory mechanism of heat shock response and present a biochemical reactions model of this process, as proposed in [15]. We discuss the behavior of the system and the role of acetylation of one of the main actors driving the response.

3.1 A Molecular Model for the Heat Shock Response

The heat shock response (HSR) is a highly conserved regulatory mechanism among eukaryotes, crucial for the survival of cells under stress conditions. At high temperatures proteins misfold and tend to form large aggregates, with destructive effects on the cell, leading to apoptosis. To counter this, cells produce heat shock proteins (hsp's), whose role is to assist misfolded proteins in their correct refolding.

During the response, heat shock factor (hsf) monomers in inactive state are transported to the nucleus of the cell, where they form trimers, hsf_3, and bind onto the promoter of the DNA heat shock genes (hse), expressing heat shock proteins (hsp). When the number of hsp's is sufficient, they will negatively

regulate the reaction, binding to hsf active trimers and causing them to detach from DNA and dissociate into inactive monomers.

We consider the molecular model of the HSR proposed in [15]. The atomic species considered in the system are $\mathsf{hsf}, \mathsf{hse}, \mathsf{hsp}, \mathsf{prot}$, and mfp. The complex species and their composition are: $\mathsf{hsf}_2 = \{\mathsf{hsf}, \mathsf{hsf}\}$, $\mathsf{hsf}_3 = \{\mathsf{hsf}, \mathsf{hsf}, \mathsf{hsf}\}$, $\mathsf{hsf}_3\colon \mathsf{hse} = \{\mathsf{hsf}, \mathsf{hsf}, \mathsf{hsf}, \mathsf{hse}\}$, $\mathsf{hsp}\colon \mathsf{hsf} = \{\mathsf{hsf}, \mathsf{hsp}\}$, $\mathsf{hsp}\colon \mathsf{mfp} = \{\mathsf{hsp}, \mathsf{mfp}\}$.

The molecular model describing the heat shock response consists of 17 irreversible reactions, listed in Table 1. They cover the trimerization of heat shock factors in two steps, hsf_3 binding to heat shock elements, transcription of DNA and translation of hereby synthesized RNA into heat shock proteins. The negative regulation of the response is modeled with reactions 5-8, and degradation of hsp's is modeled with reaction 9. Protein misfolding and chaperon activity of hsp's are modeled through reactions 10-12.

3.2 A Refinement of the HSR Model

Acetylation of hsf's has a great influence on the heat shock response. A refinement of the model in [15] that considers hsf molecules as either acetylated or non-acetylated has been proposed in [6]. We consider the same refinement of the hsf molecules, but implement it differently, as we take into account the order of molecules in a compound.

More specifically, the atomic species hsf is replaced in the refined model with two of its variants: $\mathsf{hsf}^{(0)}$, a non-acetylated hsf molecule, and $\mathsf{hsf}^{(1)}$, an acetylated hsf molecule. The implicit assumption in [15] is that the order of the molecules in a compound does not matter, what matters is only the number of acetylated sites. We make instead the assumption that the order of molecules matters, a valid assumption since proteins have multiple binding sites and this could introduce ordering. Thus a dimer $\mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}$ is different from $\mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}$, although both dimers have one acetylated site. This small refinement induces an explosion of the model. Our refined model is fully listed in Table 2.

As opposed to the approach in [4], we consider that the number of acetylated sites is conserved by the reactions. For example, a reaction $\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}$ will not appear in our refined model sincs it violates the conservation of acetylated sites constraint. One could think of this as a particular case of [4] where the reactions not present in the model have kinetic constant 0.

4 A Petri Net Model for the Basic HSR Model

We modeled the heat shock response model presented in [15] as a Petri net, following the standard methodology for modeling metabolic systems as Petri nets see [17]. The resulting network can be seen in Figure 2, and its Snoopy implementation is available at [1]. Throughout the paper we denote this network by $H_{bas} = (P, T, F, f_1, M_{0,1})$ for some initial marking $M_{0,1}$.

In order to validate our model, we simulated it with the numerical setup of [15] and checked that the continuous evolution of species concentrations is identical with the one reported in [15], data omitted here due to lack of space. We also checked that the P-invariants of H_{bas} correctly encode the three mass conservation relations of the HSR model, see [15].

5 Colored Petri Nets Hide the Combinatorial State Space Explosion for the Refined HSR Model

In this section we present our colored Petri net model of the refinement of the HSR model, and our modeling choices. One option of modeling the refinement of the HSR model in Table 2 is to use a standard Petri net. But this network will have a transition for each reaction, thus 77 transtions and 29 places, an explosion we avoid through the use of colors.

There are multiple ways of choosing color sets in a colored Petri net model of a biological system. Depending on the choice, additional transitions, guards or complicated functions may have to be introduced in the network, see [5]. For example, hsf dimers could be modeled as a place with an int color set with values {0,1,2} denoting the number of acetylated sites. They could also be modeled with an int color set with values {0,1,2,3} to account for the order of the acetylated sites. The same could be done with a compound color set $\{0, 1\} \times \{0, 1\}$.

We chose to model the hsf molecules as a place with a color set Monomer with values {0,1} denoting whether the molecule is acetylated (1) or not (0). hsf dimers are modeled as a Cartesian product of two hsf's, and hsf trimers are modeled as a Cartesian product of three hsf's. All complex species subject to refinement are modeled as Cartesian products of their atomic components. The atomic and complex species that are not refined have the default color set offered by Snoopy, Dot with a single color dot. This representation is very

compact, and leaves the structure of the network unchanged when going from a standard Petri net representation to a colored Petri net representation.

Our colored Petri net representation of the refined model is presented in Figure 3, and its Snoopy implementation is available at [1]. The structure of the network (places, transitions and the arcs connecting them) is the same as the one in the basic model. For this reason we will use the same sets of places, transitions and arcs in the definition of the refined network. The context will make it clear whether we are talking about the basic network or the refined one. We denote by $H_{ref} = (P, T, F, \Sigma, V, C, G, f_2, M_{0,2})$ our colored Petri net for the refined HSR model. The initial marking $M_{0,2}$ is defined so that Equation (1) holds for all $p \in P$.

$$\sum_{c \in C(p)} M_{0,2}(p)\langle c \rangle = M_{0,1}(p) \ . \tag{1}$$

The entire complexity is encapsulated in the color sets of the places and the arc expressions. To explain the choice of arc expressions, we first give the example of three reactions, and then give a general rule. We consider reactions $\mathsf{hsp} + \mathsf{hsf} \rightarrow \mathsf{hsp{:}\,hsf}$ and $2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$. Their representation as a colored Petri net is shown in Figure 1. Places $\mathsf{p}, \mathsf{q}, \mathsf{r}, \mathsf{s}$ denote $\mathsf{hsf}, \mathsf{hsp}, \mathsf{hsf}_2$ and $\mathsf{hsp{:}\,hsf}$, respectively.

Reaction $\mathsf{hsp} + \mathsf{hsf} \rightarrow \mathsf{hsp{:}\,hsf}$ is refined into two reactions, see Table 2. The arc expression of arc $(\mathsf{q}, \mathsf{t}')$ is dot, meaning a token with color dot is consumed by reaction t'. Arc $(\mathsf{p}, \mathsf{t}')$ has arc expression $\mathsf{v1}$, a variable of type $\mathsf{Monomer}$. The variable can be bound to either value 0 or value 1. In the transition instance where $\mathsf{v1} = 0$, the product of transition t' is $(\mathsf{dot}, 0)$; thus this transition instance models the reaction $\mathsf{hsp} + \mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp{:}\,hsf}^{(0)}$. Similarly, the transition instance where $\mathsf{v1}$ is bound to value 1 models reaction $\mathsf{hsp} + \mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp{:}\,hsf}^{(1)}$.

Reaction $2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$ is refined into four reversible reactions, accounting for all possible combinations of acetylated and non-acetylated hsf's, see Table 2. The forward direction of the reaction is modeled by transition $\mathsf{t\_f}$, and the reverse direction is modeled by transition $\mathsf{t\_b}$ in Figure 1. Variables $\mathsf{v1}, \mathsf{v2}$ can be bound independently to values 0 or 1. The expression $\mathsf{v1} + +\mathsf{v2}$ denotes a multiset with variables $\mathsf{v1}$ and $\mathsf{v2}$. The arc expression $(\mathsf{v1}, \mathsf{v2})$ denotes a tuple of type Dimer, with the particular values of its components given by the values of variables $\mathsf{v1}$ and $\mathsf{v2}$. It is crucial for the components of a compound type to be explicitly referred in arc expressions, in order to satisfy the conservation

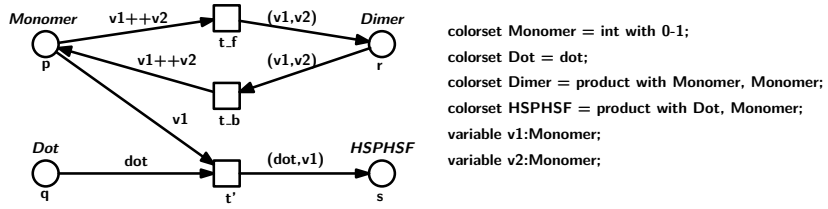


Fig. 1. Example of a colored Petri net. The italic text on top of places represents the color set. The text below places and transitions represents their name, and the text on top of an arc is the arc expression.

of acetylated sites constraint and the ordering of molecules. Another crucial aspect to this end is that the arc expressions of arcs connecting a transition with its post-places use the variables of the arc expressions connecting the pre-places with the transition. Thus, the transition instance of $\mathsf{t\_f}$ with $\mathsf{v1} = 0$ and $\mathsf{v2} = 1$ uniquely represents reaction $\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)} \rightarrow \mathsf{hsf}^{(0)} : \mathsf{hsf}^{(1)}$ because the compound it produces, $(\mathsf{v1}, \mathsf{v2})$, is bound to $(0, 1)$. Similarly, the instance of $\mathsf{t\_b}$ where $\mathsf{v1}, \mathsf{v2}$ are bound to 1 and 0 respectively uniquely represents reaction $\mathsf{hsf}^{(1)} : \mathsf{hsf}^{(0)} \rightarrow \mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)}$ because it produces one $\mathsf{v1}$ and one $\mathsf{v2}$. The bindings of the variables $\mathsf{v1}, \mathsf{v2}$ to values $\{(0,0),(0,1),(1,0),(1,1)\}$ give all the variants that reaction $2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$ is refined to.

As a general rule, the arc expression of an arc $a \in F$ connected to a place p that represents a species that is subject to refinement (hsf or a complex species containing hsf) uses variables. If p represents hsf, then the arc expression uses as many variables with type $C(p)$ as the multiplicity of arc a in H_{bas}: $f_2(a) = v_1 + + \ldots + + v_n$, where $v_i \in V$ with $\mathsf{type}(v_i) = C(p)$ and $n = f_1(a)$. If p denotes a complex species S, then the arc expression uses $f_1(a)$ ordered tuples of variables of types $C(q)$, where q is the place denoting hsf and values for places denoting atomic species that are contained in complex species S, e.g. $(\mathsf{dot}, \mathsf{v1})$ for $\mathsf{hsp}\colon \mathsf{hsf}$ or $(\mathsf{v1}, \mathsf{v2}, \mathsf{v3}, \mathsf{dot})$ for $\mathsf{hsf}_3\colon \mathsf{hse}$. With this construction of H_{ref} as a refinement of H_{bas}, the arc expressions obey the rule $\mid f_2(a) \mid = f_1(a)$ for all arcs $a \in F$, where $\mid f_2(a) \mid$ denotes the cardinality of arc expression $f_2(a)$.

We denote by $H_{unf} = (P^*, T^*, F^*, f_2^*, M_{0,2}^*)$ the standard Petri net obtained by unfolding H_{ref}. H_{unf} contains 29 places and 77 transitions (one for each reaction in Table 2), as opposed to 10 places and 17 transitions for the colored

model. By definition of f_2^* as the equivalent of f_2 in the unfolded network, we have that $\sum_{\substack{q\in I_P(p)\\ q\in {}^\bullet t^*}} f_2^*(q,t^*) =\mid f_2(p,t) \mid,\ \forall t\in T,\ \forall p\in {}^\bullet t,\ \forall t^*\in I_T(t)$, thus

$$\sum_{\substack{q\in I_P(p)\\ q\in {}^\bullet t^*}} f_2^*(q,t^*) = f_1(p,t),\ \forall t\in T,\ \forall p\in {}^\bullet t,\ \forall t^*\in I_T(t)\ . \tag{2}$$

6 Bisimilarity of the two Petri Net Models

Several equivalence criteria have been proposed for Petri nets, e.g. *bisimilarity, language (trace) equivalence*, and *reachability set equality*. In our case we cannot consider reachability set equality, since H_{unf} has a different number of places than H_{bas}. Instead, we will prove that the two are bisimilar. First, we recall the concept of bisimilarity in the context of standard labeled Petri nets, and then we extend the definition to bisimilarity between a labeled standard Petri net and a labeled colored Petri net. Finally, we prove the bisimilarity between our two Petri net models.

Definition 4. *[7] Given two labeled Petri nets* $N_1 = (P_1, T_1, F_1, f_1, M_{0,1}, L_1)$ *and* $N_2 = (P_2, T_2, F_2, f_2, M_{0,2}, L_2)$ *with* $L_1 : T_1 \to A$ *and* $L_2 : T_2 \to A$*, a binary relation* $R \subseteq \mathbb{N}^{P_1} \times \mathbb{N}^{P_2}$ *is a* bisimulation *if for all tuples* $(M_1, M_2) \in R$ *and for each label* $a \in A$*:*

1. *if* $M_1 \xrightarrow{a}_{N_1} M_1'$ *for some* M_1'*, then there is some* M_2' *such that* $M_2 \xrightarrow{a}_{N_2} M_2'$ *and* $(M_1', M_2') \in R$*;*
2. *if* $M_2 \xrightarrow{a}_{N_2} M_2'$ *for some* M_2'*, then there is some* M_1' *such that* $M_1 \xrightarrow{a}_{N_1} M_1'$ *and* $(M_1', M_2') \in R$*.*

Two labeled Petri nets N_1, N_2 *are* bisimilar *if there is a bisimulation relation* R *such that* $(M_{0,1}, M_{0,2}) \in R$*.*

We now introduce a labeling of the unfolded equivalent of a colored Petri net based on its original labeling.

Definition 5. *Consider a colored Petri net* $N = (P, T, F, \Sigma, C, G, f, M_0)$ *and its equivalent unfolded standard Petri net* $N^* = (P^*, T^*, F^*, M_0^*)$*. For any labeling* $L : T \to A$ *of* N *we define the* equivalent labeling of N^* *as the labeling* $L^* : T^* \to A$ *such that for all transitions* $t \in T$ *if* $L(t) = a$*, then all transitions* $t' \in T^*$ *such that* $t' \in I_T(t)$ *have the same label,* $L^*(t') = a$*.*

We next introduce a definition of bisimilarity between a standard and a colored Petri net.

Definition 6. *Given a labeled Petri net* $N_1 = (P_1, T_1, F_1, f_1, M_{0,1}, L_1)$ *and a labeled colored Petri net* $N_2 = (P_2, T_2, F_2, \Sigma, C, G, f, M_{0,2}, L_2)$ *with its corresponding unfolded Petri net with equivalent labelling,* $N_2^* = (P_2^*, T_2^*, F_2^*, f_2^*, M_{0,2}^*, L_2^*)$ *we say that* N_1 *and* N_2 *are* bisimilar *if there is a bisimulation relation* $R \subseteq \mathbb{N}^{P_1} \times \mathbb{N}^{P_2^*}$ *such that* $(M_{0,1}, M_{0,2}^*) \in R$.

We next prove that H_{bas} and H_{ref} are bisimilar. To this end, we label the two networks. Each transition in Figures 2 and 3 has a name written next to it, and moreover the transitions modeling the same reaction have the same name in the two models. We consider as labeling function of the two networks the function L that assigns to each transition its name as a label.

Theorem 1. *The Petri net* H_{bas} *developed for the basic HSR model with labeling* L *and the colored Petri net* H_{ref} *modeling the refined HSR with the same labeling* L *are bisimilar.*

Proof. The proof will use H_{unf}, the unfolded equivalent network of H_{ref}.

We define relation $R \subseteq \mathbb{N}^P \times \mathbb{N}^{P^*}$ such that:

$$(M_1, M_2) \in R \text{ iff } M_1(p) = \sum_{q \in I_P(p)} M_2(q),\ \forall p \in P\ , \tag{3}$$

where M_1 is a marking of H_{bas} and M_2 is a marking of H_{unf}.

We prove now that the first condition for R being a bisimulation relation holds: *for every* $(M_1, M_2) \in R$ *if* $M_1 \xrightarrow{a}_{H_{bas}} M_1'$ *for some* M_1'*, then there exists some* M_2' *such that* $M_2 \xrightarrow{a}_{H_{unf}} M_2'$ *and* $(M_1', M_2') \in R$;

Let t_a denote the transition with label a in H_{bas} (by our labeling L, there is only one such transition). The pre-places $p \in {}^\bullet t_a$ of transition t_a in H_{ref} have $\sum_{q \in I_P(p)} M_2(q)$ elements, or by Equation (3) exactly $M_1(p)$ colored tokens. Since t_a is enabled by M_1 in H_{bas}, it is also enabled in H_{ref} because its pre-places are sufficiently marked and the color of tokens is not important (as we consider all possible combinations of choosing colored tokens to enable a transition in H_{ref}, see Section 5). Let $t_a^* \in I_t(t_a)$ denote a transition that is enabled in H_{unf} by marking M_2.

