Consistency and Sensitivity Analysis of Multi-level Petri Net Models of Biological Systems

Shauheen Zahirazami

A Thesis

in

The Department of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

April 2013

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Shauheen Zahirazami**

Entitled: **Consistency and Sensitivity Analysis of Multi-Level Petri Net Models of Biological Systems**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________Chair
Dr. H. Akbari

______________________________ External Examiner
Dr. S. Abdelwahed

______________________________ External to Program
Dr. G. Butler

______________________________ Examiner
Dr. K Khorasani

______________________________ Examiner
Dr. L.A.C. Lopes

______________________________ Thesis Supervisor
Dr. S. Hashtrudi Zad

Approved by ____
Dr. J.X. Zhang, Graduate Program Director

March 15, 2013

____
Dr. Robin A.L. Drew, Dean
Faculty of Engineering and Computer Science

ABSTRACT

Consistency and Sensitivity Analysis of Multi-level Petri Net Models of Biological Systems

Shauheen Zahirazami

Doctor of Philosophy Concordia University 2013

The recent developments in biological experiments have awarded the research community with valuable information, which describe finely regulated systems that govern the cell dynamics. One of the greatest challenges, however, remains to represent this extensive amount of knowledge in a proper way that can be used in simulations, and validated automatically, in order to understand the dynamics and ultimately achieve a desired behaviour for the system (cell) under control.

Many tools and techniques have been proposed in the literature to address this important problem. In this research, the use of Petri nets for knowledge representation is investigated. The initial focus of this research is then to introduce a concept of consistency between Petri nets obtained from various knowledge sources. Two algorithms are provided to construct Petri net models for cell dynamics using data available in public domain biological database. The first algorithm generates a low-level model capturing protein-protein interactions and the second, produces a high-level model which describes pathway sequences and is considerably easier to analyze. Appropriate tests are developed to study consistency of such models.

In the context of biological systems, diseases that alter cell dynamics, such as cancer, can be regarded as faults in the system, and disease diagnosis and treatment will correspond to fault detection and control. In this research a framework has been proposed for sensitivity analysis in Petri net representation of biological systems. Efficient tools and procedures are developed to achieve sensitivity analysis. It is demonstrated using actual biological system models, that the results of such analysis can be used as a basis of drug discovery.

SOMMAIRE

Consistency and Sensitivity Analysis of Multi-level Petri Net Models of Biological Systems

Shauheen Zahirazami

Doctor of Philosophy Concordia University 2013

Les développements récents dans des expériences biologiques ont décerné le communauté de la recherche de précieuses informations, qui décrivent finement les systèmes réglementés qui régissent la dynamique cellulaire. Un des plus grands défis, cependant, reste à représenter cette grande quantité de connaissance d'une maniére appropriée qui peut être utilisé dans des simulations, et validé automatiquement, afin de comprendre la dynamique et, finalement, parvenir à un choix comportement du système (cellule) sous contrôle.

De nombreux outils et techniques ont été proposées dans la littérature pour résoudre ce probléme important. Dans cette recherche, l'utilisation des réseaux de Petri pour la représentation des connaissances est d'une enquête. L'objectif initial de cette recherche est donc d'introduire une notion de cohérence entre les réseaux de Petri obtenus à partir de diverses sources de connaissances. Deux algorithmes sont fournis pour construire des modèles de Petri net pour la dynamique des cellules à partir des données disponibles dans le domaine public biologique base de données. Le premier algorithme génère un modèle de bas niveau de capture interactions protéine-protéine et la seconde, un produit de haut niveau modèle qui décrit les séquences de la voie et il est beaucoup plus facile à analyser. Les tests appropriés sont mis au point pour étudier la cohérence de ces modéles.

Dans le contexte des systémes biologiques, maladies qui modifient la dynamique des cellules, telles que le cancer, peuvent être considérés comme des défauts dans le système, et le diagnostic des maladies et traitement correspond à la détection des défauts et de contrôle. Dans cette recherche, un framework a été proposée pour l'analyse de sensibilité dans la représentation de Petri net de systèmes biologiques. Des outils efficaces et des procédures sont élaborées pour arriver à une analyse de sensibilité. Il est démontré en utilisant des modèles de systèmes biologiques réels, que les résultats de cette analyse peuvent être utilisés comme base de découverte de médicaments.

In memory of my father Seyyed Mahmoud Zahirazami who always was and will forever be the motivation for all my studies, and in memory of Dr. Peyman Gohari, whose guidance was always key to improvement of my work.

ACKNOWLEDGEMENTS

My sincere gratitude and profound appreciation are due first to my supervisor Dr. Shahin Hashtrudi Zad, for his continuous support, active involvement in the project and valuable guidance. I have been honoured to have his priceless advice and help at every stage of my thesis. I would also like to thank Dr. Amir G. Aghdam for helping me with this project in every possible way.

I would like to extend my gratitude to my thesis committee members, Dr. K. Khorasani, Dr. A. G. Aghdam, Dr. G. Butler, Dr. L. Lopes and late Dr. P. Gohari. I sincerely appreciate the time, support and valuable feedback that I have received throughout my studies.

I am very thankful to my family, colleagues and friends for the tremendous help and support that they have blessed me with. In particular Soodeh, Malakeh, Mahmoud, Shahram and Bahram for your patience and support.

TABLE OF CONTENTS

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF ABBREVIATIONS AND SYMBOLS xv

1 Introduction **1**

1.1 Problem Statement and Motivation 2

1.2 Research Objectives and Methodology 4

1.3 Literature Review 5

1.4 Contribution of the Thesis 10

1.5 Organization of the Thesis 13

1.6 Summary 13

2 Background **14**

2.1 Systems Biology Background 14

2.2 Petri Nets 18

2.3 Modelling Biological Systems 21

2.4 Supervisory Control of Discrete Event Systems 26

2.5 Conclusion 28

3 Modelling Biological Systems Using Petri Nets **29**

3.1 Petri net Model Development For Biological Systems 29

3.1.1 Low-level Model Development For Protein Interactions 29

3.1.2 High-level Model Development For Pathway Knowledge 34

3.2 Conclusion 42

4 Consistency Analysis in Modelling of Biological Systems Using Petri Nets 43

4.1 Modelling Consistency 43

4.2 Biological Case Study 55

4.3 Control Consistency 63

4.4 Conclusion 64

5 An Efficient Framework for Sensitivity Analysis in Petri Net Representation of Biological Systems 65

5.1 Introduction 66

5.2 Sensitivity 68

5.3 Simulation 87

5.4 Conclusion 98

6 Conclusions and Future Work 99

6.1 Conclusions 99

6.2 Future research 100

Bibliography 103

A Simulation Codes 120

A.1 MATLAB Codes 120

A.1.1 Connecting to the KEGG API Web Service 120

A.1.2 Petri net Reachability 123

LIST OF TABLES

3.1 Iterations of *Algorithm 1* for Example 1. 35

3.2 Iterations of *Algorithm 2* for Example 2. 41

LIST OF FIGURES

1.1 Diagram from book by Al-Jazari, (13th century AD) in the Suleymaniye Library [1] 6

2.1 Mechanism of IAP mediation in Caspase Inhibition by *XIAP* [2] 17

2.2 Example of Petri net representation of Caspase cascade in Apoptosis 20

2.3 The PPI network constructed on top of 11000 interactions from [3] involving 2401 proteins from S. Cerevisiae [4]. 22

2.4 A bipartite digraph representing protein interaction and its product. 23

2.5 Examples of Petri nets representing Translocation, Association, Dissociation and Enzymatic activation for four proteins 24

2.6 Examples of Petri nets representing Inhibition and signalling cascade 25

3.1 Algorithm to construct the Petri net model from the pathway 32

3.2 Apoptosis pathway obtained from the KEGG. 33

3.3 Example of a pathway diagram to be converted using *Algorithm 1*. Only one type of interactions (translocation) has been used. 34

3.4 The resulting Petri net in Example 1 36

3.5 Algorithm to obtain high level model from KEGG 39

3.6 Example of an artificial pathway-to-pathway relation diagram to be converted using *Algorithm 2*. 40

3.7 The resulting Petri net in Example 2 41

4.1 Example of a boundary transition. 45

4.2 Consistency 48

4.3 Combined models, G_{lo}, G_{hi} and G_{mi}, with some of the arcs shown. The arcs of G_{mi} are shown in dashed lines. 50

4.4 Petri net example of the high-level and low-level models. The arcs of the middle model are shown using dashed lines. 53

4.5 Petri net example of two sample pathways v and v' 54

4.6 Simbiology Model for Casp3 in Apoptosis. 56

4.7 Part of Apoptosis pathway, and drug delivery mechanism. 57

4.8 High level model of the system depicted in Figure 4.7, constructed from biological knowledge, without reversible effect. 57

4.9 High level model of the system depicted in Figure 4.7, constructed from biological knowledge, with reversible effect. 58

4.10 Low level system with 175 states. 59

4.11 Low level system with 405 states. 60

4.12 High level system with 8 states and $M_{hi}^0 = [3,1,1]$. 61

4.13 High level system with 21 states for $M_{hi}^0 = [6,2,2]$. 62

4.14 High level system with 10 states for $M_{hi}^0 = [3,1,1]$. 62

4.15 High level system with 18 states and $M_{hi}^0 = [3,5,1]$. 63

5.1 Part of Apoptosis pathway, and drug delivery mechanism. 70

5.2 Petri net with 4 places, P_3 is assumed to be the output. 74

5.3 Rigidity measure for the system depicted in Figure 5.2. 75

5.4 Rigidity measure calculated for a linear Petri net with four places shown in Figure 5.2, P_3 is assumed to be the output. 76

5.5 Perturbation 78

5.6 Petri net example with three places 88

5.7 output rigidity measure for 3 places 89

5.8 rigidity with three places 89

5.9 Petri net example with four places 90

5.10 output rigidity measure for 4 places shown in figure 5.9 91

5.11 rigidity with four places 92

5.12 Petri net example with five places and loop. 92

5.13 output rigidity measure for 5 places with loop. 93

5.14 rigidity with five places. 93

5.15 Caspase cascade rigidity measures. 94

5.16 Sensitivity calculated using continuous model with SimBiology in MATLAB 95

5.17 Rigidity for apoptosis calculated using Algorithm 3 96

5.18 Refined rigidity for apoptosis calculated using Algorithm 4 with maximum 5000 states calculated in reachability 96

5.19 Refined rigidity for apoptosis calculated using poisson estimates on full reachability 97

5.20 Distribution of different outputs 97

LIST OF ABBREVIATIONS AND SYMBOLS

LTI	linear time-invariant
FD	finite dimensional
CT	continuous time
DT	discrete time
DES	discrete event systems
PC	piece-wise continuous
PPI	Protein-Protein Interaction
PN	Petri Net
HFPN	Hybrid Functional Petri Net
KEGG	Kyoto Encyclopedia of Genes and Genome
ROC	Region of Convergence
DNA	Deoxyribonucleic acid
VEGF	Vascular endothelial growth factor
BFS	Breadth First Search
pfn	Partial Function
FSM	Finite State Machine
FSA	Finite State Automaton
UV	Ultra Violet
GON	Genomic Object Net
WSDL	Web Service Description Language
XML	Extensible Markup Language

Chapter 1

Introduction

From a control engineering point of view, a cell of an organism can be considered as a dynamical system, for which a proper controller can be designed in such a way that it ensures desirable behaviour. However the first step to design a proper controller would be to model the system with an acceptable accuracy. In an electric motor (as an example of a commonly used control system), the system specification can usually be expressed by a list of about hundred design parameters from speed, torque, coupling type and frequency to mounting type, insulation and shaft. However in the case of a cell, the specifications required for modelling the system typically include over half a million chemical interactions and more than hundred thousand genes. Furthermore, the knowledge of interactions between biological entities is not completely available.

A very useful survey in applications of control in molecular systems biology is presented in [5] that illustrates the basic biological concepts and describes some open questions in dynamics and control of this type of systems. One of the building blocks of biological networks is protein interaction. Various types of interactions are orchestrated to achieve different objectives. Certain subsets of these interactions which construct a subsystem of the metabolic network are known as pathways.

1.1 Problem Statement and Motivation

Due to the large size of the model of the cell, it is obvious that a systematic way of representing the knowledge of the model is highly desirable. Many attempts have been made to capture the dynamics of such systems using differential equations (e.g. [6], [7], [8], [9], [10], [11]). While these models have proven to be very useful, the complexity of the nonlinear model obtained grows rapidly by the number of protein interactions involved. A detailed approach which takes the entire cell dynamics into consideration will require a huge model which is impossible to analyze. Furthermore, constructing such a model requires fine knowledge of many coefficients that are either unavailable or very hard to obtain [8]. One approach to simplify the model with such a scale is to omit the continuous behaviours, and construct a discrete event model (e.g. [12], [13], [14], [15], [16], [17]). Alternatively, to capture more knowledge of the system, the use of hybrid automaton has been suggested as a suitable modelling framework for biological protein regulatory networks (e.g. [18], [19]).

Developing early detection methods, accurate prognosis and effective therapies for numerous diseases such as cancer requires a system level understanding of the cell. It is desired to have a framework that can be effectively used for modelling systems that are knowledge-intensive. Such framework will be useful in analysis of biological systems as well as financial, social and political systems. Furthermore, in order to influence the behaviour of the system and achieve desired outcome, such model will provide a tool for control design.

As a simple example, think about an office, where an office assistant and a manager are working. Both employees will take note of the events in the office. Now suppose that an industrial or control engineer is trying to optimize the functionality of business processes at this office. He will receive two logs of events one by the manager and the other by the office assistant. Both of these logs describe the same system; however most certainly the level of granularity is not the same, neither is the language. While the assistant may have made a note about loading papers in the printer and preparing forms for purchase of new batch of papers, the manager might have just noted approving a purchase. He might have also noted the necessity to write a memo advising everyone to limit their use of the printer, to save on paper budget.

Note that the high level log was not created by summarizing the low level notes. It was rather generated by another observer who had a different perspective. It is desired to study the consistency between the two logs. Also if a management (control) strategy is designed based on the high level logs, it has to be ensured that the implementation in the low level will also achieve desired behaviour. For instance, it has to be determined if sending out a memo to everyone is the best solution for minimizing the consumption of paper, or should the manager just cut access of a single heavy user instead.

The same type of problem can occur in social systems where for instance twitter feeds and status updates on social networks could be an indication of certain major events or reactions of different groups of people. As has been studied in case of flu epidemics [20]. However there are other social indicators which could describe the same system from other perspectives such as purchasing patterns or use of public transportation. The control strategy here could be thought of as advertising campaigns or vaccination sponsorships, or even general alerts.

In the case of biological systems, proteins are the building blocks of the dynamics of the system, and the concentration of proteins defines the state of the system. In such systems a subset of proteins and their respective causal interactions which achieves a coherent functionality in the system (e.g. cell death) is a pathway [21]. In this case protein interaction network is the low level language while pathway dynamics which is a functional approach to the system is the high level. The control strategy here would translate into a drug discovery method that would try to influence the behaviour of the biological system, as has been studied in [22, 23, 24].

1.2 Research Objectives and Methodology

In this thesis a discrete event modelling framework has been proposed that allows efficient representation of knowledge intensive models such as biological systems.

In this research three major issues are dealt with. The first step is development of a model for cell dynamics. The developed model is the basis of all other steps and captures the dynamics of the cell in a useful way. This model is then used for various different knowledge sources that describe the same biological system with different levels of granularity. Hence creating multiple models that describe the system with different languages.