M_1' and M_2' are computed as the standard update of a marking after firing a transition, as detailed in Equations (4), (5).

$$M_1'(p) = \begin{cases} M_1(p) - f_1(p,t_a) + f_1(t_a,p) & \text{if } p \in {}^\bullet t_a \cap t_a^\bullet \\ M_1(p) - f_1(p,t_a) & \text{if } p \in {}^\bullet t_a \setminus t_a^\bullet \\ M_1(p) + f_1(t_a,p) & \text{if } p \in t_a^\bullet \setminus {}^\bullet t_a \\ M_1(p) & \textit{otherwise} \end{cases} \tag{4}$$

$$M_2'(q) = \begin{cases} M_2(q) - f_2^*(q,t_a^*) + f_2^*(t_a^*,q) & \text{if } q \in {}^\bullet t_a^* \cap t_a^{*\bullet} \\ M_2(q) - f_2^*(q,t_a^*) & \text{if } q \in {}^\bullet t_a^* \setminus t_a^{*\bullet} \\ M_2(q) + f_2^*(t_a^*,q) & \text{if } q \in t_a^{*\bullet} \setminus {}^\bullet t_a^* \\ M_2(q) & \textit{otherwise} \end{cases} \tag{5}$$

Whenever a place $p \in P$ is a pre-(post-)place of t_a, some of its place instances are pre-(post-)places of the transition instance t_a^*. We sum over all place instances corresponding to places in P:

$$\sum_{q \in I_P(p)} M_2'(q) = \begin{cases} \sum\limits_{q \in I_P(p)} M_2(q) - \sum\limits_{\substack{q \in I_P(p) \\ q \in {}^\bullet t_a^*}} f_2^*(q,t_a^*) + \sum\limits_{\substack{q \in I_P(p) \\ q \in t_a^{*\bullet}}} f_2^*(t_a^*,q) & \text{if } p \in {}^\bullet t_a \cap t_a^\bullet \\ \sum\limits_{q \in I_P(p)} M_2(q) - \sum\limits_{\substack{q \in I_P(p) \\ q \in {}^\bullet t_a^*}} f_2^*(q,t_a^*) & \text{if } p \in {}^\bullet t_a \setminus t_a^\bullet \\ \sum\limits_{q \in I_P(p)} M_2(q) + \sum\limits_{\substack{q \in I_P(p) \\ q \in t_a^{*\bullet}}} f_2^*(t_a^*,q) & \text{if } p \in t_a^\bullet \setminus {}^\bullet t_a \\ \sum\limits_{q \in I_P(p)} M_2(q) & \textit{otherwise} \end{cases} \tag{6}$$

We next replace the partial sums in Equation (6) with their counterparts in Equations (2) and (3):

$$\sum_{q \in I_P(p)} M_2'(q) = \begin{cases} M_1(p) - f_1(p,t_a) + f_1(t_a,p) & \text{if } p \in {}^\bullet t_a \cap t_a^\bullet \\ M_1(p) - f_1(p,t_a) & \text{if } p \in {}^\bullet t_a \setminus t_a^\bullet \\ M_1(p) + f_1(t_a,p) & \text{if } p \in t_a^\bullet \setminus {}^\bullet t_a \\ M_1(p) & \textit{otherwise} \end{cases} \tag{7}$$

The right hand side of Equations (4) and (7) is identical, so by definition of relation R (Equation (3)) we conclude $(M_1', M_2') \in R$.

We prove now that the second condition for R being a bisimulation relation holds: *for every* $(M_1, M_1) \in R$, *if* $M_2 \xrightarrow{a}_{H_{unf}} M_2'$ *for some* M_2', *then there exists some* M_1' *such that* $M_1 \xrightarrow{a}_{H_{bas}} M_1'$ *and* $(M_1', M_2') \in R$.

By definition of R, $(M_1, M_2) \in R$ implies that whenever a transition t_a^* with label a is enabled by marking M_2 in H_{unf}, the transition t_a with the same

label a in H_{bas} is enabled, as its pre-places are sufficiently marked according to Equation (2). This is shown in (8).

$$M_1(p) = \sum_{q \in I_P(p)} M_2(q), \forall p \in P \Rightarrow M_1(p) \geq \sum_{\substack{q \in I_P(p) \\ q \in {}^\bullet t_a^*}} M_2(q), \forall p \in {}^\bullet t_a \ . \tag{8}$$

Equations (5) and (4) show how the markings of places change in the two networks when firing transitions t_a^* and t_a, respectively:

We substitute M_1 for its representation in relation to M_2, as given by Equation (3), in Equation (4):

$$M_1'(p) = \begin{cases} \sum_{q \in I_P(p)} M_2(q) - f_1(p, t_a) + f_1(t_a, p) & \text{if } p \in {}^\bullet t_a \cap t_a^\bullet \\ \sum_{q \in I_P(p)} M_2(q) - f_1(p, t_a) & \text{if } p \in {}^\bullet t_a \setminus t_a^\bullet \\ \sum_{q \in I_P(p)} M_2(q) + f_1(t_a, p) & \text{if } p \in t_a^\bullet \setminus {}^\bullet t_a \\ \sum_{q \in I_P(p)} M_2(q) & \textit{otherwise} \end{cases} \tag{9}$$

We next substitute Equation (2), where we consider t_a^* as the transition instance for the summation, in Equation (9):

$$M_1'(p) = \begin{cases} \sum\limits_{q \in I_P(p)} M_2(q) - \sum\limits_{\substack{q \in I_P(p) \\ q \in {}^\bullet t_a^*}} f_2^*(q, t_a^*) + \sum\limits_{\substack{q \in I_P(p) \\ q \in t_a^{*\bullet}}} f_2^*(t_a^*, q) & \text{if } p \in {}^\bullet t_a \cap t_a^\bullet \\ \sum\limits_{q \in I_P(p)} M_2(q) - \sum\limits_{\substack{q \in I_P(p) \\ q \in {}^\bullet t_a^*}} f_2^*(q, t_a^*) & \text{if } p \in {}^\bullet t_a \setminus t_a^\bullet \\ \sum\limits_{q \in I_P(p)} M_2(q) + \sum\limits_{\substack{q \in I_P(p) \\ q \in t_a^{*\bullet}}} f_2^*(t_a^*, q) & \text{if } p \in t_a^\bullet \setminus {}^\bullet t_a \\ \sum\limits_{q \in I_P(p)} M_2(q) & \textit{otherwise} \end{cases} \tag{10}$$

A place p is connected to t_a iff at least one of its instances is connected to t_a^*. We sum the markings M_1' in Equation (5) over all instances of places $p \in P$:

$$\sum_{q\in I_P(p)} M_2'(q) = \begin{cases} \sum\limits_{q\in I_P(p)} M_2(q) - \sum\limits_{\substack{q\in I_P(p)\\ q\in {}^\bullet t_a^*}} f_2^*(q,t_a^*) + \sum\limits_{\substack{q\in I_P(p)\\ q\in t_a^{*\bullet}}} f_2^*(t_a^*,q) & \text{if } p\in {}^\bullet t_a \cap t_a^\bullet \\ \sum\limits_{q\in I_P(p)} M_2(q) - \sum\limits_{\substack{q\in I_P(p)\\ q\in {}^\bullet t_a^*}} f_2^*(q,t_a^*) & \text{if } p\in {}^\bullet t_a \setminus t_a^\bullet \\ \sum\limits_{q\in I_P(p)} M_2(q) + \sum\limits_{\substack{q\in I_P(p)\\ q\in t_a^{*\bullet}}} f_2^*(t_a^*,q) & \text{if } p\in t_a^\bullet \setminus {}^\bullet t_a \\ \sum\limits_{q\in I_P(p)} M_2(q) & \textit{otherwise} \end{cases} \tag{11}$$

The right hand side of equations (10) and (11) is identical, so we can conclude that $(M_1', M_2') \in R$.

Relation R satisfies both conditions for being a bisimulation relation. By Equation (1) we have that $(M_{0,1}, M_{0,2}) \in R$. In conclusion, H_{bas} and H_{ref} are bisimilar.

7 Conclusions

We have developed two models for the heat shock response, using Petri nets and their colored extension as modeling frameworks. The first model contains 10 places and 17 transitions, corresponding to 10 species and 17 reactions, as in [15]. The second model contains the same number of places and transitions, but these stand for 29 species and 77 reactions modeling the refinement of the heat shock response that accounts for the acetylation of one of the main actors of the response. The complexity is hidden in the colors that each token in a place may have, but the representation is very compact (an important aspect when modeling large systems).

We introduced a notion of bisimilarity between a standard and a colored Petri net, and we proved that the two networks we have built are bisimilar. The construction of H_{ref} was done in a systematic way that makes it possible to generalize the method, and this is in the scope of a future paper.

References

1. Petri net implementation of the heat shock response. `http://combio.abo.fi/research/computational-modeling-of-the-eukaryotic-heat-shock-response/petri-net-implementation-of-the-heat-shock-response/`.
2. P. Baldan, N. Cocco, A. Marin, and M. Simeoni. Petri nets for modelling metabolic pathways: a survey. *Natural Computing*, 9(4):955–989, 2010.
3. V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule-based modelling and model perturbation. *Transactions on Computational Systems Biology XI*, pages 116–137, 2009.
4. C. Gratie and I. Petre. Fit-preserving data refinement of mass-action reaction networks. In A. Beckmann, E. Csuhaj-Varjú, and K. Meer, editors, *Language, Life, Limits*, volume 8493 of *Lecture Notes in Computer Science*, pages 204–213. Springer, 2014.
5. D.-E. Gratie, B. Iancu, S. Azimi, and I. Petre. Quantitative model refinement in four different frameworks, with applications to the heat shock response, 2013.
6. B. Iancu, E. Czeizler, E. Czeizler, and I. Petre. Quantitative refinement of reaction models. *International Journal of Unconventional Computing*, 8(5-6):529–550, 2012.
7. P. Jančar. *Decidability questions for bisimilarity of Petri nets and some related problems*. Springer, 1994.
8. K. Jensen. Coloured petri nets: A high level language for system design and analysis. *Advances in Petri nets 1990*, pages 342–416, 1991.
9. K. Jensen. Coloured petri nets. volume 1, basic concepts. *EATCS Monographs in Theoretical Computer Science*, 1, 1992.
10. K. Jensen and L. M. Kristensen. *Coloured Petri nets: modelling and validation of concurrent systems*. Springer, 2009.
11. I. Koch, W. Reisig, and F. Schreiber. *Modeling in systems biology: the Petri Net approach*, volume 16. Springer, 2010.
12. F. Liu. *Colored Petri nets for systems biology*. PhD thesis, Universitätsbibliothek, 2012.
13. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
14. E. Murphy, V. Danos, J. Feret, J. Krivine, and R. Harmer. *Elements of Computational Systems Biology*, chapter Rule Based Modelling and Model Refinement, pages 83–114. Wiley Book Series on Bioinformatics. John Wiley & Sons, Inc., 2010.
15. I. Petre, A. Mizera, C. Hyder, A. Meinander, A. Mikhailov, R. Morimoto, L. Sistonen, J. Eriksson, and R. Back. A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. *Natural Computing*, 10(1):595–612, 2011.
16. C. A. Petri. *Kommunikation mit Automaten*. PhD thesis, Bonn: Institut für Instrumentelle Mathematik, 1962.
17. V. N. Reddy, M. L. Mavrovouniotis, M. N. Liebman, et al. Petri net representations in metabolic pathways. In *ISMB. International Conference on Intelligent Systems for Molecular Biology*, volume 93, pages 328–336, 1993.
18. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*, volume 1491 of *Lecture Notes in Computer Science*. Springer, 1998.
19. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets II: Applications, Advances in Petri Nets*, volume 1492 of *Lecture Notes in Computer Science*. Springer, 1998.

20. C. Rohr, W. Marwan, and M. Heiner. Snoopy - a unifying petri net framework to investigate biomolecular networks. *Bioinformatics 2010*, pages 974–975, 2010.

A The Basic HSR Molecular Model

Table 1. The molecular model for the eukaryotic heat shock response proposed in [15].

1. $2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$
2. $\mathsf{hsf} + \mathsf{hsf}_2 \rightleftarrows \mathsf{hsf}_3$
3. $\mathsf{hsf}_3 + \mathsf{hse} \rightleftarrows \mathsf{hsf}_3\colon \mathsf{hse}$
4. $\mathsf{hsf}_3\colon \mathsf{hse} \to \mathsf{hsf}_3\colon \mathsf{hse} + \mathsf{hsp}$
5. $\mathsf{hsp} + \mathsf{hsf} \rightleftarrows \mathsf{hsp}\colon \mathsf{hsf}$
6. $\mathsf{hsp} + \mathsf{hsf}_2 \to \mathsf{hsp}\colon \mathsf{hsf} + \mathsf{hsf}$
7. $\mathsf{hsp} + \mathsf{hsf}_3 \to \mathsf{hsp}\colon \mathsf{hsf} + 2\,\mathsf{hsf}$
8. $\mathsf{hsp} + \mathsf{hsf}_3\colon \mathsf{hse} \to \mathsf{hsp}\colon \mathsf{hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$
9. $\mathsf{hsp} \to \emptyset$
10. $\mathsf{prot} \to \mathsf{mfp}$
11. $\mathsf{hsp} + \mathsf{mfp} \rightleftarrows \mathsf{hsp}\colon \mathsf{mfp}$
12. $\mathsf{hsp}\colon \mathsf{mfp} \to \mathsf{hsp} + \mathsf{prot}$

B The Refined HSR Molecular Model

Table 2: The refinement of the molecular model proposed in [15]

Reaction in the basic model	Reactions in the refined model
$2\,\mathsf{hsf} \rightleftarrows \mathsf{hsf}_2$	$\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}$ $\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}$ $\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}$ $\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}$
$\mathsf{hsf} + \mathsf{hsf}_2 \rightleftarrows \mathsf{hsf}_3$	$\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}$ $\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}$ $\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}$ $\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}$ $\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}$ $\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}$ $\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}$ $\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}$
$\mathsf{hsf}_3 + \mathsf{hse} \rightleftarrows \mathsf{hsf}_3\colon \mathsf{hse}$	$\mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hse}$ $\mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hse}$ $\mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hse}$ $\mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(1)}\colon \mathsf{hse}$ $\mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(1)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hsf}^{(0)}\colon \mathsf{hse}$

Table 2: The refinement of the model proposed in [15] - Continued

	$\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse}$ $\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse}$ $\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)} + \mathsf{hse} \rightleftarrows \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse}$
$\mathsf{hsf}_3{:}\,\mathsf{hse} \rightarrow$ $\mathsf{hsf}_3{:}\,\mathsf{hse} + \mathsf{hsp}$	$\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} + \mathsf{hsp}$ $\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} + \mathsf{hsp}$
$\mathsf{hsp} + \mathsf{hsf} \rightleftarrows \mathsf{hsp}{:}\,\mathsf{hsf}$	$\mathsf{hsp} + \mathsf{hsf}^{(0)} \rightleftarrows \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)} \rightleftarrows \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)}$
$\mathsf{hsp} + \mathsf{hsf}_2 \rightarrow$ $\mathsf{hsp}{:}\,\mathsf{hsf} + \mathsf{hsf}$	$\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(1)}$
$\mathsf{hsp} + \mathsf{hsf}_3 \rightarrow$ $\mathsf{hsp}{:}\,\mathsf{hsf} + 2\,\mathsf{hsf}$	$\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + 2\,\mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + 2\,\mathsf{hsf}^{(1)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + 2\,\mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + 2\,\mathsf{hsf}^{(1)}$
$\mathsf{hsp} + \mathsf{hsf}_3{:}\,\mathsf{hse} \rightarrow$ $\mathsf{hsp}{:}\,\mathsf{hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$	$\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + 2\,\mathsf{hsf}^{(0)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(0)} + 2\,\mathsf{hsf}^{(1)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + 2\,\mathsf{hsf}^{(0)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)} + \mathsf{hsf}^{(1)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(0)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + \mathsf{hsf}^{(1)} + \mathsf{hsf}^{(0)} + \mathsf{hse}$ $\mathsf{hsp} + \mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hsf}^{(1)}{:}\mathsf{hse} \rightarrow \mathsf{hsp}{:}\,\mathsf{hsf}^{(1)} + 2\,\mathsf{hsf}^{(1)} + \mathsf{hse}$
$\mathsf{hsp} \rightarrow \emptyset$	$\mathsf{hsp} \rightarrow \emptyset$
$\mathsf{prot} \rightarrow \mathsf{mfp}$	$\mathsf{prot} \rightarrow \mathsf{mfp}$
$\mathsf{hsp} + \mathsf{mfp} \rightleftarrows \mathsf{hsp}{:}\,\mathsf{mfp}$	$\mathsf{hsp} + \mathsf{mfp} \rightleftarrows \mathsf{hsp}{:}\,\mathsf{mfp}$
$\mathsf{hsp}{:}\,\mathsf{mfp} \rightarrow \mathsf{hsp} + \mathsf{prot}$	$\mathsf{hsp}{:}\,\mathsf{mfp} \rightarrow \mathsf{hsp} + \mathsf{prot}$

C Petri Net for the Basic HSR Model

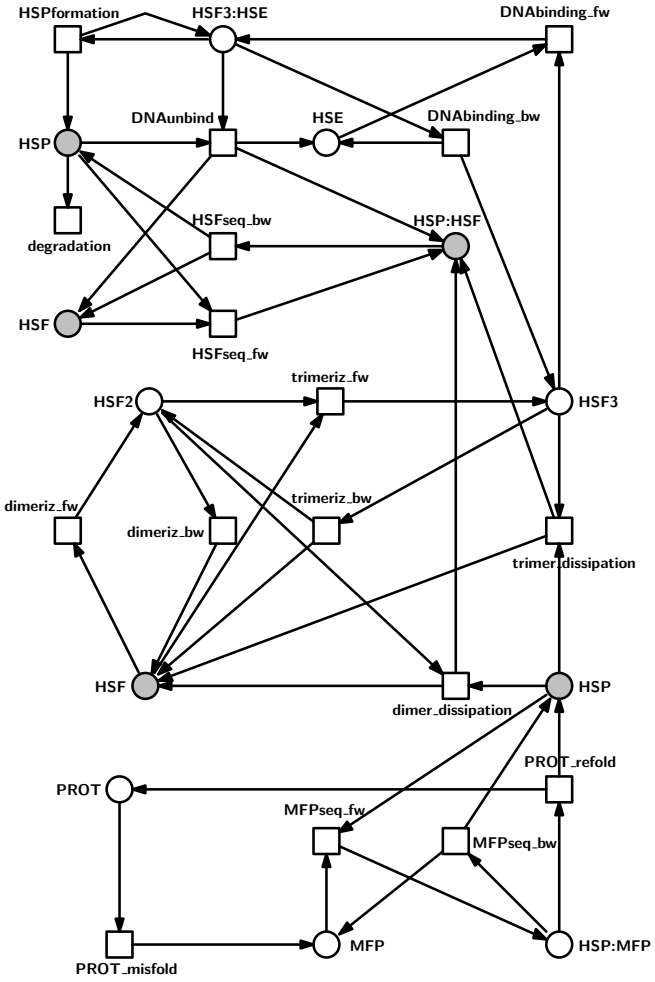


Fig. 2. Snoopy representation of the basic heat shock response model

D Colored Petri Net for the Refined HSR Model

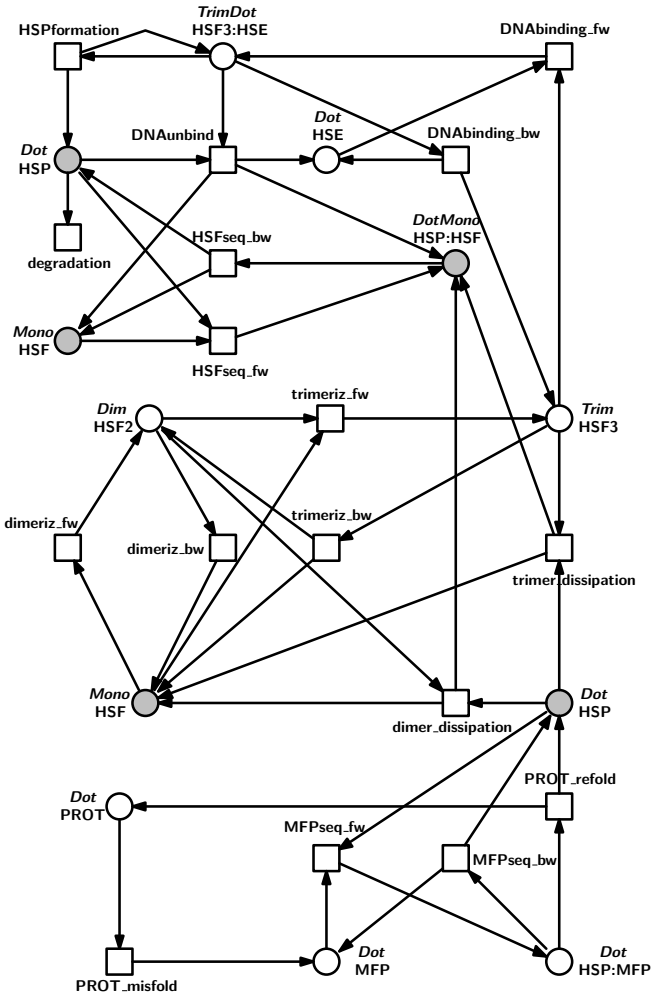


Fig. 3. Snoopy representation of the refined heat shock response model. The network is similar to the basic model network. We include here the information about each place's color set (brown text next to each place, above the name of the place), and we omit all arc expressions, for readability reasons.