The second objective is to analyze the consistency issues of the models. The goal of this analysis is to identify new pathways or to provide new hypothesis for protein interactions. Despite the wealth of available genomic data, there are still a lot of important molecular interactions that are not yet completely observed. Such models have been used to infer new information regarding the dynamics of the cell that have not been inspected experimentally or have been neglected due to physical obstacles. This analysis would also be used in better understanding of the system behaviour as well as guaranteeing that a controller designed based on an abstracted system could be used for a real system as well.

The third objective of this research is to study methods of designing proper controllers. The controller design aims at avoiding *failure* states or recovery from failure by altering the behaviour of the system. Here undesirable behaviour corresponds to diseases and *recovery* to diagnosis and treatment. Designing a controller for such systems is specifically challenging due to the limitations in the available methods of implementation. In other words there is a certain class of interactions that can be regarded as controllable. The methodology proposed is to conduct sensitivity analysis, which results in proper identification of target elements in the system that can be subject of biological tests for drug discovery and design.

One of the major challenges of this thesis is lack of laboratory data for validation of the models and parameters. It is very difficult to have the concentration of different proteins while the biological entity is still alive (i.e. in-vivo) [25, 26, 27]. In this research publicly verified mathematical models in literature were used to generate data and set parameters [28, 29]. Finally the sensitivity results obtained with this framework are compared to the similar results obtained using continuous models that are already verified in practice.

1.3 Literature Review

Some of the earliest documented works that mention automata date back to 1200 A.D. and the works of a Kurdish scholar Al-Jazari. His manuscript titled “The Book of Knowledge of Ingenious Mechanical Devices” and written in Arabic serves as a historical document on the earliest automata ever used [30].

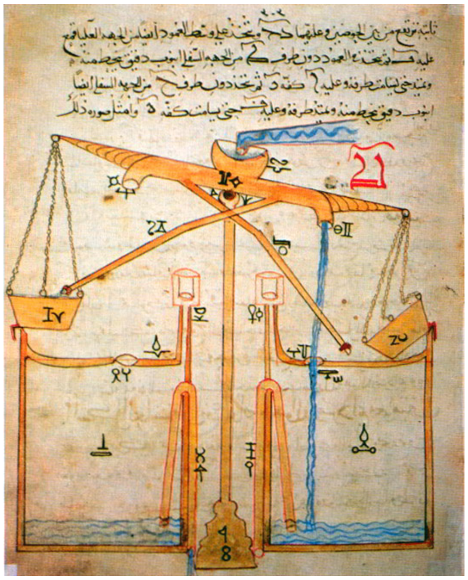


Figure 1.1: Diagram from book by Al-Jazari, (13th century AD) in the Suleymaniye Library [1]

The current literature surrounding the topic of discrete event systems in control can perhaps be traced back to the works of Ramadge and Wonham [31, 32], which follows the Eilenbergs study of automata in 1974 [33] and expands many works done in early 19th century. This extensively helped shape the current state of supervisory control of discrete event systems as is known today. On the other hand the study of Petri nets as a deviation of graph theory was coined by Petri in 1962 [34] as a response to the ever growing need to model industrial manufacturing systems. In the late 70s many researchers including Tadao Murata worked on shaping a powerful literature for Petri net [35]. Perhaps the most important work is Murata's paper [36] that gathers and defines many aspects of what is known today as the literature of Petri nets.

A considerable body of knowledge exists on various tools and techniques for modelling and control using Petri nets [37, 38, 39], which is an extremely valuable asset in any attempt to solve problems and answer questions regarding automation and modelling of discrete event systems. Refinement of Petri nets has been studied in various examples [40]. These methods have a bottom up approach, such that the refinement starts with a complex system and then iterates to an abridged form [41]. Various researchers have studied the problems of verification, or preservation of certain characteristics under such abstraction mechanisms [11, 13, 42].

Petri nets have been specifically used to model chemical systems and their application in modelling biological systems has received a lot of attention [7, 43, 44, 45, 46, 47, 48]. One of the clear advantages of such models is to provide a graphical representation of the system that is easier to understand than differential equations. One major point in this type of modelling is that the quantization of the chemical concentrations can effectively result in a much faster and more efficient way of system analysis.

The study of biological systems using engineering tools and techniques has received a lot of attention in the past few years. More specifically with various projects such as the human genome project, new questions have been asked and are being answered. The collective effort of many engineering and computer science research communities to solve problems in biology is referred to as *Systems Biology* [5, 49, 50, 51, 52, 53, 54, 55, 56, 57]. Many tools have been used for modelling biological systems such as different Algebra [12] and Petri nets [47].

One of the earliest works that uses Petri nets to model metabolic pathways was published in 1993 by Reddy et al. [58]. The authors have used Petri nets to represent pathways of biological systems primarily for qualitative analysis of the structure of the net. They further suggest some ways for simplification of the net to get rid of places and transitions that are not integral in the structure of the Petri net.

This work was followed by other researchers who attempted at applying the same idea to other cases such as gene regulation, biological regulatory networks, etc. [44, 45, 47, 48]. However two main problems constantly persists in the literature. First, the knowledge of biological systems has been limited to biology literature which traditionally lacks the same structure and coherence as engineering literature. This could in part be due to the size of the problem as well as different methodologies in practice. Second, model validation and simulation of the proposed systems has always been a big challenge partly because biological data in general is extremely difficult or expensive to find. In many cases, technology has just recently reached a point of providing high quality, and high throughput data. Nevertheless many attempts have been made to work on quantitative aspects of such models [59, 60, 61].

These challenges have partly resulted in some of the attention to be shifted towards analysis of isolated parts of the system [29, 62, 63], in the hopes that computer science community would be able to create a more holistic approach to structure information in a machine readable way [64, 65].

Limited availability of data has also resulted in many innovative approaches to simulate systems using other methods such as Monte Carlo [66]. A stochastic modelling framework based on stochastic automata networks (SANs) for the analysis of complex biochemical reaction networks has been studied in [55]. This approach takes into account the discrete character of quantities of components (i.e. the individual populations of the involved chemical species) and the inherent probabilistic nature of microscopic molecular collisions. The SAN approach has the advantage of a modular design process making it adequate for abstraction purposes.

In [54] the author has studied biological pathway modelling based on hybrid intelligent systems or soft computing technologies. Intelligent hybrid systems, refers to several related computing methods such as fuzzy logic, neural nets, genetic algorithms, and statistical analysis. Considering biological pathways as complex control systems, which medicine tries to manipulate to achieve desired results, hybrid intelligent systems may provide a useful tool for modelling biological system dynamics and computational exploration of new drug targets.

Apoptosis or the programmed cell death has been extensively studied in the literature, both as a very important biological pathway [67] as well as a desirable example in systems biology literature [68] due to incredible amount of information. Various proteins and elements involved in this pathway such as p53 [69] or Caspase cascade [70] or IAP family of proteins [71, 72, 73, 74, 75, 76] have been extensively studied. In particular XIAP has been identified as a key element in treatment of many diseases such as cancer [77, 78]. This system has been modelled and simulated in SimBiology, a MATLAB library for simulating biological systems [28].

As the size of the model grows, the data available or generated becomes more and more difficult to comprehend. Consequently it becomes inevitable to answer some questions regarding reachability analysis in Petri nets. One of the most important aspects of studying Petri nets is the questions of decidability of various problems in reachability [79]. Many authors have proposed different methods to either avoid reachability analysis or to at least reduce the burden of the complexity in their respective problems [80, 19, 81]. In [82], the authors have studied decidability of various Petri net related problems including reachability of different markings. A very extensive study of this problem is provided in

[83], in which the authors survey complexity and decidability of model checking problems using Petri nets.

The study of sensitivity analysis in ordinary differential equations is nothing new [84]. This analysis has well been extended to hybrid Petri nets as well as Systems Biology [85]. Perhaps the earliest such studies can be traced to [86], where the author speculates on the future direction of control design in discrete event systems using perturbation analysis. An in-depth survey was published on the state of perturbation analysis [87], however the focus is on very simple queue like systems. In [88] authors study sensitivity in hybrid stochastic Petri nets. The study of sensitivity in Petri net models of systems is either focused on stochasticity or continuous analysis in hybrid Petri nets. Sensitivity of parameters in a discrete sense has been mentioned in [89]. This concept has been studied for stochastic Markov chain Petri nets in [90, 91]. Analysis of sensitivity in Petri net models of biological systems had never been studied prior to this thesis, which is mostly due to novelty of the research area.

1.4 Contribution of the Thesis

In this work, a framework has been developed that employs discrete event models to represent biological systems according to the available public domain biology knowledge bases. The discrete event model used here is known as a Petri net, which is in the form of a bipartite digraph, where nodes are grouped in two groups known as places and transitions and edges are called arcs. The dynamics of the Petri net is based on a token assignment function. This function returns the number of tokens at every place of the Petri net at each moment.

Although the Petri net model developed for modelling protein-to-protein interactions is very much easier to analyze compared to a continuous models, it can get very complicated at the level of the whole cell. It will typically require more than half a million places (in the Petri net), just to begin with, and a huge number of transitions.

The developed framework consists of at least two layers of knowledge abstraction from different knowledge sources which are then analyzed for consistency. Note that this can be easily extended to multiple layers. Hence the new high-level model that represents the sequence of pathway activations, rather than individual protein-protein interactions is much smaller and easier to analyze. An algorithm is presented to construct the high-level model from the pathway-to-pathway relation diagrams obtained from public domain knowledge bases.

Next the consistency of the low-level and high-level models is examined. Loosely speaking, the high-level model is consistent with the low-level, if every pathway sequence in the low-level is also possible in the high-level model. A necessary and sufficient condition for modelling consistency is provided. The abstraction in the high-level model is meant to simplify the understanding of the dynamics of the system while maintaining the important characteristics of the low-level fine model. The high-level model may be used in the development of control systems to affect the behaviour of the system. Note that the two high-level and low-level models obtained this way have different sets of events and obviously different languages, and a direct correspondence of the places and transitions are not necessarily available.

An efficient method is then proposed to conduct sensitivity analysis on such discrete-event models which can be extensively used in new drug discovery, along with computer simulation results, to complement or replace actual experiments. Note that these experiments are very expensive in general, can raise ethical issues, and may take very long time to achieve results.

Within the proposed framework the following key contributions have been made.

1. Construction of a multi-layer knowledge representation framework from various biological knowledge sources. This is in contrast with construction of multi-layer systems using abstraction mechanisms, or bottom-up approaches, which in case of knowledge intensive systems, such as biological systems has proven inefficient.

2. Defining consistency and developing an efficient method to evaluate consistency of two models, as well as implementing appropriate MATLAB code library to execute related evaluation tests.

    - Definition of synchronous reachable marking, as a tool for comparison of languages generated by two machines with different alphabet.
    - Definition of activation function, as a tool for mapping markings of high and low level systems.

3. Defining sensitivity in the context of Petri net representation of biological systems.

    - Introducing the notion of rigidity of places of a Petri net that helps measure sensitivity.
    - Developing two efficient methods to calculate the sensitivity without conducting full reachability analysis in a Petri net.
    - Development of MATLAB library code for execution and evaluation of Petri net systems, and visualizing the desired output.

1.5 Organization of the Thesis

The rest of this document is organized as follows. In Chapter 2 the necessary background on Petri nets and the biological knowledge is provided. First an overview of necessary biological background including protein-protein interaction model (PPI) is briefly given. Then formal definition of the Petri net is discussed. Next, some knowledge modelling tools are briefly discussed and the shortcomings of some of the models to describe pathways as well as building blocks of the models are analyzed. In Chapter 3 preliminary results obtained for model construction is presented. A systematic method to construct the Petri net model from a pathway diagram as well as an algorithm to create a high-level abstract model for the system from pathway-to-pathway relation diagrams is shown. In Chapter 4 developments concerning the consistency of abstract and original models are investigated. These results are accompanied by concrete solutions to overcome practical issues with verification of the model, which is done using MATLAB codes and libraries. Chapter 6 investigates the issue of sensitivity in Petri net models of biological systems and presents two methods for efficiently calculating sensitivity. Finally Chapter 5 concludes this thesis and presents some possible future works.

1.6 Summary

This chapter provides an introduction to this thesis by introducing and motivating the problem. The relevant literature has been reviewed, and the contributions of the thesis have been explained.

Chapter 2

Background

This chapter covers background material necessary for the main developments of the thesis. First a brief background on *systems biology* is provided, next Petri nets are formally introduced, and then modelling biological systems is briefly reviewed. Finally this chapter is concluded with a brief overview of control of discrete event systems.

2.1 Systems Biology Background

The main objective of this thesis is to achieve a framework based on discrete-event models for the application of engineering, modelling and analysis techniques to biological systems. While in this thesis the discussion of biological details has been avoided as much as possible, it is necessary for the reader to have a basic understanding of some aspects of cell dynamics. Interested readers are encouraged to consult relevant literature such as [22, 92].

The basic building blocks of the dynamics of a cell are the individual protein to protein interactions. Almost every living organism has a DNA that serves as a genetic blueprint for construction of various proteins. Biologists and bioinformaticians are trying to discover different relationships between more than half a million proteins, as one of the most interesting and important areas of research in the post-genomic era. It is well-known that the state of cell evolves when certain proteins interact with each other and hence form different products which turn out to be proteins too and might involve in some other interactions.

Combination of these sequences of interactions builds a *metabolic network*. While the exact definition of a *pathway* inside a metabolic network is not straightforward [22], it can be basically described as a subset of the network that has a distinguished function. A cell constantly receives and responds to chemical signals from its environment. A signal is initiated when some extracellular signalling molecule (ligand) binds with its respective cell surface receptor. Signalling molecules inside the cell then interact to transduce the signal into cellular responses. Specific collections of interconnected interactions in a network are often referred to as pathways. Signalling pathways often consist of highly complex networks, but it has been discovered that they are usually composed of typical building blocks [22].

As an example let us consider the *Apoptosis* pathway. *Apoptosis* is known as one of the main types of programmed cell death. Hence, it is a process of deliberate life relinquishment by an unwanted cell in a multicellular organism. Any perturbation in the Apoptosis pathway can result in a cancerous cell. In other words, when a cell, due to some DNA damage for example, stops the programmed cell death, it starts reproducing in an uncontrolled way. Recently research on Apoptosis pathway has grown widely. In addition to its importance as a biological phenomenon, defective apoptotic processes have been implicated in an extensive variety of diseases. Too much Apoptosis causes cell-loss disorders, whereas too little results in uncontrolled cell production. The $P53$ tumor antigen is a protein found in a wide variety of transformed cells and is a building block of the Apoptosis pathway [69]. This protein is frequently mutated or inactivated in many types of cancer. *P*53 acts as a tumor suppressor in many tumor types, and hence, its concentration in the cell can be a parameter of interest while finding treatment.

The pathway used for case study in this thesis is the Caspase Apoptosis pathway. Apoptosis is the genetically controlled process of programmed cell death that occurs in multi-cellular organisms. This process is controlled by a diverse range of either extra-cellular (extrinsic) or intra-cellular (intrinsic) signals. Such signals may regulate Apoptosis positively or negatively. [28].

Apoptosis is fundamental in cells' life cycle [67]. It allows the organism to control cell population and to protect itself from morbid cells. In the process of Apoptosis, the nucleus breaks into several discrete chromatin bodies due to the degradation of DNA, and the cell membrane shows irregular buds known as blebs. The cell shrinks due to the breakdown of the cytoskeleton by Caspases, and ultimately the cell breaks apart into several vesicles called Apoptotic bodies, which are then consumed [67].

Mis-regulation or interference in Apoptosis can lead to several diseases, among which are autoimmune diseases, neurodegenerative diseases, such as Alzheimer, Huntington and Parkinson disease, and other diseases as AIDS and cancer [93, 94]. In all these diseases, when the natural procedure of cell Apoptosis is interrupted due to any reason, the number of cells in a tissue start to increase or decrease in an uncontrolled way, the resulting imbalance in the cell population leads to abnormalities in the tissue density [78].

Caspases play an important role in the Apoptosis procedure. Caspases are highly conserved proteins that exist in two types: initiator Caspases among which are Caspase 2, 8, 9, and 10, and effector Caspases such as Caspase 3, 6, and 7 [70]. Initiator Caspases are only activated when binded to specific proteins. When initiator Caspases are activated, they activate effector Caspases. It is these activated effector Caspases that then degrade a host of intracellular proteins to carry out the cell death program [95].

X-linked inhibitor of Apoptosis protein (*XIAP*), also known as inhibitor of Apoptosis protein 3 is a member of the inhibitor of Apoptosis family of proteins (IAP) that stops Apoptotic cell death [96, 71, 75]. *XIAP* stops Apoptotic cell death that is induced either by viral infection or by overproduction of Caspases. It has a baculoviral IAP repeat (BIR) domain [74, 72]. Inhibiting Caspase 3 activity, the BIR2 domain of *XIAP* binds to the active-site substrate groove (where a protein substrate would normally bind during Apoptosis), blocking access of the normal protein substrate that would result in Apoptosis, as a result, *XIAP* inhibits Apoptosis [97]. Figure 2.1 shows the mechanism of IAP mediation in Caspase inhibition by *XIAP* [2, 74].

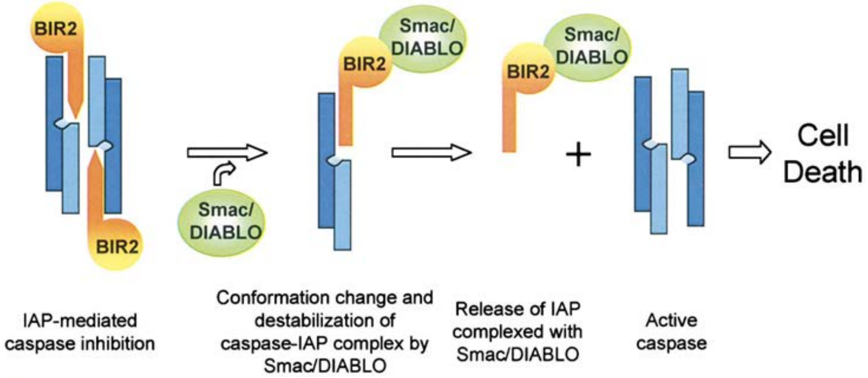

BIR2
BIR2
Smac/ DIABLO
Smac/ DIABLO
BIR2
Smac/ DIABLO
BIR2
+
Cell Death
IAP-mediated caspase inhibition
Conformation change and destabilization of caspase-IAP complex by Smac/DIABLO
Release of IAP complexed with Smac/DIABLO
Active caspase

Figure 2.1: Mechanism of IAP mediation in Caspase Inhibition by *XIAP* [2]

Caspase 3* is a protein that lices certain critical proteins at specific amino acid residuals in the cell. Sequential activation of Caspase 3* plays a central role in the execution-phase of cell Apoptosis. When the cell encounters a trigger, the level of Caspase 3* activation in the cell goes up, which means that the death signal (Apoptotic signal) is directly proportional to the levels of Caspase 3^* activation in the cell. So, in this example, Caspase 3^* can be considered as the output of the system instead of the Apoptotic signal. The goal is to determine which of the other proteins (inputs of the system) has the most effect on the changes in Caspase 3^* levels (and consequently on the Apoptotic signal) [28].

The diseases that effect the cell dynamics, (for example, cancer) can be regarded as *faults* in the system (cell). Cancer is a class of diseases which is primarily characterized by uncontrolled cell division and the ability of these cells to invade other tissues [92]. It may be possible to use a control engineering approach for diagnosis and recovery (i.e., treatment) procedures for these faults. This is one of the main motivations of the research reported in this research and will be discussed further in next section.

2.2 Petri Nets

Knowledge representation using Petri nets has been investigated in the past, e.g., [98], [99]. Reddy et al. [58] were the first to use Petri nets to model biological systems. They related various mathematical properties of the Petri net to the biological properties of pathways [58]. Shortly afterwards, Hofestädt (1994) tried to adapt the Petri net formalism to more adequately represent a biological system [43], and eventually designed a biological Petri net model with simulation capabilities (Hofestädt and Thelen, 1998) [51]. Even today, their work reflects one of the two main goals of Petri net-based modelling methods: to qualitatively study the structure and topology of metabolic pathways or to quantitatively evaluate the dynamic behaviour of biological systems.

In this research, Petri nets are used to model cell dynamics. Petri nets are briefly reviewed in this section. In a Petri net model, the edges are divided into two groups, namely places and transitions. No two places are connected directly to each other; no two transitions are connected to each other either. Each place is assigned a number of tokens. The number of tokens which is also called marking, describes the state of the Petri net. A transition can *fire* only if the places connected to it have enough tokens to overcome the weight of the arcs going from those places to the transitions. When a transition fires, it takes some tokens from its inbound places and puts the tokens into the outbound places. As a result, the state (marking) of the Petri net changes. For a complete survey of the literature on Petri nets the reader is referred to [36] and [38].

Let $\mathbb{Z}^+$ denote the set of nonnegative integers. A labelled Petri net is a six tuple $G = (P, T, F, L, M_0, \Omega)$ where P and T are finite sets of places and transitions, respectively. F is a flow function defined as,

$$F : (P \times T) \cup (T \times P) \rightarrow \mathbb{Z}^+ \tag{2.1}$$

(For $F(x,y) > 0$, there is an arc from x to y denoted by $A(x,y)$ with *multiplicity* $F(x,y)$).

For a transition t, $Pre(t)$ denotes the set of places p with an arc to t ($F(p,t) > 0$). $L : T \rightarrow \Omega$ is a labeling (Labeling attaches an action name to each transition.) The definition of L can also be extended to $L : T^* \rightarrow \Omega^*$, where Ω^* denotes the set of finite sequences of the elements of Ω, including the zero length string ε. For the Petri nets in this work, $T = \Omega$ and L is the identity map. M_0 is an initial marking of the Petri net, where a marking M is a function that gives the number of tokens in each place defined as $M : P \rightarrow \mathbb{Z}^+$. A transition t is enabled at a marking M if for all $p \in P$, $M(p) \geq F(p,t)$, and is denoted by $M \xrightarrow{t}_G$. A transition t from a marking M to a marking M' is denoted by $M \xrightarrow{t}_G M'$, where $M'(p) = M(p) - F(p,t) + F(t,p)$ for all $p \in P$. Also for any $a \in \Omega$, $M \xrightarrow{a}_G$ means $M \xrightarrow{t}_G$ for some t with $L(t) = a$.

The reachability set of a Petri net G is the set of all the states (markings) that can be reached from a specific initial condition M_0 and it is defined as,

$$R_G(M_0) = \{M | \exists \sigma \in T^*, M_0 \overset{\sigma}{\rightarrow}_G M\} \tag{2.2}$$

The reachability can be calculated using different algorithms such as depth first search or breadth first search [100]. A place $p \in P$ is unbounded if,

$$\forall k \in \mathbb{Z}^+ : \exists M \in R_G | M(p) > k. \tag{2.3}$$

A marking M is said to be coverable if there exists a marking M' such that $M'(p) \geq M(p)$ for each p in the net. A Petri net is said to be globally fair if every sequence $\sigma \in L(M_0)$ is finite or every transition on the net can appear infinitely in σ [36].

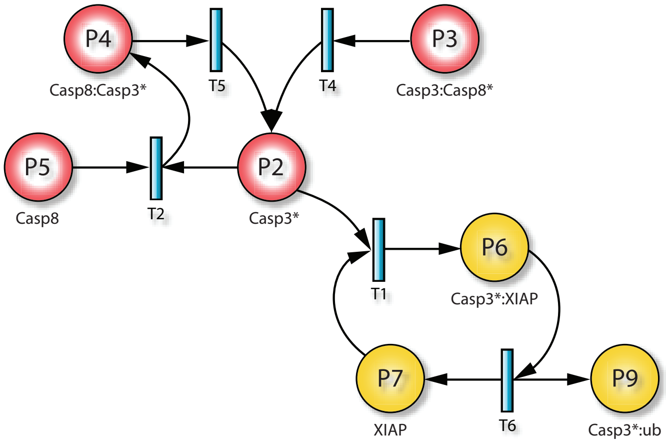


Figure 2.2: Example of Petri net representation of Caspase cascade in Apoptosis

2.3 Modelling Biological Systems

There are many different tools and techniques for representing pathway knowledge; e.g. *π-calculus*, pathway logic, compound graphs and Petri nets [46, 101] .

It was suggested in [102] to use π-calculus for modelling biochemical networks as mobile communication systems. In such a model, molecules and their individual domains are treated as computational processes and their complementary structural and chemical determinants correspond to communication channels [102]. Chemical interaction and subsequent modification coincide with communication and channel transmission. A pathway is defined as a collection of concurrently operating molecules, seen as processes with potential behaviours.

In *Pathway Logic*, the syntax of an action language allows us to specify what the valid states of the world are, how the world evolves from state to state due to execution of actions, and what observations are made about the world [103]. The semantics of action languages often formulate the notions of prediction, planning, explanation, and diagnosis. An action theory is a collection of propositions written in an action language. An action theory can be considered as a model of the world. For example, in the context of cellular modelling, the world corresponds to the cell and a state is a snap-shot of the cell at a particular time.

In order to represent the protein-to-protein relationship, a graph model has been utilized in the literature [104], [105], [106]. A comprehensive survey into the applications of graphs and network characteristics (such as motifs), in biological modelling is available in [107]. In this graph model, each protein or molecule is represented by a vertex of the graph, while the possibility of an interaction between two proteins is shown by edges. In the forthcoming discussions proteins can be regarded as proteins or other molecules whose concentration may play a role in cell dynamics. The graph $G(V,E)$ where V is the set of all proteins, and E is the set of unordered pairs of elements of V, can equivalently be shown using a binary matrix. This network is also referred to as *interactome* [108].

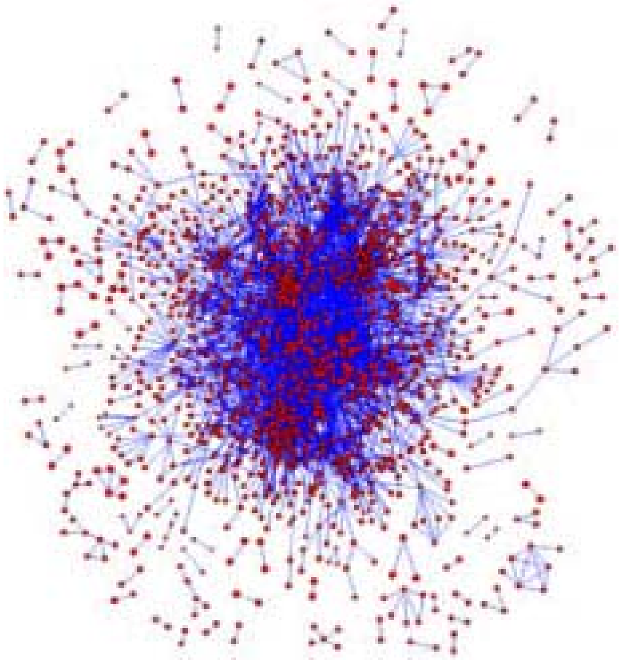


Figure 2.3: The PPI network constructed on top of 11000 interactions from [3] involving 2401 proteins from S. Cerevisiae [4].

The combinatorial interactions among these proteins yield different possible outcomes. While the graph model would present a great insight on investigating and studying different diseases such as cancer, it does not represent any information about the outcome of each individual interaction. In other words the first question would be what is the protein assigned to each edge of the graph G. Hence, nodes are added to the graph model, which represent the interactions. Furthermore the edges need to be directed.

To illustrate the Petri net model of biological systems, suppose that protein A and B can interact and produce the compound AB, there would be two directed arcs from protein A and protein B to interaction T and one arc from interaction T to the product protein AB

(see Figure 2.4). The obtained bi-partite digraph is better known as a Petri net. A formal definition of Petri nets was provided in the previous section.

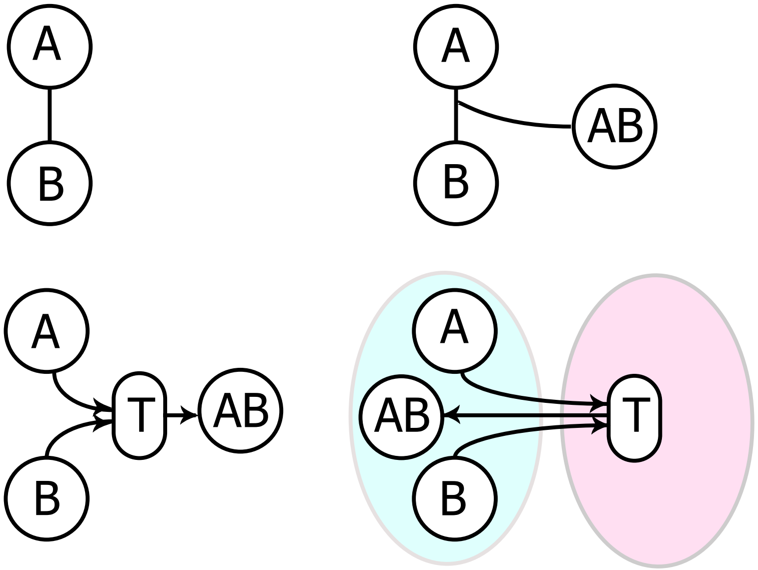
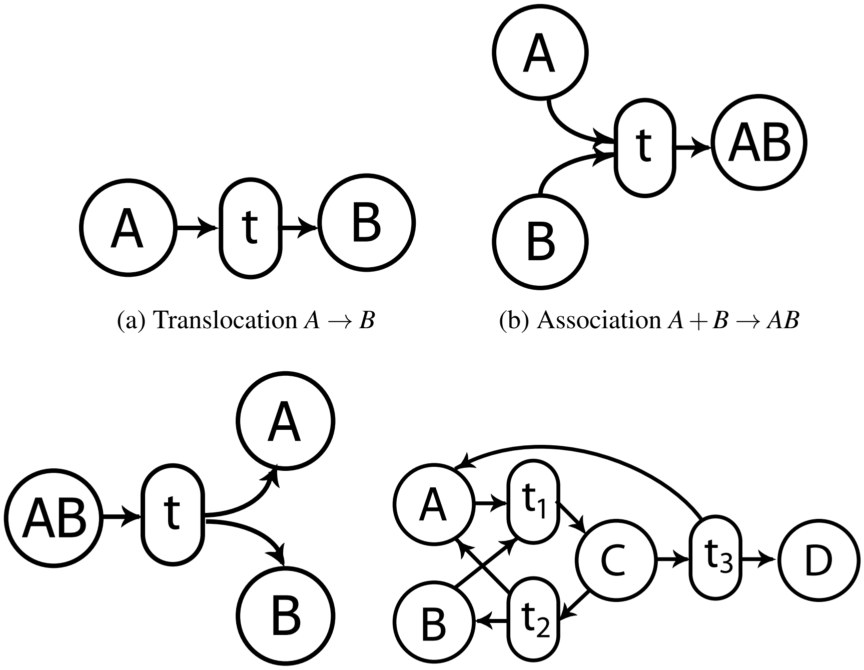


Figure 2.4: A bipartite digraph representing protein interaction and its product.