Paper IV

Full structural model refinement as type refinement of colored Petri nets

Diana-Elena Gratie and Ion Petre

Originally published in Monika Heiner and Annegret K. Wagler (Eds.) *Proceedings of the 6th International Workshop on Biological Processes and Petri Nets*. Vol. 1373 of Ceur Workshop Proceedings, pp. 70–84. CEUR Workshop Proceedings, 2015. The publication is available at CEUR via `http://ceur-ws.org/Vol-1373/paper5.pdf`.

Full structural model refinement as type refinement of colored Petri nets

Diana-Elena Gratie[1,2,3] and Ion Petre[1,2,3]
{dgratie,ipetre}@abo.fi

[1] Turku Centre for Computer Science
[2] Computational Biomodeling Laboratory
[3] Department of Computer Science, Åbo Akademi University
Joukahaisenkatu 3-5, FIN-20520 Turku

Abstract. In this paper we propose a method for implementing a full structural model refinement of a (biological) model represented as a (colored) Petri net. We build on the full structural data refinement definition of C. Gratie and Petre, and the type refinement of colored Petri nets introduced by Charles Lakos. Given a (biological) reaction-based model and a desired full structural refinement of it, we propose a general coloring scheme for a colored Petri net implementation of the model and give an algorithm for adding the refinement details in the Petri net model. We then prove that the construction is a type refinement, and that by our choice of color sets the resulting refined colored Petri net implements the full structural refinement of the given model.

Keywords: Colored Petri nets, type refinement, reaction network, structural model refinement.

1 Introduction

Model refinement, the process of adding more details to an existing model, is an important step in the model building cycle. Many refinement methods have been proposed for different modeling frameworks and formalisms, e.g., action systems [1], Petri nets [17, 11], kappa [4], biochemical reaction networs [7], π-calculus [16], etc. We bridge here two modelling frameworks and their respective ways of implementing refinement, namely reaction network models with structural refinement and colored Petri nets with type refinement.

Type refinement of colored Petri nets has been introduced in [11], and consists of refining the color sets of places such that the new color sets are polymorphic with the initial color sets. The authors see this as adding some supplementary data to a given data type represented as a color set, e.g. include in the entry of a book in a library not only its title and authors, but also the maximum number of days it can be borrowed.

The concept of *(full) structural refinement* of a reaction network (bio-)model has been introduced in [7] (where it was called data refinement), with a focus on

M Heiner, AK Wagler (Eds.): BioPPN 2015, a satellite event of PETRI NETS 2015,
CEUR Workshop Proceedings Vol. 1373, 2015.

an ODE-based representation of a model and its refinement. A sufficient condition for the refined model to preserve the fit of the original one was discussed in [6] for mass-action models. We follow in this paper the terminology of [6]. We use the main concepts of *species refinement* and *(full) structural refinement* for models represented as (colored) Petri nets, and give a methodology for implementing full structural refinements as type refinements of colored Petri nets. An approach to implementing model refinement in the colored Petri net framework has been exemplified for a model of the eukaryotic heat shock response mechanism in [8]. The authors present there two coloring schemes that can be used for the particular refinement they were implementing. We derive here a general coloring scheme for model refinement that can be used when implementing a full structural data refinement of a model.

We assume the reader is familiar with (colored) Petri nets, but we recall some of the basic definitions so that the paper is self-contained.

The paper is structured as follows: in Section 2 we present reaction network (also called reaction-based) models and the notions of *species refinement* and *(full) structural refinement* of such models, with a discussion on the explosion of the model induced by a refinement, in terms of number of species and reactions that the initial model refines to. In Section 3 we recall some notions and notations for Petri nets and their colored version, give a coloring scheme and discuss how a reaction network model can be implemented as a (colored) Petri net. We continue in Section 4 with proposing a type refinement based on a refinement relation ρ and prove that the chosen type refinement results in a colored Petri net that is the implementation of the full structural $\rho-$refinement of the initial model. We draw our conclusions and discuss about the model size and successive refinements in Section 5.

2 Model Refinement

In systems biology, model refinement comprises two aspects: the structural side and the quantitative side. The structural side handles the newly introduced species and presents a methodology for computing the new set of reactions, while the quantitative side deals with changes in the kinetic constants of the model and ways of setting the new parameters in such a way that previous data is used. *Quantitative model refinement* was introduced in [15, 4] for rule-based models, and for reaction-based models in [13, 7]. We recall here the structural refinement of reaction network models, as presented in [7] and based on the terminology of [6]. We are only interested in the structural refinement, so we will not focus on any quantitative details.

A *reaction-based model* M consists of a finite set of *species* $\mathscr{S} = \{A_1, \ldots, A_m\}$ and a finite set of *reactions* $\mathscr{R} = \{r_1, \ldots, r_n\}$ using only species in $\mathscr{S}$. A reaction $r_j \in \mathscr{R}$ can be formulated as a rewriting rule of the form:

$$r_j \ : \ c_{1,j}A_1 + \ldots + c_{m,j}A_m \xrightarrow{k_{r_j}} c'_{1,j}A_1 + \ldots + c'_{m,j}A_m, \tag{1}$$

with the meaning that $c_{i,j}$ copies of species A_i are consumed by the reaction and $c'_{i,j}$ copies of species A_i are produced, $i = 1..m$. Constants $c_{1,j}, \ldots, c_{m,j}, c'_{1,j}, \ldots, c'_{m,j} \in \mathbb{N}$ are the *stoichiometric coefficients* of r_j and $k_{r_j} \geq 0$ is the *kinetic rate constant* of reaction r_j. We denote by $\mathbf{r_j^-} = (c_{1,j}, \ldots, c_{m,j})$ the vector of stoichiometric coefficients on the left hand side of the reaction, for the species being consumed in reaction r_j, and by $\mathbf{r_j^+} = (c'_{1,j}, \ldots, c'_{m,j})$ the vector of stoichiometric coefficients on its right hand side, those of species being produced. Without a risk of ambiguity, reaction r_j can then be written as $\mathbf{r_j^-} \xrightarrow{k_{r_j}} \mathbf{r_j^+}$.

Example 1. A biological system with two irreversible reactions that encode the dimerization of a molecule P can be represented as a reaction-based model $M = (\mathscr{S}, \mathscr{R})$ where $\mathscr{S} = \{P, P_2\}$ and $\mathscr{R} = \{2P \to P_2,\ P_2 \to 2P\}$. P represents the monomeric molecule and P_2 is the dimer that is formed from two P monomers.

Data refinement is the type of refinement of a model that consists in adding details related to the species of the model, i.e., it replaces a species with several of its subspecies. The subspecies may account for post-translational modifications of macromolecules, or distinguish between possible variants of some trait.

All species are considered to be refined at once, thus each species in an initial model is replaced by a non-empty set of refined species to yield a refined model, as dictated by a *species refinement relation* ρ. This is formalized in Definition 1.

Definition 1 ([6]). *Given two sets of species $\mathscr{S}$ and $\mathscr{S}'$, and a relation $\rho \subseteq \mathscr{S} \times \mathscr{S}'$, we say that ρ is a* species refinement relation *iff it satisfies the following conditions:*

1. *for each $A \in \mathscr{S}$ there exists $A' \in \mathscr{S}'$ such that $(A, A') \in \rho$;*
2. *for each $A' \in \mathscr{S}'$ there exists* exactly one *$A \in \mathscr{S}$ such that $(A, A') \in \rho$.*

We denote $\rho(A) = \{A' \in \mathscr{S}' \mid (A, A') \in \rho\}$. We say that all species $A' \in \rho(A)$ are siblings.

Intuitively, each species $A \in \mathscr{S}$ is replaced in the refined model with the set of species $\rho(A)$. For the case where $\rho(A)$ is a singleton set, one may consider that species A does not change, even if its refined counterpart is denoted by a different name in $\mathscr{S}'$; such a refinement of a species is called *trivial*.

Next we recall the definitions of refinement of a vector (of stoichiometric coefficients), of a reaction, and of a reaction-based model.

Definition 2 ([6]). *Let $\mathscr{S} = \{A_1, \ldots, A_m\}$ and $\mathscr{S}' = \{A'_1, \ldots, A'_p\}$ be two sets of species, and $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ a species refinement relation.*

1. *Let $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{N}^{\mathscr{S}}$ and $\alpha' = (\alpha'_1, \ldots, \alpha'_p) \in \mathbb{N}^{\mathscr{S}'}$. We say that α' is a* ρ-refinement *of α if*

$$\sum_{\substack{1 \leq j \leq p \\ A'_j \in \rho(A_i)}} \alpha'_j = \alpha_i, \text{ for all } 1 \leq i \leq m\ .$$

We denote by $\rho(\alpha)$ the set of all $\rho-$refinements of α.

2. *Let* $r:\ r^- \to r^+$ *and* $r':\ r'^- \to r'^+$ *be two reactions over* $\mathscr{S}$ *and* $\mathscr{S}'$*, resp. We say that* r' *is a* ρ-refinement *of* r *if*

$$r'^- \in \rho(r^-) \text{ and } r'^+ \in \rho(r^+)\ .$$

We denote by $\rho(r)$ *the set of all* $\rho-$*refinements of* r*. Note that* $\rho(r) = \rho(r^-) \times \rho(r^+)$.

3. *Let* $M = (\mathscr{S}, \mathscr{R})$ *and* $M' = (\mathscr{S}', \mathscr{R}')$ *be two reaction-based models, and* $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ *a species refinement relation. We say that* M' *is a* ρ-structural refinement of M *if*

$$\mathscr{R}' \subseteq \bigcup_{r \in \mathscr{R}} \rho(r) \text{ and } \rho(r) \cap \mathscr{R}' \neq \varnothing\ \forall r \in \mathscr{R}\ .$$

In case $\mathscr{R}' = \bigcup_{r \in \mathscr{R}} \rho(r)$*, we say* M' *is the* full structural ρ-refinement *of* M*, denoted* $M' = M_\rho$.

Model explosion. Note that a vector of coefficients $\alpha' \in \mathbb{N}^{\mathscr{S}}$ that respects the sum condition $\sum_{\substack{1 \le j \le p \\ A'_j \in \rho(A_i)}} \alpha'_j = \alpha_i$, for all $1 \le i \le m$ can be seen as a way of choosing α_i elements from a bag containing elements of $|\rho(A_i)|$ types, where the selection may contain several elements of the same type. The total number of different ways in which one may choose k elements from a bag with elements of n types (assuming enough copies of each type are available) is $\left(\!\binom{n}{k}\!\right) = \binom{n+k-1}{k}$, the so-called *multiset coefficient, n multichoose k*.

A reaction r_j of the form (1) can refine to $\prod_{1 \le i \le m} \left(\!\binom{|\rho(A_i)|}{c_{i,j}}\!\right) \cdot \left(\!\binom{|\rho(A_i)|}{c'_{i,j}}\!\right)$ different reactions. The number stems from the number of possible ways of choosing $c_{i,j}$ ($c'_{i,j}$, resp.) copies from the possible refinements of a species $A_i \in \mathscr{S}$. The number of reactions in a full structural $\rho-$refinement of a model with n reactions is thus:

$$\sum_{1 \le j \le n} \prod_{1 \le i \le m} \left(\!\binom{|\rho(A_i)|}{c_{i,j}}\!\right) \cdot \left(\!\binom{|\rho(A_i)|}{c'_{i,j}}\!\right)\ .$$

Example 2. Consider the reaction-based model $M = (\mathscr{S}, \mathscr{R})$ from Example 1. One possible refinement for this model is to consider that molecule P can be in two states: acetylated ($P^{(1)}$) and non-acetylated($P^{(0)}$). Then the dimer P_2 could have none ($P_2^{(0)}$), one ($P_2^{(1)}$) or both ($P_2^{(2)}$) of its composing monomers acetylated. Consider a set of species $\mathscr{S}' = \{P^{(0)}, P^{(1)}, P_2^{(0)}, P_2^{(1)}, P_2^{(2)}\}$. A relation $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ that would capture such a refinement is $\rho = \{(P, P^{(0)}), (P, P^{(1)}), (P_2, P_2^{(0)}), (P_2, P_2^{(1)}), (P_2, P_2^{(2)})\}$. One can easily see that ρ is a refinement relation, based on Definition 1.

A full structural ρ-refinement of M is the model $M' = (\mathscr{S}', \mathscr{R}')$, where $\mathscr{R}' =$

$$\begin{array}{lll}
\{2P^{(0)} \to P_2^{(0)}, & 2P^{(0)} \to P_2^{(1)}, & 2P^{(0)} \to P_2^{(2)}, \\
2P^{(1)} \to P_2^{(0)}, & 2P^{(1)} \to P_2^{(1)}, & 2P^{(1)} \to P_2^{(2)}, \\
P^{(0)} + P^{(1)} \to P_2^{(0)}, & P^{(0)} + P^{(1)} \to P_2^{(1)}, & P^{(0)} + P^{(1)} \to P_2^{(2)}, \\
P_2^{(0)} \to 2P^{(0)}, & P_2^{(0)} \to 2P^{(1)}, & P_2^{(0)} \to P^{(0)} + P^{(1)}, \\
P_2^{(1)} \to 2P^{(0)}, & P_2^{(1)} \to 2P^{(1)}, & P_2^{(1)} \to P^{(0)} + P^{(1)}, \\
P_2^{(2)} \to 2P^{(0)}, & P_2^{(2)} \to 2P^{(1)}, & P_2^{(2)} \to P^{(0)} + P^{(1)}\}.
\end{array}$$

3 Modeling Biological Systems as (Colored) Petri Nets

Many biological models are implemented as Petri nets due to the graphical, intuitive formalism, and the many simulation strategies they offer. We start our discussion over refinement and implementations of models as Petri nets from the standard version of Petri nets. We then continue with colored Petri nets.

3.1 Preliminaries

We assume the reader is familiar with the basic notions and notations related to Petri nets and we refer to [5], [14] for details. We also assume that the reader is familiar with constructing a standard Petri net associated to a reaction-based model; we refer to [2] for details.

In order to implement a reaction-based model as a Petri net, one represents each species via a place, and each reaction via a transition having as pre-places the places representing the reactants of the reaction, and as post-places the places representing the products of the reaction, with each arc expression being the stoichiometry of the represented species in that reaction, see [2].

Definition 3 (Implementation of a reaction network model as a Petri net). *Given a reaction-based model $M = (\mathscr{S}, \mathscr{R})$, and a Petri net $N = (P, T, A, f, M_0)$ with $|\mathscr{S}| = |P|$ and $|\mathscr{R}| = |T|$, we say that the Petri net N* structurally implements *model M if there exists a bijection $\delta : \mathscr{S} \cup \mathscr{R} \to P \cup T$ mapping species of M into places of N and reactions of M into transitions of N ($\delta(x) \in P$, for all $x \in \mathscr{S}$ and $\delta(x) \in T$ for all $x \in \mathscr{R}$) such that for every reaction $r_j \in \mathscr{R}$ and its corresponding transition $t = \delta(r_j)$ and for every species $S_i \in \mathscr{S}$ the following conditions hold:*

1. *if $c_{i,j} > 0$ then $(\delta(S_i), t) \in A$ and $f(\delta(S_i), t) = c_{i,j}$, otherwise $(\delta(S_i), t) \notin A$;*
2. *if $c'_{i,j} > 0$ then $(t, \delta(S_i)) \in A$ and $f(t, \delta(S_i)) = c'_{i,j}$, otherwise $(t, \delta(S_i)) \notin A$.*

Example 3. An example of a Petri net structural implementation of the model described in Example 1 is given in Figure 1. The bijection δ is defined such that $\delta(P) = \mathsf{P\_}$, $\delta(P_2) = \mathsf{P\_2}$, $\delta(2P \to P_2) = \mathsf{T\_fw}$, $\delta(P_2 \to 2P) = \mathsf{T\_bw}$. One can easily see that the arc multiplicities respect the two conditions in Definition 3.

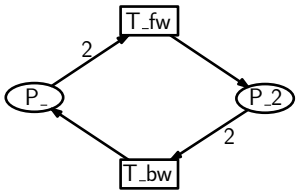


Fig. 1. Standard Petri net structural implementation of a dimerization model (only multiplicities greater than 1 are displayed)

There exist two ways of defining colored Petri nets, one proposed by Kurt Jensen in [9], and an equivalent one adapted from the first definition, by Charles Lakos in [11]. In this paper we consider the definition of colored Petri nets proposed by Lakos because it does not explicitly include transition guards (that we are not using in our construction) and because of the definition of type refinement of colored Petri nets proposed in [11]. We use the following less well known notations: Σ denotes a universe of non-empty color sets with an associated partial order $<:\subseteq \Sigma \times \Sigma$ indicating that values from one color set X with $X <: Y$ can be used in contexts expecting values of Y. Π_Y is a projection function mapping values of X into values of Y. $\Phi\Sigma = \{X \to Y \mid X, Y \in \Sigma\}$ denotes the functions over Σ, and $\mu\mathbf{X} = \{X \to \mathbb{N}\}$ denotes the multisets over X. $E^-, E^+ : \mathbb{Y} \to \mathbb{M}$ represent the *incremental negative and positive, resp. changes* of the occurrence of a step Y, and are given by the linear extension of: $E^-((t,c)) = \sum_{p \in P}\{p\} \times E((p,t))(c)$ and $E^+((t,c)) = \sum_{p \in P}\{p\} \times E((t,p))(c)$, $\forall t \in T,\ \forall c \in C(t)$.