In the Petri net model as a bipartite directed graph, the node classes represent "proteins" and "reactions", while the edges represent the possibility of proteins to enter into a reaction. In this model, all steps of a pathway with all known details of the underlying reaction mechanisms are represented. In order to generate the model, two classes of nodes have to be constructed. The first class of nodes are the set of places denoted by P, which are all proteins participating in the signalling pathways of interest. The second class is the set of transitions denoted by T modelling interactions. Any interaction type can be fully encoded with a list of substrates and products properly linked, thus explicitly specifying all molecular interactions. Subsequently, they are compiled as a predefined regulatory-interaction and molecular-interaction candidates such as the ones given in Figures 2.5a to 2.6b [109]. Furthermore, tokens in each place will represent the concentration of the respective biological entity represented by that place.

As an example, Figure 2.6a shows an enzymatic activation, where the presence of enzyme *C* in the environment (i.e. tokens in place *C*) facilitates the chemical process from *A* to *B*, and *C* remains intact after the process.



(a) Translocation $A \to B$

(b) Association $A + B \to AB$

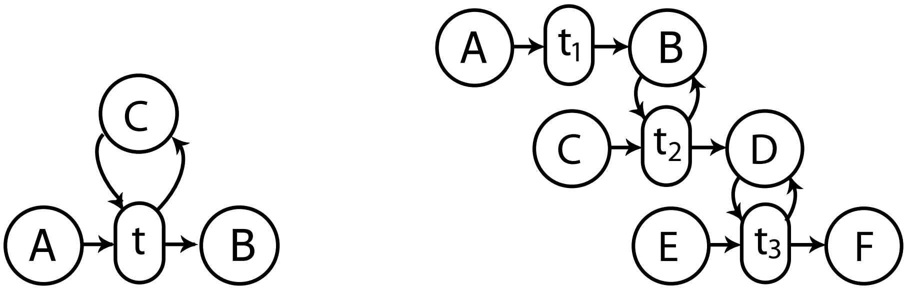
(c) Dissociation $AB \to A + B$

(d) Enzymatic activation (inhibition) $A + B \leftrightarrow C \to D + A$

Figure 2.5: Examples of Petri nets representing Translocation, Association, Dissociation and Enzymatic activation for four proteins

While in some applications or references, the definition of the pathway could be fuzzy, leading to difficulties in using the publicly available biological knowledge [110], in this work, a pathway is an exact set of interactions obtained from public domain biological knowledge-bases such as Kyoto Encyclopedia of Genes and Genomes (KEGG).

KEGG is a bioinformatics resource, developed as part of the research projects in



(a) Inhibition $A \overset{C}{\to} B$

(b) Signaling cascade

Figure 2.6: Examples of Petri nets representing Inhibition and signalling cascade

the Kanehisa Laboratory of Kyoto University Bioinformatics Centre. As an example, the Apoptosis pathway and the role of the protein $P53$ in this pathway, as described in KEGG, can be seen in Figure 3.2. Notice that in order to capture maximum knowledge possible, the pathway is represented by a block diagram with different types of connections. In order to understand different types of interactions and events which are represented using various forms of arcs, a legend table has been provided. This diagram specifies the starting points of the pathway (i.e., cell surface receptors) through other interactions and up to where the pathway completes, and triggers other pathways. A *pathway-to-pathway relationship diagram* showing the causal effect of pathways on each other can also be obtained in terms of tables of consequent pathways as discussed in Chapter 3, Section 3.1.2.

Petri nets were originally designed for modelling concurrent systems. As discussed in the introduction, many extensions of the Petri nets have been used in modelling biological phenomena. Petri nets in their various forms have proven to provide a powerful tool for the purpose of modelling the knowledge of the cell, and hence coming up with its dynamical model [111], [17]. Efforts have been made to represent the biological knowledge using Hybrid functional Petri nets [112], as extensions of functional Petri nets and hybrid object nets, by introducing continuous input and output arcs, and discrete and continuous transitions, as well as simple hybrid Petri net models [59].

While most of these models can be converted to each other, it is to be noted that Petri net modelling offers some advantages such as a well-established literature of supervisory control as well as the rich tools of graph theory for analysis. In this research, Petri net modelling of biological data has been studied.

While Petri nets in their various forms and definitions have been used for representing biological systems, valuable analysis of these models are usually accompanied with other high-level objects. It has been shown that executable high-level net models can lead to several useful observations while low-level models can only result in a part of the desired goal [113]. Therefore, alternative high-level models might become useful at times.

2.4 Supervisory Control of Discrete Event Systems

The control of discrete-event systems (DES) is an active research area, stimulated by the objective of discovering general principles common to a wide range of application domains, such as manufacturing systems, traffic systems, database management systems, communication protocols, and logistic (service) systems. The contributing specialities are notably control, computer and communication science and engineering, together with industrial engineering and operations research. With this variety of supporting disciplines, it is no surprise that the DES research area embraces a large variety of problem types and modelling approaches [114, 115, 116]

In a formal framework, a DES as a dynamic system, is equipped with a state space and a state-transition structure. In particular, a DES is discrete in time and (usually) in state space. The DES model is asynchronous or event-driven, so that it is driven by events other than, or in addition to, the clock ticks. A DES model can also be nondeterministic, by having the capacity of transitional choices by internal chance or other mechanisms not necessarily modelled by the system analyst.

The formal structure of a DES to be controlled is that of a generator (automaton) in the form of a five tuple $G = (Q, \Sigma, \delta, q_0, Q_M)$, where Q is the set of states, Σ is the finite alphabet of symbols which is referred to as event labels. δ is a partial function (pfn) defining the transitions as $\delta : Q \times \Sigma \to Q$, q_0 is the initial state and finally $Q_m \subseteq Q$ is the subset of marked states. In the general sense, it is assumed that the alphabet Σ consists of two disjoint components, namely, $\Sigma = \Sigma_u \cup \Sigma_c$, where Σ_u represents the set of uncontrollable events and Σ_c represents the set of controllable events.

The languages associated with the DES G are the closed behaviour $L(G)$ and the marked behaviour $L_m(G)$ defined as follows.

$$L(G) = \{s \in \Sigma^* | \delta(q_0, s)\text{is defined}\}$$

$$L_m(G) = \{s \in \Sigma^* | \delta(q_0, s) \in Q_m\}$$

In order to enforce a certain behaviour on the language generated by the system, a particular subset of events to be enabled can be selected by specifying a subset of controllable events. Normally all of the uncontrollable events will be added to this subset as these are automatically enabled, and there can be no control over them. Each such subset of events is a control pattern. Naming the set of all control patterns as Γ where $\Gamma \subseteq 2^{\Sigma}$, a supervisory control for G is any map $V : L(G) \to \Gamma$. The pair (G, V) will be written V/G, to show that G is under the supervision of V. The closed behaviour of V/G is denoted to be the language $L(V/G) \subseteq L(G)$ described as follows.

1. $\varepsilon \in L(V/G)$

2. If $s \in L(V/G)$, $\sigma \in V(s)$, and $s\sigma \in L(G)$ then $s\delta \in L(V/G)$

3. no other string belongs to $L(V/G)$.

The theory of supervisory control of discrete event systems then develops the notions of supremal controllable sublanguage and optimal supervision as well as theories of fault detection, isolation and recovery (see, e.g., [116], [117], [118], [119]). The general idea of supervisory control can be applied to the Petri net models as well [38]. Since a bounded Petri net can be easily converted to a finite state automaton (FSA), the results obtained and tools available for automata can be applied to Petri nets as well.

2.5 Conclusion

This chapter provides the basic background that is used in the developments of this thesis. A brief background in biology and Petri net modelling which is required has been covered.

Chapter 3

Modelling Biological Systems Using Petri Nets

This chapter covers the preliminary developments of the thesis. The main results on development of a model for biological systems has been discussed. Also the methodology is applied for obtaining two different Petri nets representing low-level and high-level systems. The aim is to provide a systematic framework for creating discrete-event models for protein-protein interactions and pathway activation sequences based on a public domain knowledge base.

3.1 Petri net Model Development For Biological Systems

3.1.1 Low-level Model Development For Protein Interactions

As discussed in the previous chapter, various pathway diagrams showing protein-protein interactions have been obtained and are available from a number of databases such as KEGG.

Figure 3.2 shows the Apoptosis pathway obtained from KEGG. Note that although the sketch has been manually curated and hence, is highly accurate and readable to human, the representation is not necessarily suitable to be taken as a machine knowledge representation tool.

In this chapter, an algorithm is proposed that can translate a pathway block diagram such as the one in Figure 3.2, to a Petri net model that efficiently describes the pathway knowledge. The resulting Petri net can be understood and updated automatically, and queried by a program. Furthermore, it can be studied using the tools and methods readily available for the analysis of Petri nets.

Assume that all protein interactions are translocations (Figure 2.5a). Other types of reactions can be easily accommodated as discussed later in Remark 1.

Algorithm 1.

Step 1. *Construct a set $\boldsymbol{I}$, consisting of the initial starting proteins in the pathway.*

Step 2. *Set $j = 1$.*

Step 3. *Choose a protein in $\boldsymbol{I}$, call it p_j, and set $\boldsymbol{I} = \boldsymbol{I} - \{p_j\}$.*

Step 4. *Add a place p_j in the Petri net set of places P if it has not already been added to this set.*

Step 5. *Construct $\boldsymbol{L}$, the set of all proteins directly connected to p_j with an incoming arc in the pathway.*

Step 6. *Set $\boldsymbol{I} = \boldsymbol{I} \cup (\boldsymbol{L} - P)$.*

Step 7. *For all p_k in L if $p_k \notin P$, add a place p_k in P.*

Step 8. *For all p_k in L, create transition $t_{j,k}$ and corresponding arcs $A(p_j, t_{j,k})$ and $A(t_{j,k}, p_k)$.*

Step 9. *If $\boldsymbol{I} \neq \emptyset$, then set $j = j+1$ and go to* Step 3*; else stop the algorithm.*

Algorithm 1 starts by constructing a set of initial points, and each time a protein is encountered in the pathway block diagram and added to the Petri net model it gets deleted from this set. In this way, all proteins that have been connected to it will be added to the set. The Algorithm will eventually stop when all proteins and their respective interactions have been examined and added to the Petri net. That is after the set **I** has only one member whose immediate neighbourhood has already been added to the model. Therefore, after *Step 3* the set **I** becomes empty, and in *Step 5*, the set **L** is also empty or contains elements that have already been examined. Hence, in *Step 6*, the union will cause **I** to remain empty and the condition to exit the algorithm will be satisfied. The output of the algorithm is a low-level model G_{lo}. In the resulting Petri net, the tokens in each place would correspond to the concentration of a protein and the transitions would represent protein-to-protein interactions.

Remark 1. Step 8 *of* Algorithm 1 *can easily be modified to: (i) incorporate a case decision maker that constructs the transitions based on a lookup table, for different types of interactions (as shown in Figures 2.5a-2.6b of Sec 2.3); and (ii) set different arc weights based on the stoichiometric rate of different proteins in each interaction.*

Example 1. *Suppose that a simple (artificial) protein interaction network consisting of ten proteins is given as shown in Figure 3.3. It is now desired to use* Algorithm 1 *to construct a Petri net model of the network.*

Table 3.1 shows the iterations of Algorithm 1. *Columns +P and +T show the places and transitions that are being added to the set of places and set of transitions in each*

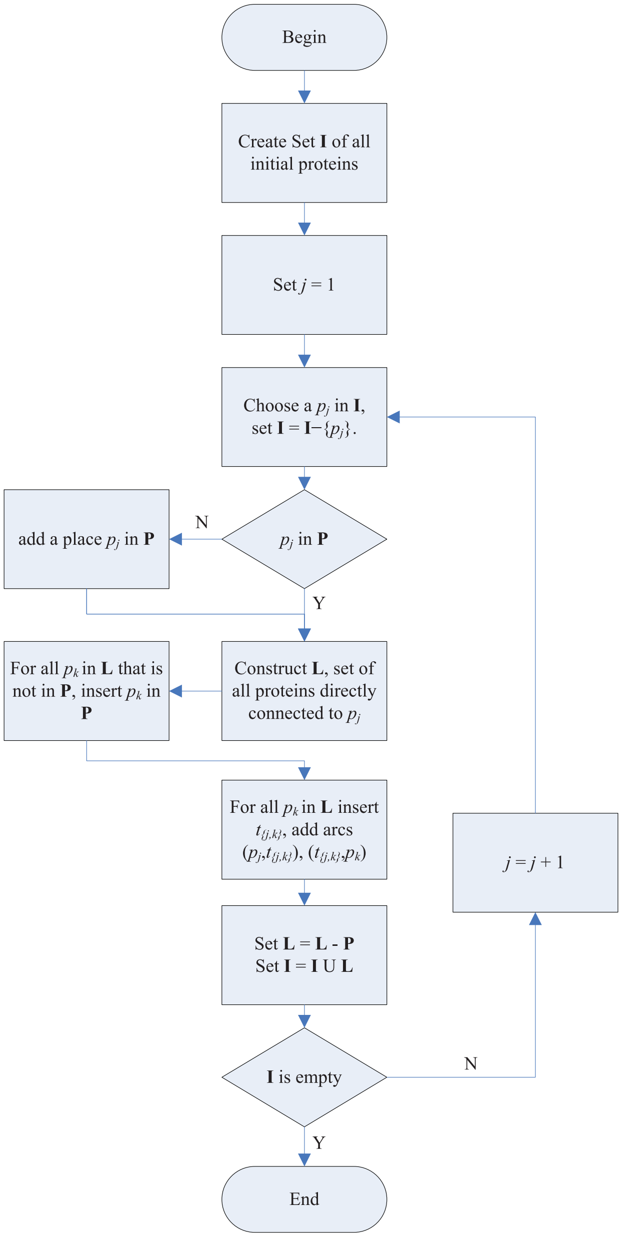

Begin
Create Set **I** of all initial proteins
Set j = 1
Choose a p_j in **I**, set **I** = **I**−{p_j}.
p_j in **P**
N
add a place p_j in **P**
Y
Construct **L**, set of all proteins directly connected to p_j
For all p_k in **L** that is not in **P**, insert p_k in **P**
For all p_k in **L** insert $t_{\{j,k\}}$, add arcs (p_j,$t_{\{j,k\}}$), ($t_{\{j,k\}}$,p_k)
Set **L** = **L** - **P**
Set **I** = **I** U **L**
I is empty
N
Y
j = j + 1
End

Figure 3.1: Algorithm to construct the Petri net model from the pathway

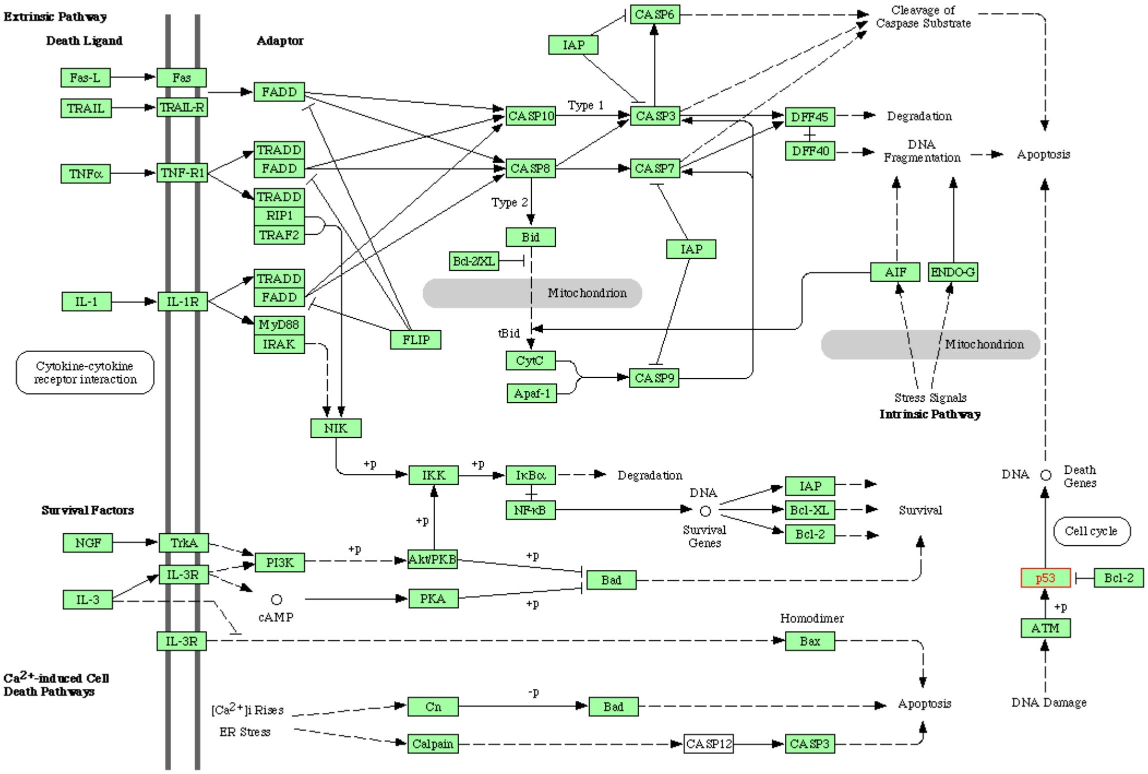


Figure 3.2: Apoptosis pathway obtained from the KEGG.

iteration, respectively. Also, the index that will be omitted from the set I at each iteration has been crossed out.