Definition 4 ([11]). *A* colored Petri net *is a tuple* $N = (P, T, A, C, E, \Sigma, \mathbb{M}, \mathbb{Y}, M_0)$ *where:*

- P *is the finite set of places;*
- T *is the finite set of transitions, such that* $P \cap T = \varnothing$*;*
- $A \subseteq P \times T \cup T \times P$ *is the finite set of arcs;*
- Σ *is a universe of non-empty color sets with an associated partial order;*
- $C : P \cup T \to \Sigma$ *is the color set function, assigning color sets to places and (modes) of transitions;*
- $E : A \to \Phi\Sigma$ *is the arc expression function, where* $E(p,t), E(t,p) : C(t) \to \mu C(p)$*;*
- $\mathbb{M} = \mu\{(p,c) \mid p \in P, c \in C(p)\}$ *is the set of markings;*
- $\mathbb{Y} = \mu\{(t,c) \mid t \in T, c \in C(t)\}$ *is the set of steps;*
- M_0 *the initial marking, with* $M_0 \in \mathbb{M}$*.*

Arc expressions may contain variables, which are seen as symbols whose value is determined by the color (mode) of the transition the arc is connected with.

For any colored Petri net with finite color sets there exists a standard Petri net that is behaviorally equivalent, see [10]. The process of transforming a colored Petri net into its standard Petri net equivalent is called *unfolding*. We give in the following the definition of the unfolding of a colored Petri net as adapted from [10] to the notations we use.

Definition 5 ([10]). *Given a colored Petri net* $N = (P, T, A, \Sigma, C, E, \mathbb{M}, \mathbb{Y}, M_0)$*, its unfolded Petri net is denoted by* $N^* = (P^*, T^*, A^*, f^*, M_0^*)$*, where:*

- P^* *is the set of place instances, pairs* (p,c) *with* $p \in P$ *and* $c \in C(p)$*;*
- T^* *is the set of transition instances, pairs* (t,c) *with* $t \in T$ *and* $c \in C(t)$*;*
- $A^* = \{((p,c),(t,c')) \in P^* \times T^* \mid E((p,t))(c')(c) > 0\} \cup \{((t,c'),(p,c)) \in T^* \times P^* \mid E((t,p))(c')(c) > 0\}$*;*
- $f^*((p,c),(t,c')) = E((p,t))(c')(c),\ \forall((p,c),(t,c')) \in A^*$ *and* $f^*((t,c'),(p,c)) = E((t,p))(c')(c),\ \forall((t,c'),(p,c)) \in A^*$*;*
- $M_0^*((p,c)) = M_0(p,c)$*.*

3.2 Coloring a Standard Petri Net

A colored Petri net representation of a model can be obtained from a standard Petri net implementation of the model by assigning to each place a color set with just one element. We propose here a general coloring scheme that uses record color sets (i.e. a data structure containing a finite collection of fields, each with a name and an associated data type) and can easily be extended to incorporate refinement details by adding new fields. Each place is assigned its own record color set with one field that has exactly one value. Each transition is assigned a color set that is a multiset of color sets of its pre- and post-places, where the multiplicity of each color set is given by the multiplicity of the arc connecting the place and the transition. It is basically a multiset with elements of different types. For example, the color set $\mathsf{CS\_T\_fw}$ in Figure 2 is a collection of two elements of type $\mathsf{CS\_P}$ and one element of type $\mathsf{CS\_P2}$. Note that this is not the only possible coloring scheme and moreover it may not be optimal (in terms of number of variables and data structures used), but it is general. One may use integers, records, sets, Cartesian products, or whatever coloring scheme better suits the system being modeled.

A further change that is required when turning a standard Petri net into a colored one is assigning to each arc a with arc function $f(a) = k$ where $k \in \mathbb{N}$ the expression $E(a) = \mathsf{v}_1 + + \ldots + + \mathsf{v}_k$ where $++$ denotes multiset addition and $\mathsf{v}_i : C(p)$ are typed variables with $i = 1..k$, and p is the place of arc a. Intuitively, we use a different variable for each token that may traverse an arc. The total number of variables needed in a model is thus $\sum_{a \in A} f(a)$. A further change is in the initial marking, where each place p is assigned the same number of tokens as in the standard network, and all tokens have as color the one color in p's color set. We call such a colored Petri net the *trivial coloring* of the initial network.

We denote by $\mathcal{C}(x)$ the one color in the color set of a place/transtition x. In order to identify precisely the variables used in the expression of an arc $(x, y) \in A$ we denote the variables by $\mathsf{v}_{x,y,i}$, where $i = 1..f((x, y))$. We also use the shorthand notation $\mathsf{v}_{a,i}$ to denote the i-th variable on arc $a \in A$.

Definition 6 (Trivial coloring of a Petri net). *Given a standard Petri net $N = (P, T, A, f, M_0)$, we call a* trivial coloring of N *a colored Petri net $T(N) = (P, T, A, \Sigma, C, E, \mathbb{M}, \mathbb{Y}, M_0')$ such that:*

- *$\Sigma = \bigcup_{p \in P} C_p \cup \bigcup_{t \in T} C_t$ where $C_t : \{C_p \mid p \in P\} \to \mathbb{N}$ is a multiset such that:*

$$C_t(C_p) = \begin{cases} 0 & (p,t) \notin A \text{ and } (t,p) \notin A \\ f((p,t)) & (p,t) \in A \text{ and } (t,p) \notin A \\ f((t,p)) & (p,t) \notin A \text{ and } (t,p) \in A \\ f((p,t)) + f((t,p)) & \text{otherwise} \end{cases};$$

- *$C : P \cup T \to \Sigma$, such that $C(x)$ is a record color set defined as above if $x \in P$ and a multiset defined as above if $x \in T$;*

- $E(a) = {}^{++}\sum_{1 \leq i \leq f(a)} \mathsf{v_{a,i}} = \mathsf{v_{a,1}} + + \cdots + + \mathsf{v_{a,f(a)}}$, *for all* $a \in A$, *where* $\mathsf{v_{a,i}} : C(p)$ *with* p *being the place of arc* a*;*
- $\mathbb{M}$ *is the set of markings;*
- $\mathbb{Y}$ *is the set of steps;*
- $M_0'(p) = M_0(p)\,`\mathcal{C}(p)$, *for all* $p \in P$.

Example 4. An example of a trivial coloring of the Petri net described in Example 3 is given in Figure 2.

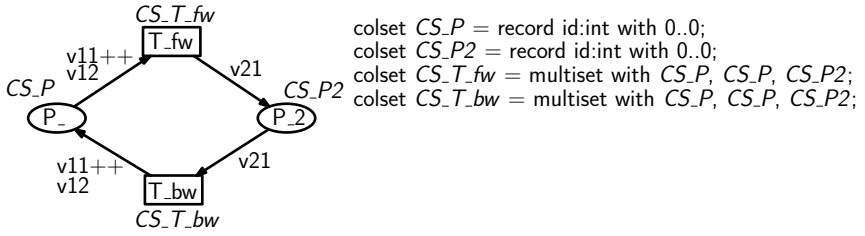


Fig. 2. Trivial coloring of a Petri net structural implementation of a dimerization model

Definition 7 (Implementation of a reaction-based model as a colored Petri net). *We say that a colored Petri net* N structurally implements *a given reaction-based model* M *iff* N^**, the unfolding of* N*, structurally implements model* M *in the sense of Definition 3.*

Proposition 1. *The unfolding* $T(N)^*$ *of a trivial coloring* $T(N)$ *of a standard Petri net* N *is equivalent to the initial net* N *(as every color set has exactly one color).*

Proposition 2. *If a standard Petri net* N *structurally implements a reaction-based model* M*, then its trivial coloring* $T(N)$ *structurally implements the same model* M*.*

Proof. By Proposition 1, N and $T(N)^*$ are equivalent, thus the unfolding of $T(N)$ structurally implements model M and, by Definition 7, $T(N)$ structurally implements M.

3.3 Type Refinement of Colored Petri Nets

Refinements of Petri nets have been a subject of interest for many years. In particular, we are concerned here with the work of Charles Lakos, who has identified and formalized three types of refinements: *type refinement*, *subnet refinement* and *node refinement*, see [11] for details. The concepts of type and node refinement have been further extended by Choppy et. al., see [3]. We prove in this paper that a full structural refinement of a model can be implemented via a type refinement of the colored Petri net representing the model.

We recall now the definition of type refinement of a colored Petri net as it was proposed in [11].

Definition 8 ([11]). *Let N and N' be two colored Petri nets. A morphism $\Phi : N \to N'$ captures a* type refinement of a colored Petri net *if:*

1. *Φ is the identity function on P, T, A;*
2. *$C(x) <: \Phi(C)(x)$, for all $x \in P \cup T$;*
3. *$\Phi(1\ `(x, c)) = 1\ `(x, \Pi_{\Phi(C)(x)}(c))$ for all $x \in P \cup T$ and for all $c \in C(x)$;*
4. *$\Phi(E^-(1\ `(t, m)))(p) = \Pi_{\Phi(C)(p)}(E(p, t)(m)) = \Phi(E)(p, t)(\Pi_{\Phi(C)(t)}(m))$, for all $(p, t) \in A$ and for all $(t, m) \in \mathbb{Y}$;*
5. *$\Phi(E^+(1\ `(t, m)))(p) = \Pi_{\Phi(C)(p)}(E(t, p)(m)) = \Phi(E)(t, p)(\Pi_{\Phi(C)(t)}(m))$, for all $(t, p) \in A$ and for all $(t, m) \in \mathbb{Y}$.*

A morphism that captures a type refinement is a *system morphism*, see [11], which means that it is a *behavior-respecting* mapping of two colored Petri nets. Expressing structural refinement as a type morphism will thus guarantee that the behavior of the initial network is preserved in the refined network. Moreover, as discussed in [12], type refinement ensures bisimilarity between the initial and the refined network.

Note that for every refined state or action there exists a corresponding abstract state or action, resp. via the projection from subtype to supertype. Also note that in Definition 8, N denotes the refined network.

4 Full Structural Refinement as Type Refinement of Colored Petri Nets

In this section we prove that the full structural refinement of a reaction-based model implemented as a Petri net can be implemented as a type refinement of the trivial coloring of the Petri net. We give a coloring strategy (type refinement) for implementing a full structural data refinement of a model represented as a Petri net, and conclude by proving that our construction indeed implements the required full structural data refinement.

4.1 Implementing a Full Structural Model Refinement via a Type Refinement in a Colored Petri Net Model

Intuitively, species refinement implies replacing each species with a non-empty set of species. This can be done in a colored Petri net by replacing for each place representing a species its default color set by a new record or enumeration color set having as many elements as the set of species that its corresponding species refines to. Or, assuming color sets defined as records, by replacing a single value field with a new field with as many possible values as the cardinality of the refined subspecies set. Formally, we need to define a morphism from the refined colored Petri net to the initial colored Petri net that respects all the properties of a type refinement, as described in [11] and presented in Section 3.3.

Definition 9 (Colored Petri net implementation of a structural refinement of a reaction network model). *We say that a colored Petri net N structurally implements the full structural refinement of a model M as described by a refinement relation ρ iff the unfolding of N, N^*, structurally implements the full structural refinement of M, $\rho(M)$ in the sense of Definition 3.*

We describe next a type refinement of a given trivial coloring of a Petri net implementation of a reaction-based model M that captures the full structural data refinement of M as described by a given refinement relation ρ.

Algorithm 1 TypeRef

function TypeRef(N, ρ)
  $\Sigma' \leftarrow \varnothing$;
  ▷ create the new color sets based on the old ones;
  for all $p \in P$ **do**
    $cs \leftarrow C(p)$;
    define a new color set cs' that extends cs with a new field with $\rho(\delta^{-1}(p))$ values;
    $\Sigma' \leftarrow \Sigma' \cup \{cs'\}$;
    $C'(p) \leftarrow cs'$;
  end for
  for all $t \in T$ **do**
    define cs as a multiset $cs : \{C'(p) \mid p \in P\} \rightarrow \mathbb{N}$ such that $cs(C'(p)) = C(t)(C(p)),\ \forall p \in P$;
    $\Sigma' \leftarrow \Sigma' \cup \{cs\}$;
    $C'(t) \leftarrow cs$;
  end for
  ▷ re-type the arc expressions: for each variable in an arc expression, create one having as type the new color set of the place that the arc is connected to; the new arc expression is a multiset sum of these variables;
  $E' \leftarrow \varnothing$;
  for all $e \in E$ **do**
    $p \leftarrow$ the place connected to e;
    $V \leftarrow$ set of variables appearing in e;
    $V' \leftarrow \varnothing$;
    for all $v_i \in V$ **do**
      define $v_i' : C'(p)$;
      $V' \leftarrow V' \cup \{v_i'\}$
    end for
    $e' \leftarrow {}^{++}\sum_{v \in V'} v$; ▷ ${}^{++}\sum$ denotes multiset addition;
    $E' \leftarrow E' \cup \{e'\}$;
  end for
  $\mathbb{M}' \leftarrow \mu\{(p, c) \mid p \in P, c \in C'(p)\}$;
  $\mathbb{Y}' \leftarrow \mu\{(t, c) \mid t \in T, c \in C'(t)\}$;
  $\mathbb{M}'_0$ is designed such that $\sum_{c \in C'(p)} \mid \mathbb{M}'_0(p, c) \mid = \mid \mathbb{M}_0(p, \mathcal{C}(p)) \mid, \forall p \in P$;
  $N' \leftarrow (P, T, A, \Sigma', C', E', \mathbb{M}', \mathbb{Y}', M'_0)$;
return N';
end function

Let $N = (P, T, A, \Sigma, C, E, \mathbb{M}, \mathbb{Y}, M_0)$ be a trivially colored Petri net that implements a reaction-based model $M = (\mathscr{S}, \mathscr{R})$ with correspondence function δ. Let $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ be a full structural refinement relation that refines model M to model $M' = (\mathscr{S}', \mathscr{R}')$. We build a colored Petri net $N' = (P, T, A, \Sigma', C', E', \mathbb{M}', \mathbb{Y}', M_0')$ and then show that the construction is a type refinement. Moreover, we show that the resulting network implements the full structural refinement $\rho(M)$. The procedure takes as input a trivially colored Petri net that implements M, and the refinement function ρ. It then updates the color sets of the network such that the color set of each place is extended with a new field that will account for the new subtypes of the species that the place stands for. Each transition gets as color set a multiset of the color sets of its pre- and post-places, with multiplicities dictated by the cardinality of each arc expression, just like in the trivial coloring. Note that this means that the refined transition color sets are subtypes of the initial transition color sets, as multisets of subtypes of a color set that is a multiset of supertypes, with identical multiplicities.

Using a distinct variable for each token on every arc is important because it allows for exact identification of each token. One can thus encode all possible combinations of in- and out- tokens for a transition t, i.e. the full set of refinements of the reaction encoded by transition t.

Proposition 3. *Given a trivially colored Petri net N that is an implementation of a reaction-based model M, and a full structural refinement relation ρ of M, the colored Petri net $N' = \textsc{TypeRef}(N, \rho)$ is a type refinement of the initial network.*

Proof. Based on the construction described in Algorithm 1, we detail here the type refinement morphism between the two networks.

Note that N is trivially colored, so all color sets have exactly one color. The projection from any color in a color set of Σ' onto its corresponding supertype color set is the one color in the supertype color set: $\Pi_{C(x)}(c) = \mathcal{C}(x)$, for any $x \in P \cup T$, and any color $c \in C'(x)$.

We now describe a morphism $\Phi_\rho : N' \to N$ between the two networks, that is a type morphism.

1. $\Phi_\rho(x) = x$ for all $x \in P \cup T \cup A$.
2. $\Phi_\rho(C')(x) = C(x)$. By definition of the color sets in N', the color set of each place and of each transition in N' is a subtype of the color set of the same place/transition in N, i.e. $C'(x) <: \Phi_\rho(C')(x)$. Moreover, for any color $c \in C'(x) : \Pi_{\Phi_\rho(C')(x)}(c) = \Pi_{C(x)}(c) = \mathcal{C}(x)$.
3. $\forall x \in P \cup T : \forall c \in C'(x) : \Phi_\rho(1\ `(x, c)) = 1\ `(x, \Pi_{C(x)}(c)) = 1\ `(x, \mathcal{C}(x))$: for every colored place/transition in N' with color c, the morphism Φ_ρ returns the same place/transition (because Φ_ρ is the identity on $P \cup T$), having as color the projection of c on the color set of x as given by the morphism Φ_ρ, namely $\mathcal{C}(x)$.
4. $\forall (p, t) \in A : \forall (t, m) \in \mathbb{Y}' : \Phi_\rho(E'(p, t)) = E(p, t)$ and the multiset of colored tokens consumed from place p at the firing of transition t in mode m is $E'(p, t)(m)$. By construction of E', the number of consumed tokens is $E(p, t)(\mathcal{C}(t))$. The projection of every color in $C'(p)$ is $\mathcal{C}(p)$, thus we get:

$$\Phi_\rho(E^-(1`(t,m))(p)) = \Pi_{\Phi_\rho(C')(p)}(E'(p,t)(m)) = E(p,t)(\mathcal{C}(t)) = \\ = E(p,t)(\Pi_{C(t)}(m)) = \Phi_\rho(E')(p,t)(\Pi_{\Phi_\rho(C')(t)}(m)).$$

5. Similarly, $\forall(t,p) \in A : \forall(t,m) \in \mathbb{Y}' : \Phi_\rho(E'(t,p)) = E(t,p)$ and the multiset of colored tokens added to place p at the firing of transition t in mode m is $E'(t,p)(m)$. By construction of E', the number of produced tokens is $E(t,p)(\mathcal{C}(t))$. The projection of every color in $C'(p)$ is $\mathcal{C}(p)$, thus we get:

$$\Phi_\rho(E^+(1`(t,m))(p)) = \Pi_{\Phi_\rho(C')(p)}(E'(t,p)(m)) = E(t,p)(\mathcal{C}(t)) = \\ = E(t,p)(\Pi_{C(t)}(m)) = \Phi_\rho(E')(t,p)(\Pi_{\Phi_\rho(C')(t)}(m)).$$

Because the morphism Φ_ρ respects all conditions for being a type refinement of a Petri net it follows that Algorithm 1 computes a type refinement of its input Petri net.

Theorem 1. *Given a reaction-based model $M = (\mathscr{S}, \mathscr{R})$, a structural refinement relation $\rho \subseteq \mathscr{S} \times \mathscr{S}'$, and a colored Petri net $N = (P, T, A, \Sigma, C, E, \mathbb{M}, \mathbb{Y}, M_0)$ that is trivially colored and implements model M with function $\delta : \mathscr{S} \cup \mathscr{R} \to P \cup T$, the colored Petri net $\textsc{TypeRef}(N, \rho)$ implements the full structural ρ-refinement of model M.*

Proof. Let N' denote the refined colored Petri net $\textsc{TypeRef}(N, \rho)$, and let $M' = (\mathscr{S}', \mathscr{R}')$ denote the full structural ρ-refinement M_ρ. By construction of the refined colored Petri net N' there exists a type morphism between N' and N, as detailed in the proof of Proposition 3.

First, note that N is trivially colored and thus the network is equivalent to its unfolding (see Proposition 1). With a slight abuse of notation, we will use x to denote the unfolded equivalent of a place/transition $x \in P \cup T$, $(x, \mathcal{C}(x))$.

We show now that the unfolding of N' implements the full structural refinement of M. Let $N^* = \{P^*, T^*, A^*, f^*, M_0^*\}$ be the unfolding of N'. The color set of a place $p \in P'$ has $\mid \rho(\delta^{-1}(p)) \mid$ elements, where each color represents one refined species $S' \in \mathscr{S}'$, $(\delta^{-1}(p), S') \in \rho$. The places of N^* represent pairs (p, c) such that $p \in P$ and $c \in C'(p)$. Given that every place p has a symbolic correspondence with one species $S = \delta^{-1}(p)$ in $\mathscr{S}$, and the colors of places in N' can be thought of as the refinements of S, there exists a one-to-one correspondence between places in P^* and species in $\mathscr{S}'$. Let $\delta_\rho : \mathscr{S}' \to P^*$, with $\delta_\rho(S') = (\delta(S), c) \in P^*$ where $(S, S') \in \rho$ and no two siblings are mapped to the same value.

δ_ρ can be extended to map also reactions in $\mathscr{R}'$ to (t, m) pairs. The color m of a transition t uniquely identifies its pre- and post-places in the unfolded network, and the arc inscriptions. By definition of the color sets of transitions as multisets over the color sets of neighbouring places, it follows that every possible combination of colored tokens flowing through a transition is captured by a transition color. This means that a transition t in N' encodes all possible refinements $\rho(r)$ of the reaction $r = \delta^{-1}(t)$ that transition t stands for in N.

A transition $(t, m) \in T^*$ encodes the reaction

$$\sum_{(p,c) \in {}^\bullet(t,m)} f^*((p,c),(t,m))\delta_\rho^{-1}((p,c)) \to \sum_{(p,c) \in (t,m)^\bullet} f^*((t,m),(p,c))\delta_\rho^{-1}((p,c)).$$

The reaction $r' = \delta_\rho^{-1}(t,m)$ that a transition $(t,m) \in T^*$ implements in N'^* is a ρ-refinement of the reaction $r = \delta^{-1}(t)$ that transition t implements in N. This comes from the type refinement conditions 4 and 5 (see Definition 8). The incremental effects of executing a step (t,m) in the refined network equal the incremental effects of executing the step $(t, \Pi_{C(t)}(m))$ in the initial network. The negative incremental effect E^- encodes the left hand side of a reaction, and the positive incremental effect E^+ encodes the right hand side.

We detail here the negative incremental effect of a step, and relate it to its meaning in the model M'. $E^-(1`(t,m)) = \sum_{(p,t)\in A} p \times E((p,t))(m)$. In the unfolded network N^* a transition (t,m) is connected to places via edges $((p,c),(t,m)) \in A^*$ where $f^*((p,c),(t,m)) = E((p,t))(m)(c)$. Summing over all unfolded instances of a place in N^* yields

$$\sum_{c\in C'(p)} f^*((p,c),(t,m)) = \sum_{c\in C'(p)} E((p,t))(m)(c) =\mid E((p,t))(m) \mid .$$

Note that the arc expressions in N and N' are the same, which means that their cardinality is also the same. N implements model M, thus $|E((p,t))| = c_{i,j}$ and $|E((t,p))| = c'_{i,j}$ where $c_{i,j}$ is the stoichiometric coefficient of species $S_i = \delta^{-1}(p)$ on the left hand side of reaction $r_j = \delta^{-1}(t)$ and $c'_{i,j}$ is the soichiometric coefficient of S_i on the right hand side of r_j. Arc multiplicities in N^* represent stoichiometries, and for any place p of N' its unfolded places $\{(p,c) \mid c \in C'(p)\}$ represent the sibling species in $\rho(\delta^{-1}(p))$.

A similar argument can be made for the right hand side of a reaction, starting from the positive incremental effect of a step. With both the left and the right hand side of a reaction represented by (t,m) being a ρ-refinement of the left or right, respectively hand side of the reaction $\delta^{-1}(t)$, it follows that (t,m) implements a ρ-refinement of the reaction implemented by t.

5 Discussion

In this paper we have made a connection between the notions of type refinement of a colored Petri net proposed in [11] and that of full structural refinement of reaction network models proposed in [6]. The connection is based on modeling a reaction network system as a Petri net and using a coloring scheme that allows for easy type refinement. Starting from a Petri net implementation of a reaction-based model, we proposed a general coloring scheme that uses record color sets and further detailed the construction and how the color sets can be refined. We proved that the colored Petri net obtained by coloring the initial Petri net with our coloring strategy is also an implementation of the model implemented by the initial net. We further proved that our strategy is in fact using a type refinement that implements a full structural refinement of a model.

The size of the refined colored Petri net model We discuss here about the size of the colored Petri net model obtained by refining a given model, in terms of number of places and transitions.

A type refinement of a colored Petri net preserves the structure of the network unchanged, i.e. the number of places and transitions does not change. But the semantics of each place and transition is different, and we will therefore consider the unfolding of the colored Petri net.

Given $N = (P, T, A, \Sigma, C, E, \mathbb{M}, \mathbb{Y}, M_0)$ a trivial colored Petri net implementation of a reaction-based model $M = \mathscr{S}, \mathscr{R}$, a refinement relation $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ and a colored Petri net $N' = (P, T, A, \Sigma', C', E, \mathbb{M}', \mathbb{Y}', M_0')$ which is the implementation of the full structural $\rho-$refinement of M by Algorithm 1 with function $\delta : \mathscr{S} \cup \mathscr{R} \to P \cup T$, we discuss the size of the unfolding of N', denoted byN^*.

N has by construction $|\mathscr{S}|$ places and $|\mathscr{R}|$ transitions. In N' by construction each place representing a species $S \in \mathscr{S}$ has $\rho(S)$ colors, and will therefore unfold to $\rho(S)$ places. The total number of unfolded places is $\sum_{S \in \mathscr{S}} |\rho(S)| = |\mathscr{S}'|$. The total number of possible colors of a transition depends on the number of colors in the color set of the pre- and post-places of the transition, and on the cardinality of the arc expressions of arcs connected on either end with the transition. A transition $t \in T$ will thus unfold to

$$\prod_{p \in {}^\bullet t} \left\langle\!\!\left\langle \begin{matrix} |\rho(\delta^{-1}(p))| \\ E((p,t)) \end{matrix} \right\rangle\!\!\right\rangle \cdot \prod_{p \in t^\bullet} \left\langle\!\!\left\langle \begin{matrix} |\rho(\delta^{-1}(p))| \\ E, ((t,p)) \end{matrix} \right\rangle\!\!\right\rangle$$

transitions in N^*, which yields a total number of transitions in N^* equal to

$$\sum_{t \in T} \left(\prod_{p \in {}^\bullet t} \left\langle\!\!\left\langle \begin{matrix} |\rho(\delta^{-1}(p))| \\ E((p,t)) \end{matrix} \right\rangle\!\!\right\rangle \cdot \prod_{p \in t^\bullet} \left\langle\!\!\left\langle \begin{matrix} |\rho(\delta^{-1}(p))| \\ E, ((t,p)) \end{matrix} \right\rangle\!\!\right\rangle \right).$$

Depending on the refinement function ρ, this number can be much larger than the number of transitions in the colored network N', which successfully avoids this explosion in number of places and transitions of the network.

Consecutive full structural refinements Very often models go through several steps of refinement, as new information about the modeled system is available, and a more detailed representation is needed. We discuss in this paragraph how subsequent full structural refinements of a model can be implemented using our approach. The problem can be formulated as follows. Given a reaction-based model $M = (\mathscr{S}, \mathscr{R})$ and two refinement relations $\rho \subseteq \mathscr{S} \times \mathscr{S}'$ and $\rho' \subseteq \mathscr{S}' \times \mathscr{S}''$, obtain the full structural $\rho'-$refinement of the full structural $\rho-$refinement of M. In our construction, we start from a trivial coloring of a Petri net implementation of a model. This is however not a limitation of the approach, since subsequent refinements can be implemented as one single refinement that is the composition of the two (or more) successive refinements to be implemented.

We conclude that colored Petri nets can be used to implement full structural refinements of reaction-based models. The major advantage of using the colored Petri nets formalism lies in their ability to represent the fully structurally refined system in a compact way, using the same network structure and adding all refinement details in the colors of places and transitions.

References

1. Ralph-Johan Back and Joakim Wright. *Refinement calculus: a systematic introduction*. springer Heidelberg, 1998.
2. Claudine Chaouiya. Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4):210–219, 2007.
3. Christine Choppy, Laure Petrucci, and Alfred Sanogo. Coloured petri nets refinements. In *PNSE+ ModPE*, pages 187–201. Citeseer, 2013.
4. Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling and model perturbation. *Transactions on Computational Systems Biology XI*, pages 116–137, 2009.
5. René David and Hassane Alla. Petri nets for modeling of dynamic systems: A survey. *Automatica*, 30(2):175–202, 1994.
6. Cristian Gratie and Ion Petre. Fit-preserving data refinement of mass-action reaction networks. In Arnold Beckmann, Erzsébet Csuhaj-Varjú, and Klaus Meer, editors, *Language, Life, Limits*, volume 8493 of *Lecture Notes in Computer Science*, pages 204–213. Springer, 2014.
7. Bogdan Iancu, Elena Czeizler, Eugen Czeizler, and Ion Petre. Quantitative refinement of reaction models. *International Journal of Unconventional Computing*, 8(5-6):529–550, 2012.
8. Bogdan Iancu, Diana-Elena Gratie, Sepinoud Azimi, and Ion Petre. On the implementation of quantitative model refinement. In Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, *Algorithms for Computational Biology*, volume 8542 of *Lecture Notes in Computer Science*, pages 95–106. Springer International Publishing, 2014.
9. Kurt Jensen. Coloured petri nets: A high level language for system design and analysis. *Advances in Petri nets 1990*, pages 342–416, 1991.
10. Kurt Jensen. Coloured petri nets. volume 1, basic concepts. *EATCS Monographs in Theoretical Computer Science*, 1, 1992.
11. Charles Lakos. Composing abstractions of coloured petri nets. In *Application and Theory of Petri Nets 2000*, pages 323–342. Springer, 2000.
12. Charles Lakos and Glenn Lewis. A catalogue of incremental changes for coloured petri nets. Technical report, the International Conference of Application and Theory of Petri Nets, 1999.
13. Andrzej Mizera, Eugen Czeizler, and Ion Petre. Self-assembly models of variable resolution. *LNBI Transactions on Computational Systems Biology*, pages 181–203, 2011.
14. Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
15. Elaine Murphy, Vincent Danos, Jérôme Feret, Jean Krivine, and Russell Harmer. *Elements of Computational Systems Biology*, chapter Rule Based Modelling and Model Refinement, pages 83–114. Wiley Book Series on Bioinformatics. John Wiley & Sons, Inc., 2010.
16. Marco Pistore and Davide Sangiorgi. A partition refinement algorithm for the pi-calculus. In *Proceedings of CAV'96, volume 1102 of Lecture Notes in Computer Science*, pages 200–1. Springer-Verlag, 1996.
17. Ichiro Suzuki and Tadao Murata. A method for stepwise refinement and abstraction of petri nets. *Journal of Computer and System Sciences*, 27(1):51 – 76, 1983.

Paper V

Composition colored Petri nets for the refinement of reaction-based models

Diana-Elena Gratie and Cristian Gratie

Originally published in *Electronic Notes in Theoretical Computer Science*, Vol. 326C, pp. 51–72, in press. The final publication will be available at Elsevier via `http://dx.doi.org/10.1016/j.entcs.2016.09.018`.

Composition colored Petri nets for the refinement of reaction-based models

Diana-Elena Gratie [1] Cristian Gratie[2]

Computational Biomodeling Laboratory, Turku Centre for Computer Science
Department of Computer Science, Åbo Akademi University
Turku, Finland

Abstract

Model refinement is an important step in the model building process. For reaction-based models, data refinement consists in replacing one species with several of its variants in the refined model. We discuss in this paper the implementation of data refinement with Petri nets such that the size of the model (in terms of number of places and transitions) does not increase. We capture the compositional structure of species by introducing a new class of Petri nets, *composition* Petri nets (ComP-nets), and their colored counterpart, *colored composition* Petri nets (ComCP-nets). Given a reaction-based model with known compositional structure, represented as a ComP-net, we propose an algorithm for building a ComCP-net which implements the data refinement of the model and has the same network structure as the initial ComP-net.

Keywords: Composition Petri nets, composition colored Petri nets, compositional structure, reaction-based model, data refinement.

1 Introduction

Models represent abstractions of real systems, that capture some of the most important behavioral properties of the system. A biological system can be abstracted to a set of biochemical reactions, based on a system-level understanding of the interactions among species. The dynamics is captured in the kinetic rate constants of the reactions. One of the heaviest computational activities for dynamical models is parameter estimation. Usually in the model building process one starts with an abstraction of the system, which is subsequently refined in a stepwise manner so as to include more details. This refinement can be done in a quantitatively correct way, ensuring that at each step the model fit is preserved. Several approaches have been discussed in the literature for reusing previously computed parameters, in order to obtain a more detailed model while avoiding (at least initially) the parameter

[1] Email: dgratie@abo.fi
[2] Email: cgratie@abo.fi

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.com/locate/entcs

estimation step for all newly introduced parameters, see [9,2,7,17]. We will consider throughout this paper the concept of data refinement as defined in [5].

Colored Petri nets have been introduced as a programmable high-level class of Petri nets that combines the modeling capabilities of Petri nets with the capabilities of a programming language. They allow the use of data types and parametrization, via the use of color sets (data types) and variables, see [11]. Colors can be used to describe a system in a more compact form, e.g. by representing two identical subsystems with different actors as a single subsystem, where each element has been assigned a color set with two colors (one color for each subsystem). They can also be used to implement refinements of systems via altering the color sets, a process called *type refinement*, see [15]. A method for implementing structural refinements of models using type refinements of colored Petri nets has been proposed in [6]. We use the framework of Petri nets for representing models, and we extend it with a passive part to encode the composition of elements acting in the modeled system. We implement structural refinements of models in our extended framework via type refinements. Our approach to refinement is thus different from the transition refinement discussed in [21] or the transition/place stepwise refinement discussed in [20].

In this paper we focus on qualitative Petri nets, as the goal is to introduce a new class of Petri nets suitable for automatable structural refinement of models. Particularities regarding the continuous and stochastic approaches are beyond the scope of this paper. We consider as a starting point the standard Petri nets, and not colored Petri nets (although a recent book on Petri nets, [18], defines markings as multisets of several types of tokens – a definition similar to that of colored Petri nets) because any colored Petri net can be unfolded to a corresponding equivalent standard Petri net.

The paper is organized as follows: in Section 2 we detail the concept of structural model refinement, as discussed in [4,5]. In Section 3 we introduce the concept of Petri nets with a compositional part, which can capture not only the functioning of a model, but also the compositional relationships between its elements. We first introduce composition Petri nets (ComP-nets), and then give a method of coloring a given ComP-net into a colored composition Petri net (ComCP-net). We discuss next in Section 4 how to implement reaction-based models as Com(C)P-nets. We detail the implementation of model refinement using ComCP-nets in Section 5, and we draw some conclusions in Section 6.

2 Model refinement

We give in this section a formal definition of reaction-based models with known composition of their species. We then introduce the data refinement of such models, in the spirit of [4], but with an explicit distinction between *atomic* and *refined* species, as first presented in [5].

Intuitively, a reaction-based model consists of a set of reactions, usually represented as rewriting rules over a given set of species. For example, consider the

following chemical reaction:

$$2H_2 + O_2 \to 2H_2O \ . \tag{1}$$

We distinguish in this paper between *atomic* species, which – as far as the considered model is concerned – cannot be divided into constituent parts, and *complex* species, which consist of several atomic species. We rely on multisets for encoding the linear combinations of species on either side of a reaction, as well as for denoting the composition of complex species. We denote multiset addition by $+\!+$, and repetitive multiset addition by ${}^{+\!+}\!\sum$.

Definition 2.1 [1] Let $S = \{s_1, s_2, \ldots\}$ be a set of elements. A *multiset* over S is a function $\sigma : S \to \mathbb{N}$, which maps each element s of S into a non-negative integer $\sigma(s)$ called the *multiplicity* (or *number of instances*) of s in σ. The multiset σ can also be written as:

$$\sigma = {}^{+\!+}\!\sum_{s \in S} \sigma(s) \backprime s = \sigma(s_1) \backprime s_1 +\!+ \sigma(s_2) \backprime s_2 +\!+ \ldots \ ,$$

where the zero coefficient terms can be omitted on the right hand side.

Whenever $\sigma(s) > 0$ we say that σ *contains* s and we write this as $s \in \sigma$. Furthermore, for any two multisets σ, τ over S, we define their *sum* as the multiset $\sigma +\!+ \tau : S \to \mathbb{N}$ satisfying $(\sigma +\!+ \tau)(s) = \sigma(s) + \tau(s)$, for any $s \in S$. The set of all multisets over S will be denoted by S_{MS}.

We now go back to our example reaction (1) and formalize it as a reaction-based model with known composition of its species. We consider that the atomic species are the hydrogen and oxygen atoms and write this as $\Gamma = \{H, O\}$. The complex species are the hydrogen, oxygen and water molecules and are encoded as multisets over Γ to yield the set of complex species $\Delta = \{2 \backprime H, 2 \backprime O, 2 \backprime H +\!+ 1 \backprime O\}$. The given reaction then becomes a pair of multisets corresponding to the respective stoichiometric coefficients:

$$2 \backprime (2 \backprime H) +\!+ 1 \backprime (2 \backprime O) \to 2 \backprime (2 \backprime H +\!+ 1 \backprime O) \ . \tag{2}$$

Note that atomicity is relative to the considered model, more precisely it depends on the chosen level of detail. For example, in a larger model where the focus is on macromolecules rather than atoms, the complex species from the previous reaction might be considered to be atomic.

We formalize in the following definition the intuition presented above for reaction-based models.

Definition 2.2 A reaction-based model with known composition of its species is a tuple $M = (\Gamma, \Delta, R)$, where:

- Γ is a set of *atomic species*.
- $\Delta \subseteq \Gamma_{MS}$ is a set of *complex species*, defined as multisets over the set of atomic species Γ, where the intuition is that any complex species $\sigma \in \Delta$ contains at least two instances of atomic species, i.e. $\sum_{A \in \Gamma} \sigma(A) \geq 2$.

- $R \subseteq (\Gamma \cup \Delta)_{MS} \times (\Gamma \cup \Delta)_{MS}$ is a set of *reactions* written as $\alpha \to \beta$ or, alternatively, (α, β), where $\alpha, \beta \in (\Gamma \cup \Delta)_{MS}$ are multisets encoding the stoichiometric coefficients corresponding to the left- and right-hand sides of the reaction, respectively.

The goal of refinement is to introduce details into the model, in the form of distinguishing several subspecies or variants of a given species. The distinction between the subspecies is very often drawn by post-translational modifications such as acetylation, phosphorylation, etc., by cell differentiation, but it could also account for different possible types of a particular trait (e.g. fur color of animals in a breeding experiment). This type of refinement is called *data refinement*, because it focuses on refining the species (data) of the model.