The resulting Petri net is depicted in Figure 3.4. Notice that in a real example, the names of proteins will most probably be stored in a database, and instead of protein names, object ids will be used.

Following *Algorithm 1*, Petri net models can be constructed for various pathways, which in turn, can be combined to form a large model. This model can be used in the analysis of cell behavior or possibly be used in the development of supervisors to enforce

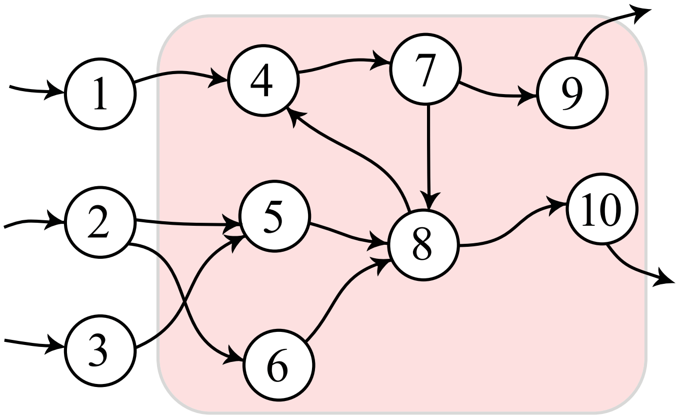


Figure 3.3: Example of a pathway diagram to be converted using *Algorithm 1*. Only one type of interactions (translocation) has been used.

a desired behavior (e.g., activation of the desirable pathways and disablement of the undesirable ones). Even though this model is extremely useful, the number of places can get very large, as we add different proteins to the model. Therefore, it is desirable to construct a more abstract compact model that, for instance, describes cell dynamics at the pathway level. A pathway level of abstraction is particularly useful since it is not uncommon for the desired cell behavior (control specification) to be expressed in terms of pathways, rather than the individual proteins.

3.1.2 High-level Model Development For Pathway Knowledge

As can be easily observed, the Petri net obtained by examining all pathways with proteins involved in them would make a large model that would require a huge effort for analysis. Therefore, it is highly desirable that an abstracted model be derived which preserves certain characteristics of the original one and can be studied easier and with less computational effort. As discussed in [120], reduction is defined as a validity-preserving transformation

j	I	L	+P	+T
1	$\{\cancel{1},2,3\}$	$\{4\}$	p_1,p_4	$t_{1,4}$
2	$\{\cancel{2},3,4\}$	$\{5,6\}$	p_2,p_5,p_6	$t_{2,5},t_{2,6}$
3	$\{\cancel{3},4,5,6\}$	$\{5\}$	p_3	$t_{3,5}$
4	$\{\cancel{4},5,6\}$	$\{7\}$	p_7	$t_{4,7}$
5	$\{\cancel{5},6,7\}$	$\{8\}$	p_8	$t_{5,8}$
6	$\{\cancel{6},7,8\}$	$\{8\}$	-	$t_{6,8}$
7	$\{\cancel{7},8\}$	$\{8,9\}$	p_9	$t_{7,8},t_{7,9}$
8	$\{\cancel{8},9\}$	$\{4,10\}$	p_{10}	$t_{8,4},t_{8,10}$
9	$\{\cancel{9},10\}$	-	-	-
10	$\{\cancel{10}\}$	-	-	-

Table 3.1: Iterations of *Algorithm 1* for Example 1.

that reduces a complex assertion into a much simpler one in such a way that the validity of the latter implies the validity of the former. Pathway modelling using Petri nets is briefly discussed in [109].

In order to proceed with the abstraction, it is desired to identify the subsystems within the obtained model. The approach presented here considers each pathway as a subsystem with a different functionality. However, instead of directly reducing the previously obtained low-level model, the abstract (high-level) model is derived using the *pathway-to-pathway relation diagram* available in databases such as KEGG. For instance, in KEGG the list of all pathways triggered by each pathway is provided. In the case of the Apoptosis pathway, for example, the list is as follows:

1. Cell communication

2. Neurodegenerative disorders

3. Calcium signaling pathway

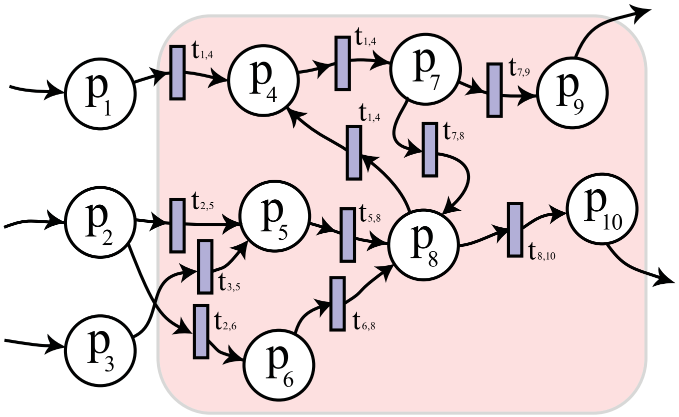


Figure 3.4: The resulting Petri net in Example 1

4. Cell cycle
5. Insulin signaling pathway
6. Type II diabetes mellitus
7. Type I diabetes mellitus
8. Alzheimer's disease
9. Parkinson's disease
10. Pathogenic Escherichia coli infection - EHEC
11. Colorectal cancer
12. Pancreatic cancer

The relation among pathways is represented in the form of a digraph which may very well have loops. Furthermore, the causal effects of the pathways may change the dynamics of the whole system. One advantage of not developing a high-level model from the low-level one is that the consistency of the two models can be examined through a proper comparison method. (Details will be provided in the next section.) Inconsistencies, if any, point to deficiencies or inaccuracies in the information used for setting up the low-level and high-level models and to possible future experimental work necessary to correct the inconsistencies.

In this section, an algorithm is proposed for translating pathway-to-pathway diagrams into Petri net models. In the resulting Petri net, places represent pathways and transitions represent completion of one pathway and preparation of the other. Hence, a pathway is active when there are tokens in the respective place. Notice that there is no specific starting point in the pathway-to-pathway relation diagram, and the algorithm finishes only when all the knowledge has been captured.

Algorithm 2.

Step 1. *Set $j = 1$ and $\boldsymbol{I} = \{p_1\}$.*

Step 2. *Choose a random pathway from $\boldsymbol{I}$ and call it p_j.*

Step 3. *Remove the chosen p_j from $\boldsymbol{I}$.*

Step 4. *Add a place corresponding to p_j in the Petri net set of places P and a transition t_j with an arc going from p_j to t_j.*

Step 5. *From the pathway relation graph, construct a set $\mathbf{L}$ of all the pathways resulting from p_j (i.e., with an incoming arc from p_j) and a set B containing the pathways that result in p_j (i.e. with an arc incoming to p_j).*

Step 6. *Set $\boldsymbol{M} = \boldsymbol{L} \cap P$, and $\boldsymbol{B} = \boldsymbol{B} \cap P$, and $\boldsymbol{L} = \boldsymbol{L} - P$.*

Step 7. *If $\boldsymbol{B} = \emptyset$ then go to* Step 10.

Step 8. *Choose any p_k from* $\boldsymbol{B}$*, and add an arc* $A(t_k, p_j)$.

Step 9. *Remove p_k from* $\boldsymbol{B}$*, and then go to* Step 7.

Step 10. *If* $\boldsymbol{M} = \emptyset$*, then go to* Step 13.

Step 11. *Choose any p_k from* $\boldsymbol{M}$*, and add an arc* $A(t_j, p_k)$.

Step 12. *Remove p_k from* $\boldsymbol{M}$*, and then go to* Step 10.

Step 13. *Set* $\boldsymbol{I} = \boldsymbol{I} \cup \boldsymbol{L}$.

Step 14. *If* $\boldsymbol{I}$ *is not empty, then increment j and go to* Step 2*; otherwise stop the algorithm.*

The algorithm finally stops when all pathways are examined, and the last pathway either results in no other pathways, or all of its succeeding pathways have already been added to P, in which case *Step 6* makes the set **L** empty, and hence, in *Step 13*, the set **I** will be empty and the condition to exit the algorithm is met.

The reduced high-level model G_{hi} obtained from *Algorithm 2* does not consider individual transitions of the involved proteins; however, it describes the overall advancement of the pathways in the cell. Therefore, based on an analysis of the simplified high-level model, one would be able to draw valuable conclusions about the more complex low-level model.

The tokens in the high-level model represent the activation of each pathway, it will be shown in Chapter 4 that the tokens in G_{hi} can be determined based on the tokens in places of G_{lo}.

A high-level model would specifically be useful when designing a controller to alter the cell dynamics (i.e. drug design). Towards this end, the high-level model could be used

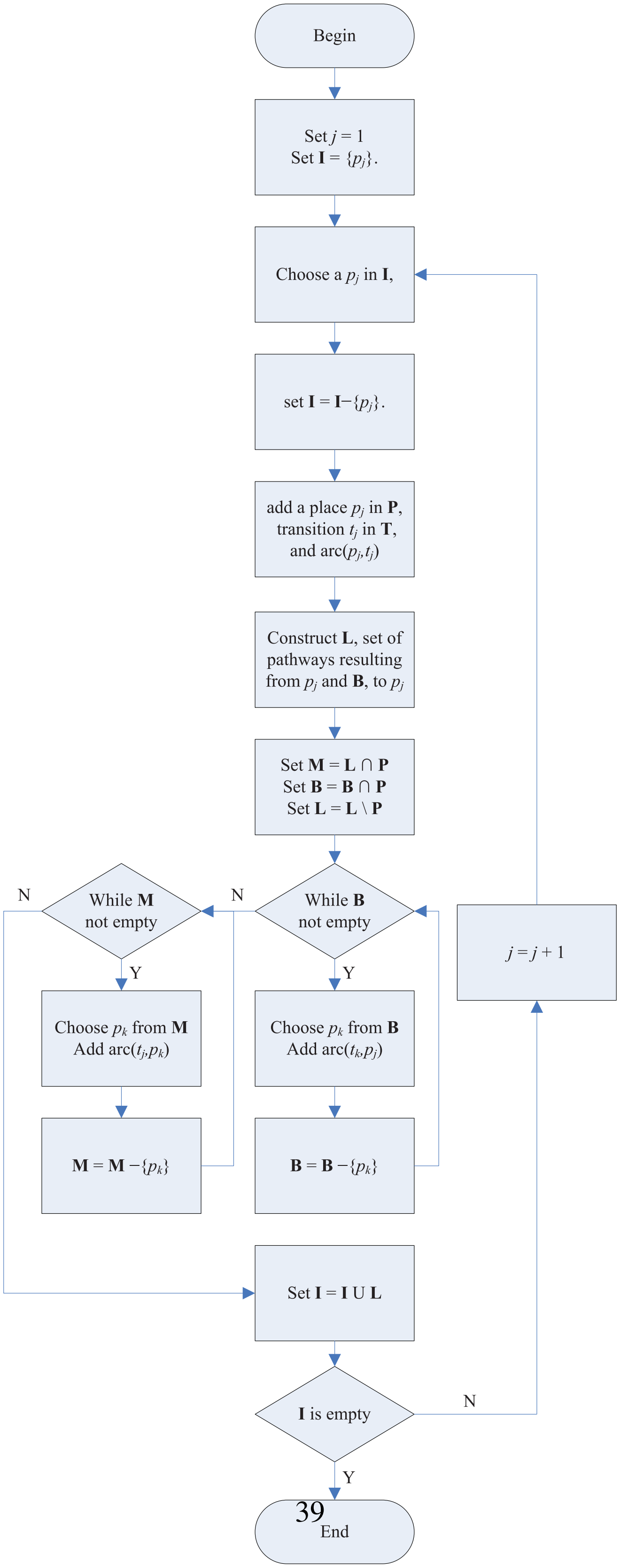

Begin
Set $j = 1$
Set $\mathbf{I} = \{p_j\}$.
Choose a p_j in $\mathbf{I}$,
set $\mathbf{I} = \mathbf{I} - \{p_j\}$.
add a place p_j in $\mathbf{P}$,
transition t_j in $\mathbf{T}$,
and arc(p_j,t_j)
Construct $\mathbf{L}$, set of
pathways resulting
from p_j and $\mathbf{B}$, to p_j
Set $\mathbf{M} = \mathbf{L} \cap \mathbf{P}$
Set $\mathbf{B} = \mathbf{B} \cap \mathbf{P}$
Set $\mathbf{L} = \mathbf{L} \setminus \mathbf{P}$
N
While $\mathbf{M}$
not empty
N
While $\mathbf{B}$
not empty
$j = j + 1$
Y
Y
Choose p_k from $\mathbf{M}$
Add arc(t_j,p_k)
Choose p_k from $\mathbf{B}$
Add arc(t_k,p_j)
$\mathbf{M} = \mathbf{M} - \{p_k\}$
$\mathbf{B} = \mathbf{B} - \{p_k\}$
Set $\mathbf{I} = \mathbf{I}$ U $\mathbf{L}$
$\mathbf{I}$ is empty
N
Y
End

Figure 3.5: Algorithm to obtain high level model from KEGG

to identify the chain of interactions that should be altered to obtain desired behaviour. Finding new drugs could be interpreted as analyzing the system using known control methods of discrete event systems (e.g. [119], [118]) to come up with a set of hypotheses that would suggest what missing interactions facilitate or interrupt a certain pathway.

Example 2. *A simple artificial pathway-to-pathway relation diagram consisting of six pathways is given in Figure 3.6. It is now desired to use* Algorithm 2 *to construct a Petri net model of the network. The resulting Petri net is depicted in Figure 3.7.*

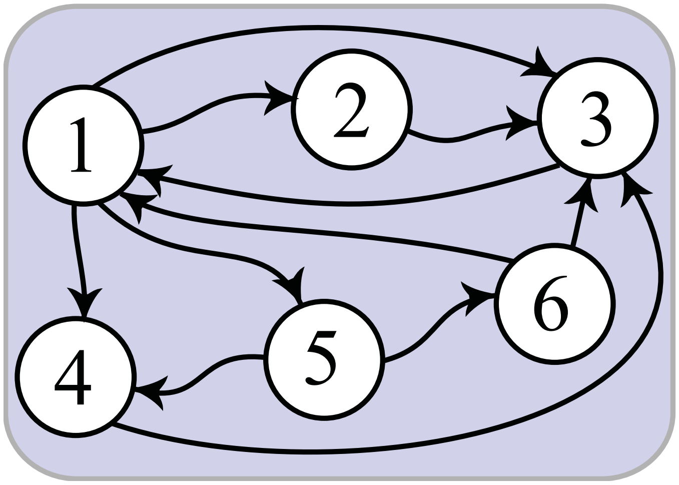


Figure 3.6: Example of an artificial pathway-to-pathway relation diagram to be converted using *Algorithm 2*.

Table 3.2 shows the iterations of Algorithm 2. *Notice that columns +P, +A and +T represent the nodes and arcs that are being added to the Petri net sets. After the sixth iteration, the set I becomes empty, and hence the algorithm stops.*

A sample code to retrieve proteins and interaction from KEGG into MATLAB is provided in Appendix A.1.1. This code uses the WSDL available on KEGG and can be

j	I	L	M	B	+P	+A	+T
1	$\{\cancel{1}\}$	$\{2,3,4,5\}$	$\emptyset$	$\{\cancel{3},\cancel{6}\}$	p_1	$p_1 \rightarrow t_1$	t_1
2	$\{\cancel{2},3,4,5\}$	$\{3\}$	$\emptyset$	$\{1\}$	p_2	$p_2 \rightarrow t_2$ $t_1 \rightarrow p_2$	t_2
3	$\{\cancel{3},4,5\}$	$\{\cancel{1}\}$	$\{1\}$	$\{1,2,\cancel{4},\cancel{6}\}$	p_3	$p_3 \rightarrow t_3$ $t_1 \rightarrow p_3$ $t_2 \rightarrow p_3$ $t_3 \rightarrow p_1$	t_3
4	$\{\cancel{4},5\}$	$\{\cancel{3}\}$	$\{3\}$	$\{1,\cancel{5}\}$	p_4	$p_4 \rightarrow t_4$ $t_1 \rightarrow p_4$ $t_4 \rightarrow p_3$	t_4
5	$\{\cancel{5}\}$	$\{\cancel{4},\cancel{6}\}$	$\{4\}$	$\{1\}$	p_5	$p_5 \rightarrow t_5$ $t_1 \rightarrow p_5$ $t_5 \rightarrow p_4$	t_5
6	$\{\cancel{6}\}$	$\{\cancel{3},\cancel{1}\}$	$\{1,3\}$	$\{5\}$	p_6	$p_6 \rightarrow t_6$ $t_5 \rightarrow p_6$ $t_6 \rightarrow p_1$ $t_6 \rightarrow p_3$	t_6

Table 3.2: Iterations of *Algorithm 2* for Example 2.

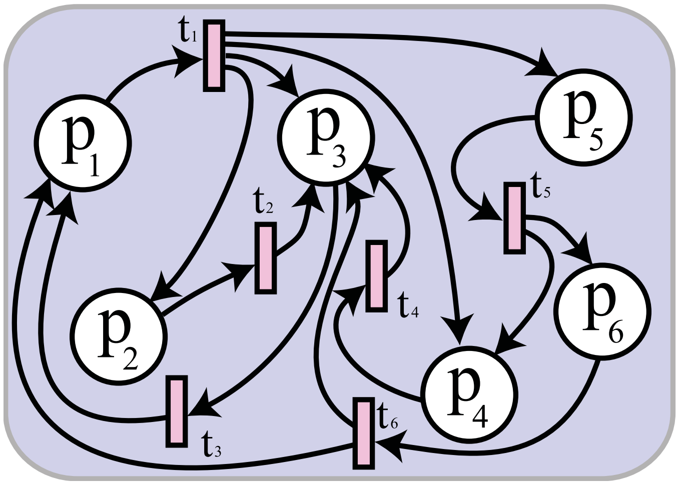


Figure 3.7: The resulting Petri net in Example 2

used to retrieve all desired proteins, pathways and interactions of available organisms from KEGG into MATLAB structures.

3.2 Conclusion

In this chapter two methods have been developed that will allow an efficient use of biological knowledge from publicly available knowledge sources, and construction of Petri net models for a low-level protein interaction and high-level pathway relationship systems. The issue of consistency of such models will be studied in the next chapter.

Chapter 4

Consistency Analysis in Modelling of Biological Systems Using Petri Nets

In this chapter the main developments regarding consistency analysis of multi-level Petri net models of biological systems is provided. Two assumptions have been made to specify weakly branched pathways. A notion of boundary transition has been introduced, as well as pathway activation potential. Consistency of the high-level and low-level models in capturing the dynamics is discussed.

4.1 Modelling Consistency

Suppose the two models G_{hi} and G_{lo} have been obtained from two different knowledge sources for instance following the procedures outlined in the previous chapter. It is now desired to investigate whether the language behaviours of the two models can be directly compared. G_{hi} is an abstract model of the system and captures the information about the sequences of pathway activations in the cell. It is expected, at least, that every sequence of pathway activation that is feasible in G_{lo} be also feasible in G_{hi}. In this thesis, this property is referred to as *modelling consistency*. An important feature of the problem of comparing G_{lo} and G_{hi} is that they have different sets of transitions, and thus, their languages are defined over different alphabets.

First, the notation in previous chapter is slightly modified such that the transitions and places of G_{lo} and G_{hi} are not confused. Let P_{lo}, T_{lo} and P_{hi}, T_{hi} designate the place and transition sets of the low-level and high-level models, respectively. The places in G_{lo} (elements of P_{lo}) are shown by p and those in G_{hi} (which correspond to the pathways) are denoted by v. The low-level transitions are denoted by t and the high-level ones by l. The low-level and high-level markings are represented by M_{lo} and M_{hi}, respectively.

In order to study the consistency of the models, the following relations are defined.

Definition 1. *Let $R_T \subseteq T_{lo} \times P_{hi}$ be a binary relation such that a pathway $v \in P_{hi}$ is in relation with an interaction $t \in T_{lo}$ (written as tR_Tv or $(t,v) \in R_T$) if and only if t can happen while pathway v is active.*

Definition 2. *Let $R_P \subseteq P_{lo} \times P_{hi}$ be a binary relation such that pathway $v \in P_{hi}$ is in relation with $p \in P_{lo}$ (pR_Pv or $(p,v) \in R_P$) if and only if p is a protein that has a role in pathway v.*

Remark 2. *R_T and R_P could be set up according to the info from KEGG.*

Definition 3. *Define the set of* boundary transitions *that can trigger new pathways by T_b:*

$$\begin{aligned} T_b = \{t \in T_{lo} | \exists v_1, v_2 \in P_{hi}, p \in P_{lo}, v_1 \neq v_2 : \\ (t,v_1) \in R_T, (t,v_2) \notin R_T, (p,v_2) \in R_P, F(t,p) \neq 0\}. \end{aligned} \tag{4.1}$$

The following two assumptions are made about the boundary transitions.

Assumption 1. *Each pathway can have at most one boundary transition.*

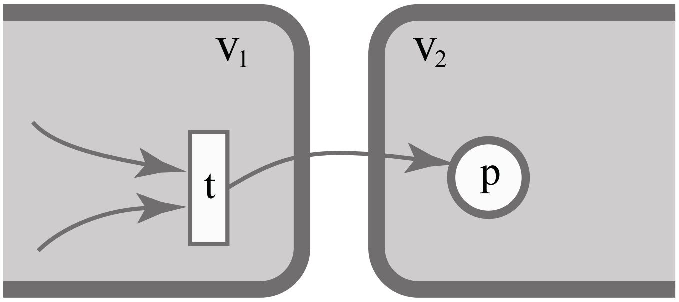


Figure 4.1: Example of a boundary transition.

Suppose Assumption 1 does not hold, and for instance, a pathway v can trigger another pathway such as v'_1 with boundary transition t_1 and pathway v'_2 with boundary transition t_2. In this case, in order to be able to trace pathway activations at the high-level model, the information that v may trigger v'_1 or v'_2 but not both at the same time, should be available in the high-level knowledge source, which may not be true in general (as in the case of pathway-to-pathway relationship diagrams of KEGG). To ensure the assumption remains valid, one may split v into two pathways: one triggering v'_1 and the other v'_2.

Assumption 2. *A boundary transition belongs to one pathway only (pathways do not have common boundary transitions).*

Suppose Assumption 2 does not hold, and for instance, pathways v_1 and v_2 have a common boundary transition t which triggers v'. In order to be able to correctly trace pathway activations at the high-level, the information concerning the combination of the two pathways v_1 and v_2 for triggering v' must be available in the high-level knowledge source. This information may not be available in general (as in the case of the KEGG pathway-to-pathway relationship diagrams). In order to keep the assumption valid, one can combine pathways v_1 and v_2 in a larger section with one boundary transition t.

The marking of a Petri net describes the state of the Petri net (as a dynamic system).

In this case, any marking of G_{lo}, such as M_{lo}, is a possible vector of protein concentrations in the cell at a specific time. The concentration levels in M_{lo} will cause different pathways to become active or inactive. For a given concentration of proteins (i.e. tokens in the places) of the pathway v, the maximum number of times that the pathway v can trigger its boundary transition in the absence of any external transitions (i.e. transitions not belonging to the pathway) depositing tokens in the pathway places, is considered in this work as the number of times the pathway v is activated.

Specifically, let $G_{lo,v}$ denote the sub-Petri net of G_{lo} consisting of places and transitions of pathway v, and $M_{lo,v}$ denote a marking of $G_{lo,v}$. Furthermore, let $L(G_{lo,v}, M_{lo,v})$ denote the language generated by $G_{lo,v}$ starting from $M_{lo,v}$ and t_b be the boundary transition of $G_{lo,v}$. Then the number of times v is considered activated at $M_{lo,v}$ (or simply the *activation* of v at $M_{lo,v}$), is

$$\alpha(v, M_{lo,v}) = \max\{n \in \mathbb{N} | \exists s \in L(G_{lo,v}, M_{lo,v}) : s \text{ contains } n \text{ instances of } t_b\}. \qquad (4.2)$$

The number $\alpha(v, M_{lo,v})$ is, in fact, a measure of the potential of the pathway v to trigger other pathways in the absence of any external transitions. Note that from the definition of α that if as a result of an internal transition, $M_{lo,v}$ becomes $M'_{lo,v}$, then $\alpha(v, M_{lo,v}) \geqslant \alpha(v, M'_{lo,v})$. In other words, in the absence of external transitions, the activation potential does not increase.

At any given time, the set of the activations of all pathways simply translates to a marking for the high-level model. In other words it is possible to define a function $\theta : \mathscr{M}_{lo} \rightarrow \mathscr{M}_{hi}$, where $\mathscr{M}_{lo}$ and $\mathscr{M}_{hi}$ represent the set of all low level and high level reachable markings respectively, and the function θ maps any low-level marking M_{lo} to a high-level marking $M_{hi} = \theta(M_{lo})$ with $M_{hi}(v) = \alpha(v, M_{lo,v})$.

$\theta(M_{lo})$ can be regarded as what the tokens of a high-level model should be, based on the concentrations of proteins and the corresponding state (active/inactive) of the pathways.

Let M^0_{lo} and $M^0_{hi} := \theta(M^0_{lo})$ denote the initial markings of G_{lo} and G_{hi}. M^0_{lo} is the vector of initial protein concentrations, and M^0_{hi} gives the corresponding active pathways.

It is desired now to study the cell behaviour for a variety of initial states (markings). Let Φ^0_{lo} and Φ^0_{hi} denote the set of all initial markings of interest.

In the study of the consistency between the low-level and high-level models, the low-level markings that are reached as a result of boundary transitions play an important role in establishing synchrony between the models.

Definition 4. *A low-level marking M_{lo} is called a* synchronous reachable marking *with respect to the initial marking M^0_{lo} if M_{lo} is either equal to M^0_{lo}, or M_{lo} can be reached from M^0_{lo} through a sequence of transitions in which the last transition is a boundary transition.*

Denote the set of synchronous reachable markings with $R_s(M^0_{lo})$. One can write

$$R_s(M^0_{lo}) = \{M^0_{lo}\} \cup \{M_{lo} | \exists s \in T^*, \quad t_b \in T_b : M^0_{lo} \xrightarrow{st_b}_{G_{lo}} M_{lo}\}. \tag{4.3}$$

Notation 1. *For two low-level markings M_{lo} and M'_{lo}, and a boundary transition $t_b \in T_b$, the notation $M_{lo} \overset{t_b}{\Longrightarrow}_{G_{lo}} M'_{lo}$ is used if either $M_{lo} \xrightarrow{t_b}_{G_{lo}} M'_{lo}$ or $M_{lo} \xrightarrow{t_1,\ldots,t_n t_b}_{G_{lo}} M'_{lo}$, where $t_i \notin T_b (1 \leqslant i \leqslant n)$.*

Therefore, $M_{lo} \overset{t_b}{\Longrightarrow}_{G_{lo}} M'_{lo}$ if M'_{lo} can be reached from M_{lo} through a sequence of transitions in which the last transition is the boundary transition t_b and the rest of the transitions are not boundary transitions.

The notion of *consistency* is formally introduced next.

Definition 5. *G_{hi} is called* consistent *with G_{lo} with respect to an initial marking M_{lo}^0 if for any $M_{lo}, M'_{lo} \in R_s(M_{lo}^0)$, the relation $M_{lo} \xRightarrow{t_b}_{G_{lo}} M'_{lo}$ for some $t_b \in T_b$ implies that (i) there exists a reachable high-level marking M_{hi} with $M_{hi} \geqslant \theta(M_{lo})$, and (ii) for any such high-level marking, there exists another high-level marking M'_{hi}, with $M'_{hi} \geqslant \theta(M'_{lo})$ such that with a single high-level event $M_{hi} \rightarrow_{G_{hi}} M'_{hi}$.*

$$\begin{tikzcd}
M_{hi} \geqslant \theta(M_{lo}) \arrow[r] & M'_{hi} \geqslant \theta(M'_{lo}) \\
M_{lo} \arrow[u] \arrow[r, "t_1,\dots,t_l"] & M'_{lo} \arrow[u]
\end{tikzcd}$$

Figure 4.2: Consistency

Note that $M_{hi} \geqslant \theta(M_{lo})$ means every element of M_{hi} is greater than or equal to the corresponding element in $\theta(M_{lo})$.

Definition 6. *If G_{hi} is consistent with G_{lo} for every initial marking in the set of initial markings Φ_{lo}^0, then G_{hi} is said to be* consistent *with G_{lo}.*

The high-level model is developed to summarize the information about the sequence of activated pathways. Essentially, G_{hi} is consistent with G_{lo} if any sequence of pathway activations at the low-level can happen at the high-level (but not necessarily vice versa). The models will not be consistent if a behaviour (i.e. pathway activation sequence) at the low-level is not captured in the high-level model.