Definition 2.3 Let Γ and Γ' be two sets of atomic species. A function $\rho : \Gamma \to 2^{\Gamma'}$ is called an *atomic refinement function* if the following conditions hold:

(a) $\rho(A) \neq \varnothing$, for all $A \in \Gamma$;

(b) $\rho(A_1) \cap \rho(A_2) = \varnothing$, for all $A_1, A_2 \in \Gamma$ with $A_1 \neq A_2$;

(c) $\bigcup_{A \in \Gamma} \rho(A) = \Gamma'$.

A species A' is called an *atomic ρ-refinement* of species A if $A' \in \rho(A)$.

Note that the definition of atomic refinement is equivalent to the definition given in [4] for the species refinement relation, with the distinction that ρ is a function rather than a relation. Moreover, in this paper we also consider the composition of species and, in this context, the atomic refinement will *propagate* throughout the model and induce the refinement of all complex species and, subsequently, the refinement of reactions, following a similar intuition to that presented in [4]. Note that in this paper we prefer a formulation based on multisets rather than vectors, since the former are more common in the literature of Petri nets.

Definition 2.4 Let Γ and Γ' be two sets of atomic species and $\rho : \Gamma \to 2^{\Gamma'}$ an atomic refinement function.

(i) A complex species $\sigma' \in \Gamma'_{MS}$ is a *ρ-refinement* of a complex species $\sigma \in \Gamma_{MS}$, written as $\sigma' \in \rho(\sigma)$, if the multiplicity of any species $A \in \Gamma$ in σ equals the sum of the multiplicities of all its ρ-refinements $A' \in \rho(A)$ in σ', i.e.

$$\rho(\sigma) = \left\{ \sigma' \in \Gamma'_{MS} \mid \sum_{A' \in \rho(A)} \sigma'(A') = \sigma(A), \ \textit{for all } A \in \Gamma \right\}.$$

Given a set of complex species $\Delta \subseteq \Gamma_{MS}$, we will use $\rho(\Delta)$ to refer to the set of all ρ-refinements of complex species from Δ, i.e. $\rho(\Delta) = \bigcup_{\sigma \in \Delta} \rho(\sigma)$.

(ii) Let $\Delta \subseteq \Gamma_{MS}$ be a set of complex species. A multiset of species $\alpha' \in (\Gamma' \cup \rho(\Delta))_{MS}$ is a ρ-refinement of a multiset $\alpha \in (\Gamma \cup \Delta)_{MS}$, written as $\alpha' \in \rho(\alpha)$, if the multiplicity of any species $S \in \Gamma \cup \Delta$ in α is equal to the sum of the multiplicities of all its ρ-refinements $S' \in \rho(S)$ in α', i.e.

$$\rho(\alpha) = \left\{ \alpha' \in (\Gamma' \cup \rho(\Delta))_{MS} \mid \sum_{S' \in \rho(S)} \alpha'(S') = \alpha(S), \ \textit{for all } S \in \Gamma \cup \Delta \right\}.$$

(iii) A reaction $\alpha' \to \beta'$ is a ρ-refinement of a reaction $\alpha \to \beta$ if $\alpha' \in \rho(\alpha)$ and $\beta' \in \rho(\beta)$, i.e.

$$\rho((\alpha, \beta)) = \rho(\alpha) \times \rho(\beta) \ .$$

(iv) Let $M = (\Gamma, \Delta, R)$ and $M' = (\Gamma', \Delta', R')$ be two reaction-based models with known composition of their species and $\rho : \Gamma \to 2^{\Gamma'}$ an atomic refinement function. We say that M' is a *structural ρ-refinement* of M if $\Delta' = \rho(\Delta)$ and $R' \subseteq \bigcup_{r \in R} \rho(r)$. If we have equality in the latter relation, we say that M' is *the full structural ρ-refinement* of M.

While the definition of atomic refinement seems to imply that all atomic species are to be refined, the refinement of an atomic species A is *nontrivial* only as long as $|\rho(A)| \geq 2$, i.e. A has at least two distinct variants in the refined model. In this context, whenever $|\rho(A)| = 1$ we will say that A undergoes a *trivial* atomic refinement (which translates to a renaming of A in the refined model).

3 Petri Nets with a Compositional Part

In this section we introduce a new class of Petri nets, *composition Petri nets*. Such nets have two parts: an *active* part, that behaves as a standard Petri net, and a *passive* part, with transitions whose role is to describe how places in the network relate to one another, i.e. how elements in some places are composed of elements in other places.

We assume the reader is familiar with the concept of Petri nets, but we recall some of the definitions and notations to make the paper self-contained. For an introduction, we refer to [19]; for more recent definitions, concepts, extensions and applications to biology we refer to [18,3,13,14].

Definition 3.1 [11] A Petri net is a tuple $N = (P, T, A, E, I)$ where P and T are disjoint sets of *places* and *transitions*, respectively; $A \subseteq P \times T \cup T \times P$ is the set of *arcs*; $E : A \to \mathbb{N}_+$ is an *arc expression* function (also called weight function); and $I : P \to \mathbb{N}$ is an *initialization* function, assigning to each place a nonegative integer that represents the number of tokens in that place.

For a transition t, the set of its *pre-places* (places p such that there exists an arc from p to t) is denoted by ${}^\bullet t$; the set of its *post-places* (places p such that there exists an arc from t to p) is denoted by $t^\bullet$. An arc from a place p to a transition t is denoted by a pair $(p, t) \in A$, and an arc from a transition t to a place p is denoted by the pair $(t, p) \in A$.

3.1 Composition Petri nets (ComP-nets)

In this subsection we extend the definition of standard Petri nets with a *compositional part*. We do this by adding a set of *non-fireable composition transitions* and arcs connecting them with the places of the network. Their combined semantics represents the structural composition of the elements represented as places. Thus, a Petri net model describing the dynamics of a system can also include as a subnetwork the composition of the systems' entities (species). This is introduced formally in the following definition.

Definition 3.2 A *composition Petri net (ComP-net)* is a tuple $N = (P, T_c, T, A_c, A, E, I)$ with the following components:

(i) P, T, A, I represent the set of places, set of transitions, set of arcs and the initialization function of places, respectively, as for standard Petri nets.

(ii) T_c is a finite set of *composition transitions* such that $P \cap T_c = \varnothing$ and $T \cap T_c = \varnothing$. These transitions are used for depicting the compositional structure of places with respect to other places. Composition transitions never fire, irrespective of the marking of the network, and are also called *passive (non-active) transitions*. The regular transitions are, in contrast, called *active*.

(iii) $A_c \subseteq P \times T_c \cup T_c \times P$ is a set of *composition arcs* such that:
- for any place $p \in P$, there is *at most one* incoming composition arc; if there is no composition arc pointing to a place, then that place is considered *atomic*;
- for every composition transition $t_c \in T_c$ there is *at least one* incoming composition arc connecting a place to it, and *exactly one* outgoing composition arc connecting t_c to a place;
- the graph induced by the composition arcs and the places and transitions they connect is acyclic.

(iv) $E : A \cup A_c \to \mathbb{N}_+$ is an *arc expression* function, such that:
- the arc expression of a composition arc (t_c, p) (where $t_c \in T_c$ and $p \in P$) is always 1;
- the arc expression of a composition arc from a place $p \in {}^\bullet t_c$ to a composition transition t_c has the meaning that the post-place of t_c contains $E((p, t_c))$ copies of p;
- the arc expression of regular arcs has the usual meaning.

We say that $(P, T_c, A_c, E|_{A_c})$ is the *compositional part* of the network, and $(P, T, A, E|_A, I)$ is the *active part* of the network. Here, for a given set S, $E|_S$ denotes the restriction of the arc expression function E to arcs in S.

For a ComP-net, the properties of standard Petri nets (e.g. boundedness, liveness, deadlock, conflict, invariants, reachability graph) can be generalized, and they will refer only to the *active part of the network*.

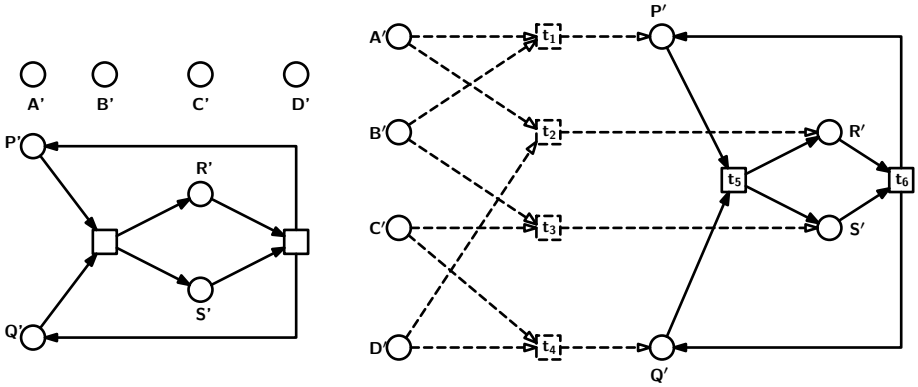
The advantage of ComP-nets is that they can explicitly represent both the dynamics of a system and the composition relationships between its elements (places). Note that there may exist pairs of transitions (t_c, t) where $t_c \in T_c$ and $t \in T$ such that ${}^\bullet t_c = {}^\bullet t$ and $t_c^\bullet = t^\bullet$, i.e. t and t_c have the same pre-places and the same post-places. This can happen because the semantics of such transitions are different. Note also that a ComP-net may contain places that do not take part in any active transition, but which are still compositionally important, and thus must appear in the place set for the compositional structure. Moreover, the fact that a place can have at most one incoming composition arc means that its compositional structure (if any) is unique.

Example 3.3 Consider a model M consisting of atomic species $\Gamma = \{A, B, C, D\}$, complex species $\Delta = \{P, Q, R, S\}$, and a single reversible reaction $P + Q \rightleftarrows R + S$.

Assume that the composition of the complex species is given by:

$$
\begin{aligned}
P &= 1\backslash A \mathbin{+\!\!+} 1\backslash B \ , \\
Q &= 1\backslash C \mathbin{+\!\!+} 1\backslash D \ , \\
R &= 1\backslash A \mathbin{+\!\!+} 1\backslash D \ , \\
S &= 1\backslash B \mathbin{+\!\!+} 1\backslash C \ .
\end{aligned}
$$

This model can be represented as a Petri net as shown in Figure 1a, where the atomic species are isolated places. The same model can be represented as a ComP-net, as shown in Figure 1b. From the figure it becomes clear what is the composition of species P, Q, R, S, namely that they are complexes $A{:}B, C{:}D, A{:}D$, and $B{:}C$, respectively. Moreover, from the network structure the reader can get an intuition on how atomic species are interchanged between complex species via active transitions (e.g. one molecule of A from P and one molecule of D from Q bind to form one R).



(a) (b)

Fig. 1. The representation of a reversible reaction $P + Q \rightleftarrows R + S$ as: a) a standard Petri net; b) a ComP-net. Circles represent places; solid squares represent active transitions; dashed squares represent passive transitions; solid arrows represent arcs; dashed arrows are arcs connected at one end to a passive transition. Figure generated using Snoopy [8].

3.2 Composition Colored Petri nets (ComCP-nets)

Sometimes, due to the complexity of a network, it becomes unfeasible or very difficult to read the corresponding Petri net. In such cases, an extension of standard Petri nets, colored Petri nets, might help reduce the size of the model. A complete description of colored Petri nets, their properties and applications can be found in [11,12,13]. Here, we consider the definitions in [13]. We extend colored Petri nets to include a compositional part, in a similar manner as we extended standard Petri nets in the previous subsection.

Notations. We use $|S|$ to denote the cardinality of a set or multiset S. For an arc expression, $|E(a)|$ denotes the cardinality of the expression. $i\backslash S$ where i is a nonnegative integer denotes i copies of S, where S can be a species, a color or a color set. If S is a color set, then $i\backslash S$ is the set of all possible ways of choosing i (not

necessarily distinct) colors from the color set S, see Table 1 for an example. In the definition of colored Petri nets, the following standard notations are used: $EXPR_V$ denotes the set of valid (under the used inscription language) expressions using variables from the typed variable set V; $Type[e]$ denotes the type of an expression $e \in EXPR$, or that of a variable e. We recall further some notions and notations we will use in this paper. The *variables of a transition* t are the set of free variables that appear in t's guard and in the arc expressions of arcs connected to t. This set is denoted by $Var(t) \subseteq V$ [13]. A *binding* of a transition t is a function b mapping each variable $v \in Var(t)$ into a value $b(v) \in Type[v]$. $B(t)$ denotes the set of all bindings for transition t [13]. A pair (t, b) with $t \in T$ and $b \in B(t)$ is called a *binding element* in [13], and a *transition instance* in [16]. We use here the terminology from [16]: $t(b)$ denotes the instance of transition t with binding b; $\mathcal{I}_T(t)$ denotes the set of all transition instances of transition t, and $\mathcal{I}_T = \bigcup_{t \in T} \mathcal{I}_T(t)$ denotes the set of all transition instances for all transitions in T. A *place instance* is a pair (p, c) with $p \in P$ and $c \in C(p)$; $\mathcal{I}_P(p)$ denotes the set of all place instances of p, and $\mathcal{I}_P = \bigcup_{p \in P} \mathcal{I}_P(p)$ denotes all place instances of all places in P [16].

Definition 3.4 [13] A *colored Petri Net (CP-net)* is a tuple $N = (P, T, A, \Sigma, V, C, G, E, I)$ satisfying the requirements below:

(i) P is a finite set of *places*.

(ii) T is a finite set of *transitions* such that $P \cap T = \varnothing$.

(iii) $A \subseteq P \times T \cup T \times P$ is a finite set of *arcs*.

(iv) Σ is a finite set of non-empty types, called *color sets*.

(v) V is a finite set of *typed variables*, where $Type[v] \in \Sigma$, for all v in V.

(vi) $C : P \to \Sigma$ is a *color set* function. It assigns a color set to each place.

(vii) G is a *guard* function that defines conditions for transitions. It is defined from T into expressions over the variables set V, i.e. $EXPR_V$, such that $Type[G(t)] = Bool$, for all transitions t in T.

(viii) $E : A \to EXPR_V$ is an *arc expression* function such that $Type[E(a)] = C(p(a))_{MS}$, for all arcs $a \in A$, where $p(a)$ is the place corresponding to arc a, and $C(p(a))_{MS}$ is a multiset of elements with color set $C(p(a))$.

(ix) I is an *initialization* function that assigns to each place p an initialization expression such that $Type[I(p)] = C(p)_{MS}$.

We want to use ComCP-nets as a means to easily model and implement the structural refinement of a system, as described in Section 2. For this, the key ingredient is the choice of color sets, especially for the complex places. The color sets should reflect the composition of places and, moreover, do it in such a way that the process of assigning color sets to complex places can be done automatically. Thus, for atomic places we propose the use of simple color sets, e.g. int or Enumeration. For complex places, the corresponding color set contains all possible multisets over the color sets of its constituent atomic places, with multiplicities dictated by the actual composition of the place. We provide in what follows a coloring example using this strategy.

Example 3.5 Let P be a molecule with two possible states, and let P_2 and P_3 denote its dimer and trimer, respectively. We will use P, P_2 and P_3 to denote both the actual molecules and the places representing them in a Petri net. We list in Table 1 a possible definition of color sets for the three entities.

Species	**Color set**	**Colors**
P	CS_P = enum with a, b	$\{a, b\}$
P_2	CS_P$_2$ = bag 2`CS_P	$\{2`a, 1`a +\!\!+ 1`b, 2`b\}$
P_3	CS_P$_2$ = bag 3`CS_P	$\{3`a, 2`a +\!\!+ 1`b, 1`a +\!\!+ 2`b, 3`b\}$

Table 1
Coloring strategy for the dimer and trimer of a molecule

Definition 3.6 A *composition colored Petri net (ComCP-net)* is a tuple $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ that satisfies the following requirements:

(i) P, T_c, T, A_c, A satisfy the constraints of Definition 3.2.

(ii) Σ, V, I have the usual meaning, namely the set of color sets, the set of variables, and the initialization function, respectively.

(iii) $C : P \to \Sigma$ is the *color function* assigning color sets to places such that:
- all atomic places have disjoint color sets, and
- for all complex places $p \in P$, $C(p) = {}^{++}\sum_{p' \in {}^\bullet t_c} |E(p', t_c)|`C(p')$, where t_c stands for the composition transition encoding the composition of p, i.e. $t_c^\bullet = \{p\}$, ;

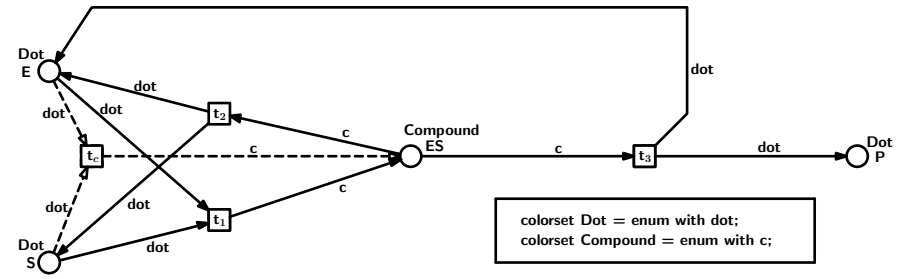
(iv) $G : T_c \cup T \to EXPR_V$ is the *guard function*, such that for each composition transition $t_c \in T_c$ with $t_c^\bullet = \{p\}$ there exists exactly one binding for which the guard is true for each color in $C(p)$.

(v) $E : A \cup A_c \to EXPR_V$ is the *arc expression function*, defined such that for every composition transition $t_c \in T_c$ with $t_c^\bullet = \{p\}$: $E(t_c, p) = {}^{++}\sum_{p' \in {}^\bullet t_c} E(p', t_c)$.

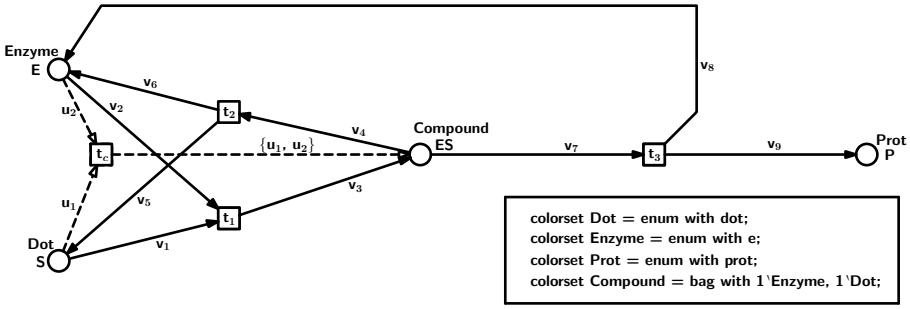
We say that $(P, T_c, A_c, \Sigma, V, C, G|_{T_c}, E|_{A_c}))$ is the *compositional part* of the network, and $(P, T, A, \Sigma, V, C, G|_T, E|_A, I)$ is the *active part* of the network.