Remark 3. *It follows from Definition 5 that in the low level model a pathway may not be able to fire a boundary transition triggering another pathway, while in the high level the corresponding pathway may have enough tokens to fire the transition to trigger the other pathway. (This may happen when $M'_{hi} > \theta(M'_{lo})$). This will be demonstrated in Example 4.*

It is desired now to develop a procedure for verifying consistency. In the supervisory control framework as introduced by Ramadge and Wonham, relationships between different languages, such as bisimulation, have been investigated [121]. Equivalence and refinement relations based on bisimulation and simulation relations between two DES models look for the pairs of states (Petri net markings) that can be reached after the occurrence of identical strings. However, in the problem discussed here, G_{lo} and G_{hi} have completely different transition sets (i.e. alphabets). To study and verify the property of consistency, a mapping from the sequences of the low-level model to those of the high-level one is needed. This mapping is performed using an intermediate Petri net G_{mi} whose places belong to G_{hi} and transitions are from G_{lo}. Specifically, from the procedure of construction of G_{hi} (Chapter 3, Section 3.1.2), it follows that for every high-level transition t, there exists an input place (pathway) v with an arc from v to t (i.e. $F_{hi}(v,t) \neq 0$).

Definition 7. *G_{mi} is obtained from G_{hi} by simply replacing every transition t in G_{hi} by the boundary transition of its input place (i.e. pathway) v. If P_{mi} and T_{mi} denote the set of places and transitions of G_{mi}, then $P_{mi} = P_{hi}$ and $T_{mi} \subseteq T_b$. Furthermore, the initial marking M^0_{mi} is defined to be M^0_{hi}, which is equal to $\theta(M^0_{lo})$.*

An example of the model G_{mi} and its relation with G_{lo} and G_{hi} can be observed in Figure 4.3.

It is now useful to define the following relation between the languages generated by two models, where natural projection is used as defined in [114].

Definition 8. *Consider two Petri nets G_1 and G_2, defined over alphabets Ω_1 and Ω_2 with $\Omega_1 \subseteq \Omega_2$. The language $L(G_1)$ summarizes (in the sense of abstraction) $L(G_2)$ with respect to alphabet Ω_1 denoted by $L(G_1) \precsim L(G_2)$ if $\Theta_{\Omega_1}(L(G_2)) \subseteq L(G_1)$, where $\Theta_{\Omega_1} : \Omega_2 \to \Omega_1$ is the natural projection of Ω_2 onto Ω_1.*

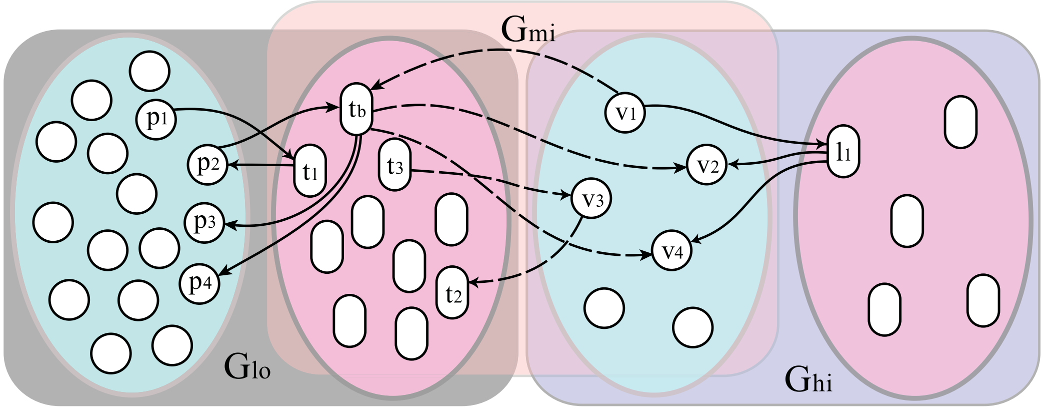


Figure 4.3: Combined models, G_{lo}, G_{hi} and G_{mi}, with some of the arcs shown. The arcs of G_{mi} are shown in dashed lines.

The natural projection Θ drops all characters that do not belong to Ω_1 (i.e., characters in $\Omega_2 - \Omega_1$). In the above definition, model G_1 can be regarded as an abstraction of G_2, providing summary information about the occurrence of the events of Ω_1 only. Note that this abstraction is conservative in the sense that $L(G_1)$ may contain sequences that are not the projection of any sequence in $L(G_2)$.

The following theorem establishes a necessary and sufficient condition for the consistency of G_{hi} with G_{lo}, with respect to a given initial condition M^0_{lo}.

Theorem 1. *G_{hi} is consistent with G_{lo} with respect to an initial marking M^0_{lo} if and only if $L(G_{mi}) \precsim L(G_{lo})$.*

Proof: First assume that the two models are consistent. It is desired to show that $L(G_{mi}) \precsim L(G_{lo})$ or $\Theta_{\Omega_{mi}}(L(G_{lo})) \subseteq L(G_{mi})$.

Suppose that $s \in L(G_{lo})$. If $\Theta_{\Omega_{mi}}(s) = \varepsilon$, then $\Theta_{\Omega_{mi}}(s) \in L(G_{mi})$. If $\Theta_{\Omega_{mi}}(s) \neq \varepsilon$, then let $\Theta_{\Omega_{mi}}(s) = t^1_b \cdots t^m_b$ where $m \geqslant 1$ and $t^i_b \in T_b$. There exist $M^1_{lo}, \cdots, M^m_{lo} \in R_s(M^0_{lo})$ such that

$$M_{lo}^0 \overset{t_b^1}{\Longrightarrow}_{G_{lo}} M_{lo}^1 \overset{t_b^2}{\Longrightarrow}_{G_{lo}} \cdots \overset{t_b^m}{\Longrightarrow}_{G_{lo}} M_{lo}^m$$

By definition $M_{hi}^0 := \theta(M_{lo}^0)$. Thus consistency implies that

$$\exists M_{hi}^1 \geqslant \theta(M_{lo}^1): \quad M_{hi}^0 \rightarrow_{G_{hi}} M_{hi}^1$$

Also,

$$\exists M_{hi}^2 \geqslant \theta(M_{lo}^2): \quad M_{hi}^1 \rightarrow_{G_{hi}} M_{hi}^2$$

Similarly there exists M_{hi}^3 such that $M_{hi}^2 \rightarrow_{G_{hi}} M_{hi}^3$, and so on. Thus,

$$M_{hi}^0 \rightarrow_{G_{hi}} M_{hi}^1 \rightarrow_{G_{hi}} \cdots \rightarrow_{G_{hi}} M_{hi}^m$$

and with $M_{mi}^i := M_{hi}^i$ for $0 \leqslant i \leqslant m$,

$$M_{mi}^0 \xrightarrow{t_b^1} M_{mi}^1 \xrightarrow{t_b^2} \cdots \xrightarrow{t_b^m} M_{mi}^m$$

and hence $t_b^1 \cdots t_b^m \in L(G_{mi})$, which shows that $\Theta_{\Omega_{mi}}(s) \in L(G_{mi})$.

Conversely, it is desired to show that $\Theta_{\Omega_{mi}}(L(G_{lo})) \subseteq L(G_{mi})$ implies consistency. In other words, if for any $s \in L(G_{lo})$, $\Theta_{\Omega_{mi}}(s) \in L(G_{mi})$, then the models should be consistent. Suppose $M_{lo}^0 \overset{t_b}{\Longrightarrow}_{G_{lo}} M_{lo}'$. Then either $M_{lo}^0 \xrightarrow{t_b}_{G_{lo}} M_{lo}'$ or there exists $t_1, \cdots, t_m \notin T_b$ such that $M_{lo}^0 \xrightarrow{t_1 \cdots t_m t_b}_{G_{lo}} M_{lo}'$. This implies $t_b \in \Theta_{\Omega_{mi}}(L(G_{lo}))$, and therefore, $t_b \in L(G_{mi})$. Thus there exists M_{mi}' such that $M_{mi}^0 = \theta(M_{lo}^0) \xrightarrow{t_b}_{G_{mi}} M_{mi}'$. We claim $M_{mi}' \geqslant \theta(M_{lo}')$. Otherwise, there exists a pathway v such that $M_{mi}'(v) < \theta(M_{lo}')(v)$ or in other words, $M_{mi}'(v)$ is less than the activation of the pathway at M_{lo}'. Let t_b' be the boundary transition of v and n the activation of v when G_{lo} is at M'_{lo}. Then $t_b t_b'^{n} \in \Theta_{\Omega_{mi}}(L(G_{lo}))$ but $t_b t_b'^{n} \notin L(G_{mi})$ which contradicts the assumption. Thus we have $M'_{mi} \geqslant \theta(M'_{lo})$. This implies that in G_{hi}, $M^0_{hi} = M^0_{mi} = \theta(M^0_{lo}) \rightarrow_{G_{hi}} M'_{hi} = M'_{mi} \geqslant \theta(M'_{lo})$.

Similarly, if $M_{lo} \in R_s(M^0_{lo})$, then $M^0_{lo} \xrightarrow{st_b}_{G_{lo}} M_{lo}$ and $\Theta_{\Omega_{mi}}(st_b) = t_b^1, \dots, t_b^k$. Therefore $t_b^1, \dots, t_b^k \in L(G_{mi})$ and corresponding to this sequence, there exists a reachable marking M_{mi} such that $M^0_{mi} \xrightarrow{t_b^1, \dots, t_b^k}_{G_{mi}} M_{mi}$. Using an argument similar to that in the previous paragraph, one can show that $M_{mi} \geqslant \theta(M_{lo})$. For any t'_b where $M_{lo} \overset{t'_b}{\Longrightarrow}_{G_{lo}} M'_{lo}$ one can conclude that $t_b^1, \dots, t_b^k, t'_b \in \Theta_{\Omega_{mi}}(L(G_{lo}))$ and therefore $t_1, \dots, t_b, t'_b \in L(G_{mi})$. For any marking such as M_{mi}, there exists another marking M'_{mi} with $M_{mi} \xrightarrow{t'_b}_{G_{mi}} M'_{mi}$. It can be observed that $M'_{mi} \geqslant \theta(M'_{lo})$. Proof can be established by noting that each time a boundary transition fires, tokens in a high-level place will be reduced by one, while the activation in the low level model will be reduced by at least one and possibly more, therefore $M'_{mi} \geqslant \theta(M'_{lo})$. □

Remark 4. *If $L(G_{mi}) \precsim L(G_{lo})$ does not hold, it could be conjectured that for the specific pathways that the relation does not hold, incomplete knowledge has been used in constructing the model. Such cases would be candidates for further lab tests.*

Remark 5. *The condition in Theorem 2 involves projection of languages and can be verified (assuming bounded Petri nets and thus, regular languages), by for example converting the Petri nets to automata and using operations on automata.*

Example 3. *To demonstrate the results, suppose that two pathways $v_1(\{p_1, p_2, p_3\}, \{t_1, t_2\})$ and $v_2(\{p_4, p_5\}, \{t_3\})$ consisting of five proteins and three interactions are defined as depicted in Figure 4.4. As can be observed, the transition t_3 is a boundary transition between pathways v_1 and v_2. An initial marking $M^0_{lo} = [p_1, p_2, p_3, p_4, p_5]^T = [0, 1, 1, 1, 1]^T$ is given for G_{lo}. Hence, $L(G_{lo}) = \overline{\{t_2t_3t_1t_2, t_3t_1, t_2t_2, t_3t_2t_1t_2t_1t_2, t_3t_2t_1t_1t_2t_2\}}$. The initial state of G_{lo}*

will result in the initial state $M_{hi}^0 = [1,0]^T$ for G_{hi}, where pathways v_1 and v_3 are active. Note that the activation of v_2 is zero since v_2 does not have any boundary transition to fire. Observe that $T_{hi} = \{l_1\}$. G_{mi} is the same as G_{hi} with l_1 replaced by t_3. Thus $L(G_{hi}) = \{\varepsilon, t_3\}$. Therefore $\Theta_{\Omega_{mi}}(L(G_{lo})) = \{\varepsilon, t_3\} = L(G_{mi})$, which means G_{hi} is consistent with G_{lo}, with respect to M_{lo}^0. Any string generated by G_{mi} will then have only one t_3, and no more t_3's will appear afterwards. This is consistent with the language generated by G_{lo}.

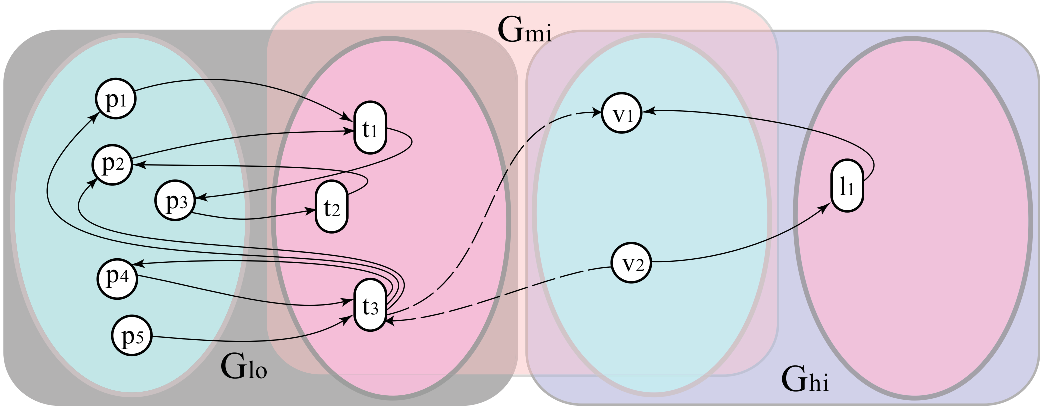


Figure 4.4: Petri net example of the high-level and low-level models. The arcs of the middle model are shown using dashed lines.