For a ComCP-net, the properties of colored Petri nets (e.g. boundedness, liveness, deadlock, conflict, invariants, reachability graph) can be generalized, and they will refer only to the *active part of the network*. We extend the notion of transition instance (binding element) to cover the composition and the active transitions of a ComCP-net respectively. We denote by $\mathcal{I}_T$ the set of active transition instances, i.e. pairs (active transition, binding of variables): $\mathcal{I}_T = \{(t, b) \mid t \in T, b \in B(t)\}$ where $B(t)$ is the set of all possible bindings for a transition t. We use $\mathcal{I}_T\langle true\rangle$ to denote the set of active transition instances whose guard evaluates to *true*: $\mathcal{I}_T\langle true\rangle = \{(t, b) \in \mathcal{I}_{T'} \mid G(t)\langle b\rangle = true\}$. Similarly, we use $\mathcal{I}_{T_c}$ and $\mathcal{I}_{T_c}\langle true\rangle$ for the composition transition counterparts of these sets.

Example 3.7 Let us consider the example net in Figure 2a. This is not a ComCP-net because of several violations of the ComCP-net definition. The network suggests that E, S, P are atomic places, and ES is a complex place. But the atomic places E



(a) Example net that is not a ComCP-net.



(b) Example ComCP-net.

Fig. 2. Example of (a) an ill-defined and (b) a properly defined ComCP-net. Circles represent places; solid squares represent active transitions; dashed squares represent passive transitions; solid arrows represent arcs; dashed arrows are composition arcs. The text on top of arcs is the arc expression. Places are labeled with a name and their corresponding color set, and the color set definition is given in the inset. Arc expressions in (a) are values, and in (b) are typed variables with the type given by the color set of the place connected to the arc. Figure generated using Snoopy [8].

and S have the same color set, while the definition requires that atomic places are assigned disjoint color sets (to allow for the identification of colors that come from different places in the color of a complex place). Moreover, the color set of ES is independent of the color sets of E and S, which compose it, and the arc expressions of the composition arcs do not capture any kind of composition.

4 Implementing models as composition (colored) Petri nets

In this section we introduce a way of modeling with composition (colored) Petri nets. We consider as input models of the form $M = (\Gamma, \Delta, R)$ as discussed in Section 2. Every species in M is represented by a place, and each reaction is represented by a transition. The compositional structure of each complex species is represented as the *compositional* part of the composition Petri net model. We give examples for both ComP-nets and ComCP-nets.

4.1 Implementing models as ComP-nets

Definition 4.1 Let $M = (\Gamma, \Delta, R)$ be a reaction-based model with known composition of its species. We say that a ComP-net $N = (P, T_c, T, A_c, A, E, I)$ *structurally implements* the reaction-based model M if there are a bijection $f_P : P \to \Gamma \cup \Delta$ between places in P and species in $\Gamma \cup \Delta$, a bijection $f_T : T \to R$ between transitions in T and reactions in R, and a bijection between composition transitions and complex species $f_c : T_c \to \Delta$ such that:

(i) for every place $p \in P$ and every composition transition $t \in T_c$ the following conditions regarding the composition transitions hold:
- $(p,t) \in A_c \Leftrightarrow f_c(t)(f_P(p)) \geq 1$ and, moreover, $E(p,t) = f_c(t)(f_P(p))$,
- $(t,p) \in A_c \Leftrightarrow f_c(t) = f_P(p)$ and, moreover, $E(t,p) = 1$;

(ii) for every place $p \in P$ and every transition $t \in T$ with $f_T(t) = \alpha \to \beta$ the following conditions hold:
- $(p,t) \in A \Leftrightarrow \alpha(f_P(p)) \geq 1$ and, moreover, $E(p,t) = \alpha(f_P(p))$,
- $(t,p) \in A \Leftrightarrow \beta(f_P(p)) \geq 1$ and, moreover, $E(t,p) = \beta(f_P(p))$.

We call the ComP-net N the (f_P, f_T, f_c)-*implementation* of model M.

Example 4.2 The ComP-net N represented in Figure 1b is an implementation of the model $M = \{\{A, B, C, D\}, \{P, Q, R, S\}, \{P + Q \rightleftarrows R + S\}\}$ presented in Example 3.3. There exist bijections f_P, f_T and f_c that satisfy the conditions in Definition 4.1. We provide the definitions of these functions in what follows. The place to species function f_P is defined as $f_P(\mathsf{X}') = X$, where X' is a place of N and X is the species with the same name that it represents, i.e. place A' represents species A of model M and so on. The composition transition to complex species function is defined as $f_c(\mathsf{t_1}) = P$, $f_c(\mathsf{t_2}) = R$, $f_c(\mathsf{t_3}) = S$, $f_c(\mathsf{t_4}) = Q$. It is easy to notice that the requirements for f_c are fulfilled. The transition to reaction function is defined as $f_T(\mathsf{t_5}) = P + Q \to R + S$, $f_T(\mathsf{t_6}) = R + S \to P + Q$. Again, the conditions on existence of arcs and their expressions are fulfilled.

4.2 Implementing models as ComCP-nets

In the colored setting, there are several aspects that one has to be very careful about. For example, consider a reaction that uses multiple instances of some species, e.g. $2A + B \to C$. If the color set of the place representing A contains more than one color, then the arc expression of the arc connecting the place that denotes A with the transition that encodes the mentioned reaction should contain variables. Moreover, the transition should have a guard so that it would not allow for two bindings that evaluate to the same multiset of colors. This can easily be implemented by considering an ordering of the elements of each color set, and a guard that tests that the values that the variables on adjacent arcs evaluate to are ordered, with a non-strict ordering. So the guards should be of the form $[!(v_i < v_j)]$, $\forall i > j$, or equivalently $(v_i \geq v_j)$, for all i, j such that $i > j$.

Definition 4.3 Let $M = (\Gamma, \Delta, R)$ be a reaction-based model with known composition of its species. We say that a ComCP-net $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ *structurally implements* the reaction-based model M if there are a bijection $f_P : \mathcal{I}_P \to \Gamma \cup \Delta$ mapping place instances $(p, c) \in \mathcal{I}_P$ to species in $\Gamma \cup \Delta$, a bijection $f_T : \mathcal{I}_T\langle true\rangle \to R$ mapping active transition instances to reactions in R, and a bijection $f_c : \mathcal{I}_{T_c}\langle true\rangle \to \Delta$ mapping composition transition instances to complex species such that:

(i) for every composition transition instance $(t_c, b) \in \mathcal{I}_{T_c}\langle true\rangle$ and every place instance $(p, c) \in \mathcal{I}_P$ such that c appears in the binding b the following conditions hold:

- $(p, t_c) \in A_c \Leftrightarrow f_c(t_c, b)(f_P(p, c)) \geq 1$ and, moreover, the corresponding arc expression satisfies $E(p, t_c)\langle b \rangle(c) = f_c(t_c, b)(f_P(p, c))$;
- $(t_c, p) \in A_c \Leftrightarrow f_c(t_c, b) = f_P(p, c)$ and, moreover, $E(t_c, p)\langle b \rangle = 1\,\text{`}c$;

(ii) for every active transition instance $(t, b) \in \mathcal{I}_T\langle true \rangle$ with $f_T(t, b) = \alpha \to \beta$ and every place instance $(p, c) \in \mathcal{I}_P$ such that c appears in the binding b the following conditions hold:
- $(p, t) \in A \Leftrightarrow \alpha(f_P(p, c)) \geq 1$ and, moreover, $E(p, t)\langle b \rangle(c) = \alpha(f_P(p, c))$;
- $(t, p) \in A \Leftrightarrow \beta(f_P(p, c)) \geq 1$ and, moreover, $E(t, p)\langle b \rangle(c) = \alpha(f_P(p, c))$.

We call the ComCP-net N the (f_P, f_T, f_c)-*implementation* of model M.

Note that there are multiple ways of representing a model M as a ComCP-net, depending on the color sets one chooses, and on the bijections f_P, f_T and f_c. Note also that, because of the bijectivity of the functions characterizing the model implementation, for every active transition it holds that each of its instances with true guards stands for a reaction in the model; moreover, a place can encode more than one species if and only if all species that it encodes take part in similar reactions, in all possible combinations. One could also formulate the previous definition to say that a ComCP-net implements a model if its unfolding implements that model.

Example 4.4 Consider a model $M = (\Gamma, \Delta, R)$ with $\Gamma = \{E, S, P\}$, $\Delta = \{ES\}$ such that $ES = 1\,\text{`}E \mathbin{++} 1\,\text{`}S$, and R containing the reactions:

$$r_1 : E + S \to ES;$$
$$r_2 : ES \to E + S;$$
$$r_3 : ES \to E + P.$$

M is a model for an enzymatic reaction, and we show next that the ComCP-net N represented in Figure 2b implements it. The place instances of N are

$$\mathcal{I}_P = \{(\mathsf{E}, \mathsf{e}), (\mathsf{S}, \mathsf{dot}), (\mathsf{P}, \mathsf{prot}), (\mathsf{ES}, 1\,\text{`}\mathsf{e} \mathbin{++} 1\,\text{`}\mathsf{dot})\}.$$

The possible active transition instances are:

$$\begin{aligned}\mathcal{I}_T = \{&(\mathsf{t}_1, \langle \mathsf{v}_1 = \mathsf{dot}, \mathsf{v}_2 = \mathsf{e}, \mathsf{v}_3 = 1\,\text{`}\mathsf{e} \mathbin{++} 1\,\text{`}\mathsf{dot}\rangle),\\ &(\mathsf{t}_2, \langle \mathsf{v}_4 = 1\,\text{`}\mathsf{e} \mathbin{++} 1\,\text{`}\mathsf{dot}, \mathsf{v}_5 = \mathsf{dot}, \mathsf{v}_6 = \mathsf{e}\rangle),\\ &(\mathsf{t}_3, \langle \mathsf{v}_7 = 1\,\text{`}\mathsf{e} \mathbin{++} 1\,\text{`}\mathsf{dot}, \mathsf{v}_8 = \mathsf{e}, \mathsf{v}_9 = \mathsf{prot}\rangle)\}.\end{aligned}$$

The only passive transition instance is $\mathcal{I}_{T_c} = \{(\mathsf{t_c}, \langle \mathsf{u}_1 = \mathsf{dot}, \mathsf{u}_2 = \mathsf{e}\rangle)\}$.

There exist bijections f_P, f_T and f_c that satisfy the conditions in Definition 4.3. We detail here the definition of these functions.

The place to species function f_P is defined as $f_P(\mathsf{X}, col(\mathsf{X})) = X$, where X is a place of N, $col(\mathsf{X})$ is its color (note that every color set has only one color), and X is the species with the same name that it represents.

The composition transition to complex species function is defined as

$$f_c(\mathsf{t_c}, \langle \mathsf{u}_1 = \mathsf{dot}, \mathsf{u}_2 = \mathsf{e}\rangle) = ES.$$

It is easy to notice that the requirements for f_c are fulfilled.

The transition to reaction function is defined as:

$$f_T(\mathsf{t}_1, \langle \mathsf{v}_1 = \mathsf{dot}, \mathsf{v}_2 = \mathsf{e}, \mathsf{v}_3 = 1\,\text{`}\mathsf{e} \mathbin{++} 1\,\text{`}\mathsf{dot}\rangle) = r_1;$$
$$f_T(\mathsf{t}_2, \langle \mathsf{v}_4 = 1\,\text{`}\mathsf{dot} \mathbin{++} 1\,\text{`}\mathsf{e}, \mathsf{v}_5 = \mathsf{dot}, \mathsf{v}_6 = \mathsf{e}\rangle) = r_2;$$
$$f_T(\mathsf{t}_3, \langle \mathsf{v}_7 = 1\,\text{`}\mathsf{dot} \mathbin{++} 1\,\text{`}\mathsf{e}, \mathsf{v}_8 = \mathsf{e}, \mathsf{v}_9 = \mathsf{prot}\rangle) = r_3.$$

Again, the conditions on existence of arcs and their expressions are fulfilled.

Example 4.5 Consider a model M consisting of atomic species $\Gamma = \{A, B\}$, complex species $\Delta = \{C, D, E\}$, with

$$C = 2\,\text{`}A;$$
$$D = 1\,\text{`}A \mathbin{++} 1\,\text{`}B;$$
$$E = 2\,\text{`}B.$$

and the set of reactions $R = \{2A \to C, A + B \to D, 2B \to E, C + E \to 2D\}$. This model can be implemented with the ComCP-net from Figure 3.

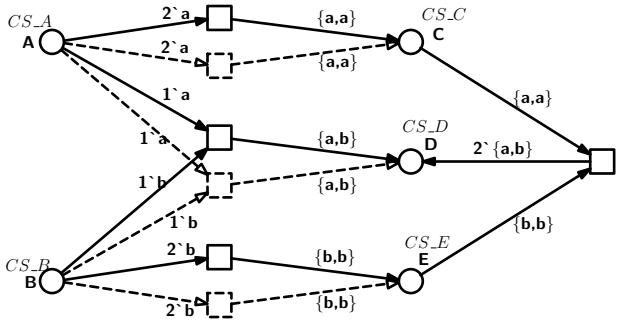


Fig. 3. The representation of an example model as a ComCP-net. The model consists of reactions $\{2A \to C, A + B \to D, 2B \to E, C + E \to 2D\}$. Circles represent places; solid squares represent active transitions; dashed squares represent passive transitions; solid arrows represent arcs; dashed arrows are arcs connected at one end to a passive transition. The name of the color set of a place is the italic text next to a place. The text on top of arcs is the arc expression. All color sets have only one color, which appears on the arc expressions. Figure generated using Snoopy [8].

4.3 From ComP-nets to ComCP-nets

In the following we give an algorithm for coloring a ComP-net to get a corresponding ComCP-net, Algorithm 1. We call the resulting ComCP-net the *natural coloring* of the given ComP-net. The ComCP-net in Figure 2b is an example of a natural coloring. Every place corresponding to an atomic species gets as color set an enumeration color set with only one element, and every place corresponding to a complex species gets as color set the set of possible multiset of all its compositional constituents' colors, each as many times as dictated by compositional arcs. The arc expressions in the built ComCP-net use a variable for each token traversing the arc, so that we don't restrict the natural coloring. All guards are set to *true*, as all color sets have exactly one color and thus there cannot exist several bindings that evaluate to the same multiset of colored tokens.

Algorithm 1 ComP_to_ComCP

Input: a ComP-net $N = (P, T_c, T, A_c, A, E, I)$;
Output: a ComCP-net $N' = (P, T_c, T, A_c, A, \Sigma, V, C, G, E', I')$;

1: **procedure** ASSIGN_CS(p) ▷ assign the color set of place p
2: $T_p \leftarrow {}^\bullet p \cap T_c$;
3: **if** $T_p = \varnothing$ **then** ▷ p is an atomic place
4: $C(p) \leftarrow$ new distinct color set with one element;
5: **return**
6: **end if**
7: $t_c \leftarrow$the one value in T_p;
8: **for all** $q \in {}^\bullet t_c$ **do**
9: **if** $C(q) =$**NIL then** ASSIGN_CS(q);
10: **end if**
11: **end for**
12: $CS_p \leftarrow {}^{++} \sum_{q \in {}^\bullet t_c} E(q, t_c)`C(q)$;
13: $\Sigma \leftarrow \Sigma \cup CS_p$;
14: $C(p) \leftarrow CS_p$;
15: **end procedure**
16:
17: $\Sigma \leftarrow \varnothing$;
18: $V \leftarrow \varnothing$;
19:
20: **for all** $p \in P$ **do**
21: $C(p) \leftarrow$**NIL**;
22: **end for**
23: **for all** $p \in P$ **do** ▷ assign color sets
24: ASSIGN_CS(p);
25: **end for**
26:
27: **for all** $a \in A \cup A_c$ **do** ▷ assign arc expressions
28: $p \leftarrow$ the place connected with arc a;
29: $V' \leftarrow \varnothing$ ▷ V' stores the variables used in the arc expression of a
30: **for all** $i \leftarrow 1$ to $E(a)$ **do**
31: create a variable $v_{a,i} : C(p)$;
32: $V' \leftarrow V' \cup \{v_{a,i}\}$;
33: **end for**
34: $E'(a) \leftarrow {}^{++}\sum_{v \in V'} v$;
35: $V \leftarrow V \cup V'$;
36: **end for**

Algorithm 1 (continued) ComP_to_ComCP

for all $t \in T_c \cup T$ **do** ▷ transition guards are all set to *true*
    $G(t) \leftarrow true$;
end for

for all $p \in P$ **do** ▷ assign initial markings
    $I'(p) \leftarrow I(p)`C(p)[0]$; ▷ the one color in the color set $C(p)$
end for

$N' = (P, T_c, T, A_c, A, \Sigma, V, C, G, E', I')$;
return N';

There are of course multiple ways of coloring the given network N. We chose here different color sets for each atomic element, so that each such element can be identified by its color set. Furthermore, we chose a representation of complex elements based on multisets, as this allows for the implementation of the refinement of a network without changing the structure of the network's implementation.

In the algorithm we assume that the network is well-defined and composition places are post-places of exactly one composition transition. The sets of places, composition transitions, transitions, composition arcs and arcs are the same in the initial and final networks.

The set of color sets contains, for places with no incoming composition arc, a color set with one color, and for places p with an incoming composition arc (t_c, p), a color set that is the set of all multisets of colors from the color sets of the pre-places of t_c, as many times as the value of the arc expression of the arc from the pre-place to t_c.

Arc expressions use a distinct variable for each colored token. We do this in order to not restrict the natural coloring, and allow for further extensions of it.

Transition guards are all set to *true*; no ordering is needed because each color set has only one element.

It is not difficult to see that, if the input of the algorithm is a ComP-net that is a (f_P, f_T, f_c)-implementation of a reaction network $M = (\Gamma, \Delta, R)$, then the output is a ComCP-net that structurally implements M.

5 Implementing Data Refinement with ComCP-nets

Colored Petri nets can be used to implement refinements of a model in a compact way, as discussed in [15,5,6]. We present here an algorithmic method for implementing the structural refinement of a model using its ComCP-net representation. Our approach differs from that of [6] via the automatic propagation of refinement from one atomic place to all places connected to it by compositional transitions. We consider the *type refinement* of colored Petri nets, namely a refinement of the color sets of some of the places in the network, see [15]. A morphism between two colored Petri nets captures a type refinement if it induces no change in the structure of the network, and the colors in the resulting network are consistently subtyped. Namely, the refinement adds details to the color sets, such that the resulting color

sets can be projected onto the initial color sets.

Definition 5.1 We say that a ComCP-net CP' is a *type refinement* of a ComCP-net CP if the compositional parts of the two networks are isomorphic and there exists a type refinement morphism between their active parts.

For a given model $M = (\Gamma, \Delta, R)$ represented as a ComCP-net $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ using the natural coloring, assume that one of the atomic species, $S \in \Gamma$, is to be refined (i.e. replaced throughout the model with several of its variants). Let γ be the number of such variants that S can be replaced with. We build a ComCP-net $N' = (P, T_c, T, A_c, A, \Sigma', V, C', G', E', I')$ to be the *type (color) refinement of* N. The sequence of steps required to implement the refinement is presented in Algorithm 2 and briefly explained here.

Reflecting the change for place q in the ComCP-net N' is done by adding more colors to the color set of q. This can be done by either adding an attribute with γ possible values, or by altering the enumeration color set such that instead of one element it has γ elements (colors). From the definition of the ComCP-net N as the natural coloring of the model M it follows that all color sets of places corresponding to complex species containing the refined species will automatically reflect the refinement (as they contain the refined color set $C(q)$). For species that are not refined, the corresponding places get as initial marking $I'(p) = I(p)$. For the refined species, there are multiple ways of choosing the initial marking for each of the newly introduced subspecies. The condition they must obey is $|I'(p)| = |I(p)|$.

The chosen method of implementing the refinement conserves the structure of the network and is thus the most compact with respect to the initial network. Moreover, based on the compositional part of the network, a simulation software that would support composition Petri nets could automatically generate the color sets for complex species based on the color sets of atomic species that are input by the modeler. This would give a significant speedup in the refinement process.

Algorithm 2 ComCP_refinement

Input: a naturally colored ComCP-net $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$; an atomic place $q \in P$ to be refined, and γ, the number of colors that q's color set refines to.

Output: the corresponding color refinement ComCP-net $N' = (P, T_c, T, A_c, A, \Sigma', V', C', G', E', I')$;

1: $C' \leftarrow C$; ▷ start with the color function of N
2: $\Sigma' \leftarrow \Sigma \setminus \{C(q)\}$; ▷ color sets from N, except the color set of place q
3: $CS \leftarrow$ *enumeration with* γ *elements*;
4: $C'(q) \leftarrow CS$; ▷ modify the color set of the place to be refined
5: $\Sigma' \leftarrow \Sigma' \cup \{CS\}$; ▷ add q's new color set to the set of color sets
6:
7: $E' \leftarrow E$;
8: **for all** $\{a \in A \cup A_c \mid a = (q, t)\ OR\ a = (t, q)\}$ **do**
9: $V_a \leftarrow$ all variables appearing in $E(a)$;

Algorithm 2 (continued) ComCP_refinement

$V' \leftarrow V' \setminus V_a$;
$V'_a \leftarrow \varnothing$;
for all $v_{a,i} \in V_a$ **do** ▷ re-type the variables of arcs connected to q
  define $v'_{a,i} : C'(q)$;
  $V'_a \leftarrow V'_a \cup \{v'_{a,i}\}$;
end for
$E'(a) \leftarrow {}^{++}\sum_{v \in V'_a} v$;
$V' \leftarrow V' \cup V'_a$;
end for

$P_q \leftarrow \{q\}$; ▷ set of places affected by the color refinement
for all $t \in T_c$ **do**
  if $(q, t) \in A_c$ **then**
    $P_q = P_q \cup t^\bullet$;
  end if
end for

for all $t \in T \cup T_c$ **do** ▷ change guards where needed
  for all $p \in t^\bullet \cup {}^\bullet t$ **do**
    if $p \in P_q$ **then**
      $G(t) \leftarrow$ new guard such that no two bindings evaluate to the same multiset of tokens;
      break;
    end if
  end for
end for
for all $p \in P$ **do** ▷ change the initial marking for places affected by the color refinement
  if $p \in P_q$ **then**
    $I'(p) \leftarrow$assign initial marking;
  else
    $I'(p) \leftarrow I(p)$;
  end if
end for

$N' = (P, T_c, T, A_c, A, \Sigma', V', C', G', E', I')$;
return N';

For the construction detailed above, the size of the model is the same as that of the initial model in terms of number of places and transitions. However, the increase in model size is encapsulated in the number of colors used in each color set, and the possible binding elements for each transition.

Theorem 5.2 *Let $M = (\Gamma, \Delta, R)$ and $M' = (\Gamma', \Delta', R')$ be two reaction-based models with known composition of their species, and $\rho : \Gamma \to 2^{\Gamma'}$ an atomic refinement function, such that M' is the full ρ-refinement of M. Let $N = (P, T_c, T, A_c, A, \Sigma,$*

V, C, G, E, I) be a naturally colored ComCP-net that is a (f_P, f_T, f_c)-implementation of M (e.g. the natural coloring of a ComP net that implements M). Then the ComCP-net $N' = (P', T'_c, T', A'_c, A', \Sigma', V', C', G', E', I')$, obtained by repeatedly running Algorithm 2 to compute a color refinement of each atomic place, structurally implements the refined model M'.

Proof. We only consider the refinement of a single atomic species and prove the claim of the theorem for this case. The result can then be easily extended for the repeated application of Algorithm 2. Let $A \in \Gamma$ be the atomic species that is refined, γ the number of variants it is refined to, and let p_A denote the place that stands for A in N.

The ComCP-net N has exactly one place for each species from model M. The set of places is the same for N', but the place instances mirror the refined model M' as follows: for each atomic place $p \in P$ except for p_A, there is only one place instance, $(p, C(p))$; for p_A there are γ instances, one for each color in the refined color set. The places corresponding to complex species have as color sets a multiset containing the color sets of the constituent places and are thus automatically updated to reflect the refinement of p_A. Each place instance will thus correspond to a refined complex species in M'.

The set of arcs is the same in the two networks, and the set of arc expressions differs only in the typing of variables.

Each active transition that is not connected to p_A or to a place that is connected to p_A via a composition transition has its guard set to true, and only one possible binding. Such transitions correspond to those reactions that refine to a singleton set in M'. Transitions connected to p_A or to a place that is connected to p_A via a composition transition have a guard that allows exactly one binding for each possible multiset of tokens. A binding (t, b) of such a transition will thus encode precisely the refinements $\rho(f_T(t, b))$ of $f_T(t, b)$.

□

Example 5.3 For the Example 4.4, let us consider a refinement where the enzyme E can be in two different conformations, E^1 and E^2, both of which can catalyze the production of P. Moreover, consider that the environment can induce the transformation of one conformation into the other, but this is not explicitly modeled in the system. In order to reflect this change, the complex species ES refines to $E^1S = 1\text{`}E^1 ++ 1\text{`}S$ and $E^2S = 1\text{`}E^2 ++ 1\text{`}S$. The new set of reactions is listed in Table 2. In order to implement the refinement for the ComCP-net in Figure 2b, we only change the color set of place E to be $\mathsf{Enzyme} = \mathsf{enum}\ \mathsf{e1}, \mathsf{e2}$. The mapping from place instances to species is straightforward, and the mapping from transition instances to reactions is captured in Table 2.

If a refinement where transitions from one conformation of the enzyme to the other is preferred to the full structural refinement (e.g. $E^1 + S \to E^2S$ is not a valid reaction), the ComCP-net can be further restricted with guards by not allowing certain bindings (e.g. $[!(\mathsf{v_1} = \mathsf{dot}\ \&\ \mathsf{v_2} = \mathsf{e1}\ \&\ \mathsf{v_3} = 1\text{`}\mathsf{e1} ++ 1\text{`}\mathsf{dot})]$).

Initial reaction	Refined reaction	Transition instance
$E + S \to ES$	$E^1 + S \to E^1S$	$(\mathsf{t}_1, \langle \mathsf{v}_1 = \mathsf{dot}, \mathsf{v}_2 = \mathsf{e1}, \mathsf{v}_3 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}\rangle)$
	$E^1 + S \to E^2S$	$(\mathsf{t}_1, \langle \mathsf{v}_1 = \mathsf{dot}, \mathsf{v}_2 = \mathsf{e1}, \mathsf{v}_3 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}\rangle)$
	$E^2 + S \to E^1S$	$(\mathsf{t}_1, \langle \mathsf{v}_1 = \mathsf{dot}, \mathsf{v}_2 = \mathsf{e2}, \mathsf{v}_3 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}\rangle)$
	$E^2 + S \to E^2S$	$(\mathsf{t}_1, \langle \mathsf{v}_1 = \mathsf{dot}, \mathsf{v}_2 = \mathsf{e2}, \mathsf{v}_3 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}\rangle)$
$ES \to E + S$	$E^1S \to E^1 + S$	$(\mathsf{t}_2, \langle \mathsf{v}_4 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_5 = \mathsf{dot}, \mathsf{v}_6 = \mathsf{e1}, \rangle)$
	$E^1S \to E^2 + S$	$(\mathsf{t}_2, \langle \mathsf{v}_4 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_5 = \mathsf{dot}, \mathsf{v}_6 = \mathsf{e2}, \rangle)$
	$E^2S \to E^1 + S$	$(\mathsf{t}_2, \langle \mathsf{v}_4 = 1\,\backslash\mathsf{e2} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_5 = \mathsf{dot}, \mathsf{v}_6 = \mathsf{e1}, \rangle)$
	$E^2S \to E^2 + S$	$(\mathsf{t}_2, \langle \mathsf{v}_4 = 1\,\backslash\mathsf{e2} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_5 = \mathsf{dot}, \mathsf{v}_6 = \mathsf{e2}, \rangle)$
$ES \to E + P$	$E^1S \to E^1 + P$	$(\mathsf{t}_3, \langle \mathsf{v}_7 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_8 = \mathsf{e1}, \mathsf{v}_9 = \mathsf{prot}, \rangle)$
	$E^1S \to E^2 + P$	$(\mathsf{t}_3, \langle \mathsf{v}_7 = 1\,\backslash\mathsf{e1} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_8 = \mathsf{e2}, \mathsf{v}_9 = \mathsf{prot}, \rangle)$
	$E^2S \to E^1 + P$	$(\mathsf{t}_3, \langle \mathsf{v}_7 = 1\,\backslash\mathsf{e2} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_8 = \mathsf{e1}, \mathsf{v}_9 = \mathsf{prot}, \rangle)$
	$E^2S \to E^2 + P$	$(\mathsf{t}_3, \langle \mathsf{v}_7 = 1\,\backslash\mathsf{e2} +\!\!+ 1\,\backslash\mathsf{dot}, \mathsf{v}_8 = \mathsf{e2}, \mathsf{v}_9 = \mathsf{prot}, \rangle)$

Table 2
Full structural refinement of an enzymatic model to consider two variants of an enzyme.

6 Conclusions

We introduced in this paper a new class of Petri nets that has capabilities for fast model refinement, when the compositional structure of the elements is known. Such Petri nets have a passive compositional part and an active part. The passive part encodes the compositional structure of the elements (species, encoded as places in the network), and all transitions in this part never fire. The active part encodes the behavior of the model.

Model refinement in some formalisms (e.g. ODE models) requires explicitly writing all possible combinations of reactions induced by replacing some species with several of their variants. With colored Petri nets, this can be done without changing the structure of the network. Internally, all these combinations are generated when binding the variables on arcs to values. Moreover, considering the compositional structure of species and choosing the colors in the manner we propose means that all species containing some atomic species that needs to be refined are automatically refined at once.

There exist also modeling frameworks that allow for a compact characterization of models and are good at handling model explosion upon refinement of a model. For example the Kappa language, see [2,7,17], allows compactness via explicitly mentioning an attribute only when its value is important, and omitting it whenever the actual value is not important, with the understanding that a reaction happens regardless of the value of that particular attribute. Refinement could resume then to adding attributes to a species, as presented e.g. for a case study of the heat shock response in [10]. The framework we are proposing allows for a similar approach of

modeling, where attributes and internal states of species can be represented in the color set. Variables can be used whenever the explicit value of some attribute is not important, and actual colors should be used in arc expressions and guards when the particular value of an attribute is important. ComCP-nets have the advantage of being graphical and adding structural information in a formalized manner to the widely used framework of Petri nets.

Some of the combinations of species that are generated when refining some species may be biologically impossible. The formalism of Petri nets is suitable for dealing with such reactions by adding guards to the "parent" reaction (the reaction that was replaced with some biologically impossible reactions). Also, if additional information is known about the model, e.g. on the way atomic species are transferred from some complex species to other complex species, it can easily be implemented by manipulating arc expressions. This will be in the scope of a future paper.

7 Acknowledgements

The authors gratefully acknowledge support from Academy of Finland through project 267915.

References

[1] Calude, C. S., P. Gheorghe, G. Rozenberg and A. Salomaa, editors, "Multiset Processing," Lecture Notes in Computer Science **2235**, Springer-Verlag Berlin Heidelberg, 2001.

[2] Danos, V., J. Feret, W. Fontana, R. Harmer and J. Krivine, *Rule-based modelling and model perturbation*, in: C. Priami, R.-J. Back and I. Petre, editors, *Transactions on Computational Systems Biology XI*, Lecture Notes in Computer Science **5750**, Springer Berlin Heidelberg, 2009 pp. 116–137.

[3] David, R. and H. Alla, "Discrete, continuous, and hybrid Petri nets," Springer Berlin Heidelberg, 2010.

[4] Gratie, C. and I. Petre, *Fit-preserving data refinement of mass-action reaction networks*, in: A. Beckmann, E. Csuhaj-Varjú and K. Meer, editors, *Language, Life, Limits*, Lecture Notes in Computer Science **8493**, Springer International Publishing, 2014 pp. 204–213.

[5] Gratie, D.-E., B. Iancu, S. Azimi and I. Petre, *Quantitative model refinement in four different frameworks, with applications to the heat shock response*, Technical Report 1067, TUCS (2013).

[6] Gratie, D.-E. and I. Petre, *Full structural model refinement as type refinement of colored Petri nets*, in: M. Heiner and A. K. Wagler, editors, *Proceedings of the 6th International Workshop on Biological Processes and Petri Nets*, CEUR Workshop Proceedings **1373** (2015), pp. 70–84.

[7] Harmer, R., *Rule-based modelling and tunable resolution*, EPTCS **9** (2009), pp. 65–72.

[8] Heiner, M., M. Herajy, F. Liu, C. Rohr and M. Schwarick, *Snoopy - a unifying Petri net tool*, in: S. Haddad and L. Pomello, editors, *Application and Theory of Petri Nets*, Lecture Notes in Computer Science **7347**, Springer Berlin Heidelberg, 2012 pp. 398–407.

[9] Iancu, B., E. Czeizler, E. Czeizler and I. Petre, *Quantitative refinement of reaction models*, International Journal of Unconventional Computing **8** (2012), pp. 529–550.

[10] Iancu, B., D.-E. Gratie, S. Azimi and I. Petre, *On the implementation of quantitative model refinement*, in: A.-H. Dediu, C. Martn-Vide and B. Truthe, editors, *Algorithms for Computational Biology*, Lecture Notes in Computer Science **8542**, Springer International Publishing, 2014 pp. 95–106.
URL http://dx.doi.org/10.1007/978-3-319-07953-0_8

[11] Jensen, K., "Coloured Petri Nets," EATCS Monographs on Theoretical Computer Science, Springer-Verlag Berlin Heidelberg, 1992.

[12] Jensen, K., "Coloured Petri Nets," Monographs in Theoretical Computer Science. An EATCS Series, Springer-Verlag Berlin Heidelberg, 1995.

[13] Jensen, K. and L. M. Kristensen, "Coloured Petri Nets," Springer-Verlag Berlin Heidelberg, 2009.

[14] Koch, I., W. Reisig and F. Schreiber, editors, "Modeling in systems biology: the Petri net approach," Computational Biology **16**, Springer-Verlag London, 2011.

[15] Lakos, C., *Composing abstractions of coloured Petri nets*, in: *Application and Theory of Petri Nets 2000*, Springer, 2000 pp. 323–342.

[16] Liu, F., "Colored Petri nets for systems biology," Ph.D. thesis, Universitätsbibliothek (2012).

[17] Murphy, E., V. Danos, J. Feret, J. Krivine and R. Harmer, *Rule-based modeling and model refinement*, Elements of Computational Systems Biology (2009), pp. 83–114.

[18] Reisig, W., "Understanding Petri Nets," Springer-Verlag Berlin Heidelberg, 2013.

[19] Reisig, W. and G. Rozenberg, editors, "Lectures on Petri Nets I: Basic Models," Lecture Notes in Computer Science **1491**, Springer-Verlag Berlin Heidelberg, 1998.

[20] Suzuki, I. and T. Murata, *A method for stepwise refinement and abstraction of Petri nets*, Journal of Computer and System Sciences **27** (1983), pp. 51 – 76.

[21] Valette, R., *Analysis of Petri nets by stepwise refinements*, Journal of Computer and System Sciences **18** (1979), pp. 35 – 46.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications
195. **Annika H. Holmbom**, Visual Analytics for Behavioral and Niche Market Segmentation
196. **Sergey Ostroumov**, Agent-Based Management System for Many-Core Platforms: Rigorous Design and Efficient Implementation
197. **Espen Suenson**, How Computer Programmers Work – Understanding Software Development in Practise
198. **Tuomas Poikela**, Readout Architectures for Hybrid Pixel Detector Readout Chips
199. **Bogdan Iancu**, Quantitative Refinement of Reaction-Based Biomodels
200. **Ilkka Törmä**, Structural and Computational Existence Results for Multidimensional Subshifts
201. **Sebastian Okser**, Scalable Feature Selection Applications for Genome-Wide Association Studies of Complex Diseases
202. **Fredrik Abbors**, Model-Based Testing of Software Systems: Functionality and Performance
203. **Inna Pereverzeva**, Formal Development of Resilient Distributed Systems
204. **Mikhail Barash**, Defining Contexts in Context-Free Grammars
205. **Sepinoud Azimi**, Computational Models for and from Biology: Simple Gene Assembly and Reaction Systems
206. **Petter Sandvik**, Formal Modelling for Digital Media Distribution

207. **Jongyun Moon**, Hydrogen Sensor Application of Anodic Titanium Oxide Nanostructures
208. **Simon Holmbacka**, Energy Aware Software for Many-Core Systems
209. **Charalampos Zinoviadis**, Hierarchy and Expansiveness in Two-Dimensional Subshifts of Finite Type
210. **Mika Murtojärvi**, Efficient Algorithms for Coastal Geographic Problems
211. **Sami Mäkelä**, Cohesion Metrics for Improving Software Quality
212. **Eyal Eshet**, Examining Human-Centered Design Practice in the Mobile Apps Era
213. **Jetro Vesti**, Rich Words and Balanced Words
214. **Jarkko Peltomäki**, Privileged Words and Sturmian Words
215. **Fahimeh Farahnakian**, Energy and Performance Management of Virtual Machines: Provisioning, Placement and Consolidation
216. **Diana-Elena Gratie**, Refinement of Biomodels Using Petri Nets

Turku
Centre *for*
Computer
Science

http://www. tucs.fi

tucs@abo.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering
- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics
- Information Systems

ISBN 978-952-12-3438-5
ISSN 1239-1883

Diana-Elena Gratie

Refinement of Biomodels Using Petri Nets

Diana-Elena Gratie

Refinement of Biomodels Using Petri Nets

Diana-Elena Gratie

Refinement of Biomodels Using Petri Nets