Example 4. *Suppose that two pathways v and v' are defined as depicted in Figure 4.5. Pathway v contains three proteins p_1, p_2 and p_3, and three interactions t_1, t_2 and t_3. Transition t_3 is also a boundary transition that activates pathway v'. Pathway v' contains protein p_4 and interaction t_4.*

Assume that the following initial markings are given, $M_{lo,v}^0 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$ and

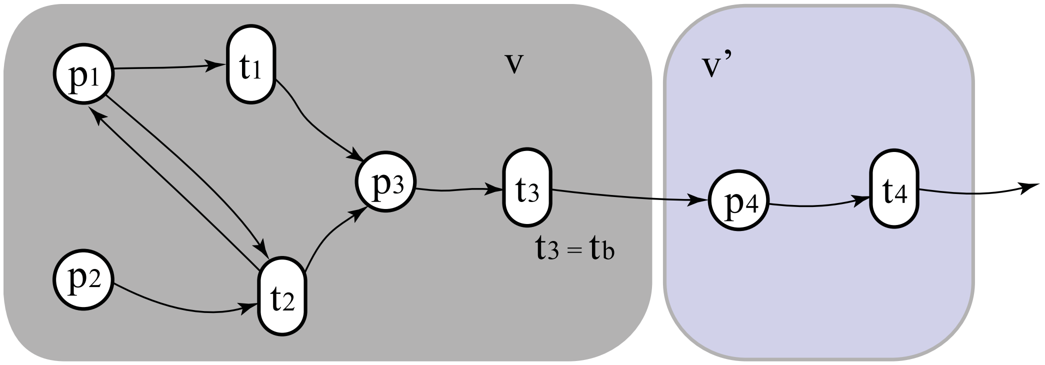


Figure 4.5: Petri net example of two sample pathways v and v'

$M_{lo} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T$. *Then*

$$\begin{aligned} L(G_{lo,v}, M^0_{lo,v}) &= \overline{\{t_1t_3, t_2t_3t_1t_3, t_2t_1t_3t_3\}} \\ L(G_{lo}, M^0_{lo}) &= \overline{\{t_1t_3t_4, t_2t_3t_4t_1t_3t_4, t_2t_3t_1t_4t_3t_4, t_2t_3t_1t_3t_4t_4, t_2t_1t_3t_4t_3t_4, t_2t_1t_3t_3t_4t_4\}} \quad (4.4) \\ \Theta_{\Omega_{mi}} L(G_{lo}, M^0_{lo}) &= \{\varepsilon, t_3, t_3t_3\} \end{aligned}$$

It follows from (5.12) *that* $\alpha(v, M^0_{lo,v}) = 2$, $\alpha(v', M^0_{lo,v'}) = 0$ *and thus* $\theta(M^0_{lo}) = \begin{bmatrix} 2 & 0 \end{bmatrix}^T$

From the perspective of G_{lo}*, the occurrence of the events* t_1t_3 *will drive the initial marking* M^0_{lo} *to* $M^1_{lo} = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T$ *and therefore a transition from* $\theta(M^0_{lo}) = \begin{bmatrix} 2 & 0 \end{bmatrix}^T$ *to* $\theta(M^1_{lo}) = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. *From the perspective of* G_{hi}*, firing the boundary transition only once will change the high-level marking from* $\begin{bmatrix} 2 & 0 \end{bmatrix}^T$ *to* $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$. *This suggests that in the high level model, pathway* v *can still fire while in the low level,* v *will no more be activated. Notice that this does not violate the condition for consistency of the model since consistency requires that the high-level model be able to generate* *all pathway sequences possible in the low-level but not vice versa. To capture more exact dynamics (making $\Theta_{\Omega_{mi}}(L(G_{lo}, M^0_{lo})$ closer to $L(G_{mi}, M^0_{mi}))$ the knowledge source for the high level should be enriched. Meanwhile the language of G_{mi} can be obtained as $L(G_{mi}) = \{\varepsilon, t_3, t_3t_3\}$, which proves consistency between the two models.*

4.2 Biological Case Study

In order to validate the theory developed in this work, a real biological system is studied. The pathway used in this study is the Caspase Apoptosis pathway. Apoptosis is the genetically controlled process of programmed cell death that occurs in multi-cellular organisms. This process is controlled by a diverse range of either extra-cellular (extrinsic) or intra-cellular (intrinsic) signals. Such signals may regulate Apoptosis positively or negatively [67, 74, 78].

Apoptosis is fundamental in cells' life cycle [67]. It allows the organism to control cell population and to protect itself from morbid cells. In the process of Apoptosis, the nucleus breaks into several discrete chromatin bodies due to the degradation of DNA, and the cell membrane shows irregular buds known as blebs. The cell shrinks due to the breakdown of the cytoskeleton by Caspases, and ultimately the cell breaks apart into several vesicles called Apoptotic bodies, which are then consumed [67].

This system has been extensively studied, the equations governing the dynamics of apoptosis can be found in literature, more specifically it has been studied using SimBiology for MATLAB [28]. Figure 4.6 shows this model in SimBiology diagram environment. The parameters of the system are then exported as a matrix that is used to build a Petri net model as depicted in Figure 4.7.

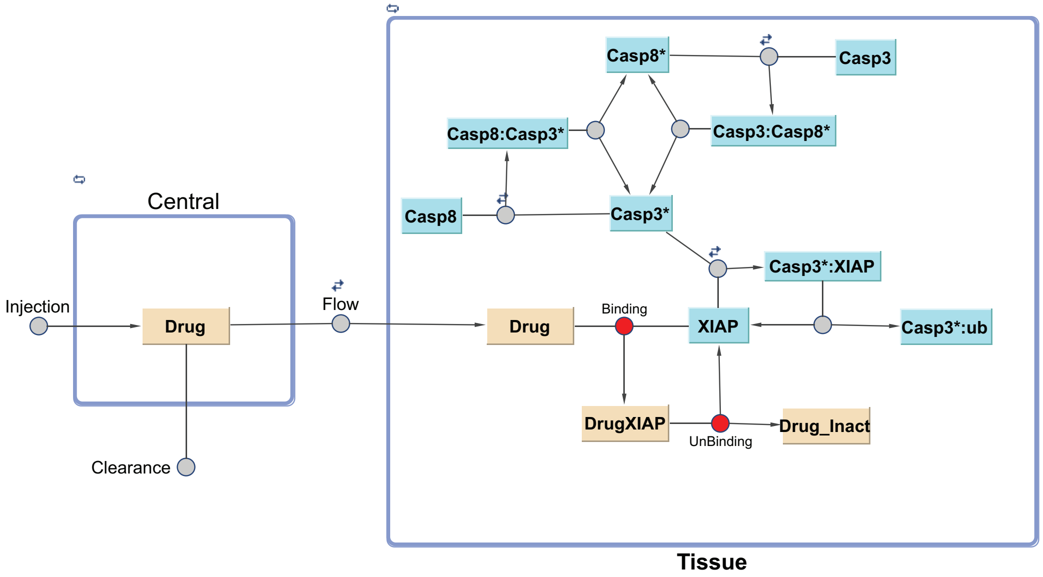


Figure 4.6: Simbiology Model for Casp3 in Apoptosis.

Note that the dynamics of a drug controlling the Caspase cascade is also introduced. In this way three main subnets can be identified, as shown with different colours in the Figure 4.7. These three subnets can be modelled as three separate high-level systems. In this case t_{11} is considered as the boundary transition for the drug dynamics subnet, t_{10} is the boundary transition for the $XIAP$ subnet and t_1 is the boundary transition for the Caspase cascade subnet. Hence the low-level model will consist of all protein interactions that are involved in Caspase cascade regulation system, and the high-level model consists of Caspase cascade as well as $XIAP$ subnet and drug dynamics. Note that both Assumptions 1 and 2 are true in this example.

Figure 4.8 shows a high level model for this system.

It is also possible to consider the reversibility of some interactions which could lead to a high level system as depicted in Figure 4.9.

What is necessary is to choose a reasonable initial condition for the low-level model

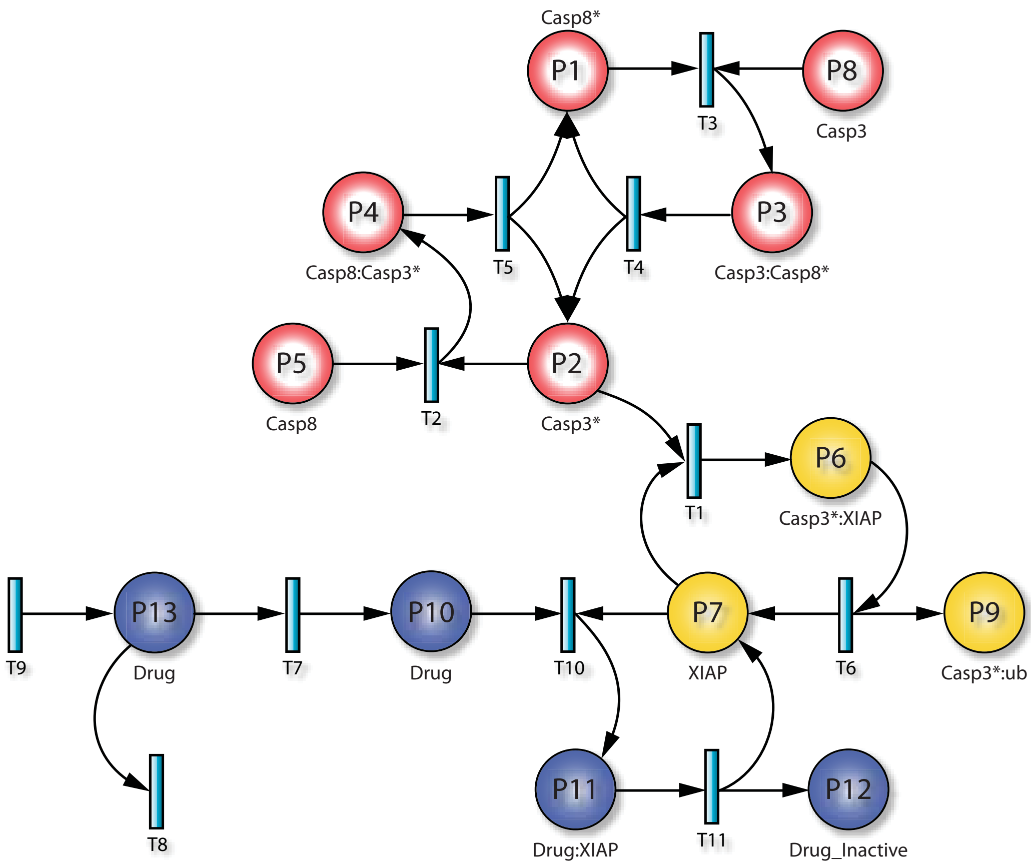


Figure 4.7: Part of Apoptosis pathway, and drug delivery mechanism.

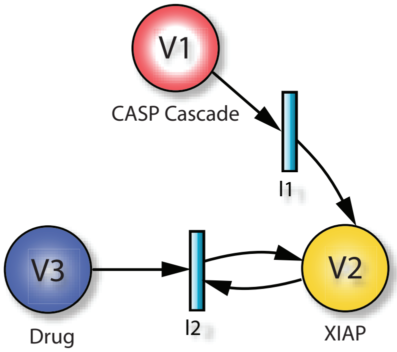


Figure 4.8: High level model of the system depicted in Figure 4.7, constructed from biological knowledge, without reversible effect.

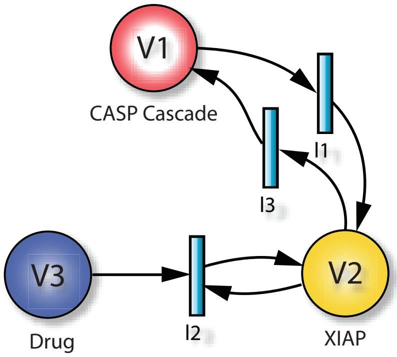


Figure 4.9: High level model of the system depicted in Figure 4.7, constructed from biological knowledge, with reversible effect.

and then generate the state space for the system. In choosing a reasonable initial condition here two things have been taken into account. First using small values, for units of concentration results in many tokens and hence a large state set. At the same time, the tokens should be enough such that every transition gets to fire at least once. Second the relative number of tokens at various places are matched to those of simulations in SimBiology examples. Initially suppose that

$$M_{lo}^{0} = [3,0,0,0,0,0,5,3,0,0,0,0,1].$$

Starting from this initial condition, the state space of the low-level system is calculated and state transition diagram is depicted in Figure 4.10. Note that this rather small initial condition generates 175 states. Let $L\uparrow$ denote an element-wise largest value for the reachable states, and $L\downarrow$ denote the smallest element-wise value for the reachable states. $R(G_{lo}, M_{lo}^{0})$

has the following statistics,

$$L\uparrow(R(G_{lo},M_{lo}^{0})) = [3,3,3,0,0,3,5,3,3,1,1,1,1]$$

$$L\downarrow(R(G_{lo},M_{lo}^{0})) = [0,0,0,0,0,0,1,0,0,0,0,0,0]$$

longest string length $= 15$.

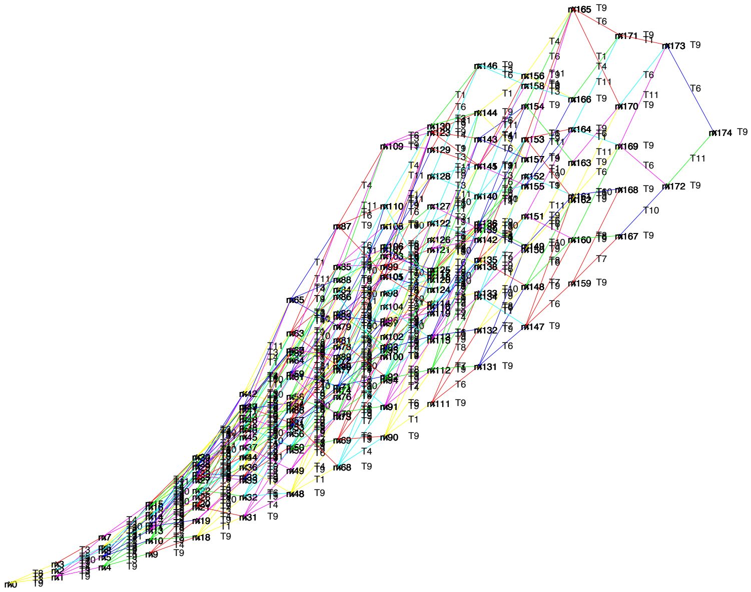


Figure 4.10: Low level system with 175 states.

Next one more token is added to ensure that all transitions in the system will fire at least once. The new initial condition is as follows

$$M_{lo}^{0} = [3,0,0,0,1,0,5,3,0,0,0,0,1].$$

This initial condition will result in 405 states. The state transition diagram is shown in Figure 4.11. Note that as state space grows it gets more difficult to interpret the state diagram of the low-level system. $R(G_{lo},M^0_{lo})$ has the following statistics,

$$L\uparrow(R(G_{lo},M^0_{lo})) = [4,3,3,1,1,3,5,3,3,1,1,1,1]$$

$$L\downarrow(R(G_{lo},M^0_{lo})) = [0,0,0,0,0,0,1,0,0,0,0,0,0]$$

longest string length $= 17$.

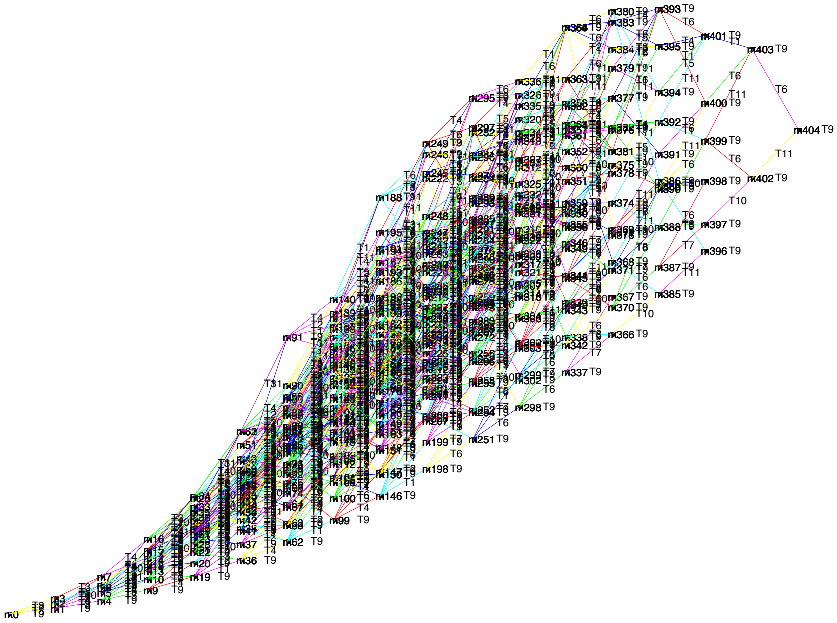


Figure 4.11: Low level system with 405 states.

Next, since $M^0_{hi} = \theta(M^0_{lo})$ with $M^0_{hi}(v) = \alpha(v, M^0_{lo,v})$, the initial condition of the high-level system is obtained as $M^0_{hi} = [3,1,1]$.

The high-level system as depicted in Figure 4.8, generates 8 states as shown in Figure 4.12. Note that for consistency, the high-level model needs to be able to produce the full behaviour of the low-level system and perhaps more. It can be easily verified (i.e. simple projection and meet operation in MATLAB) that both high-level systems do indeed satisfy

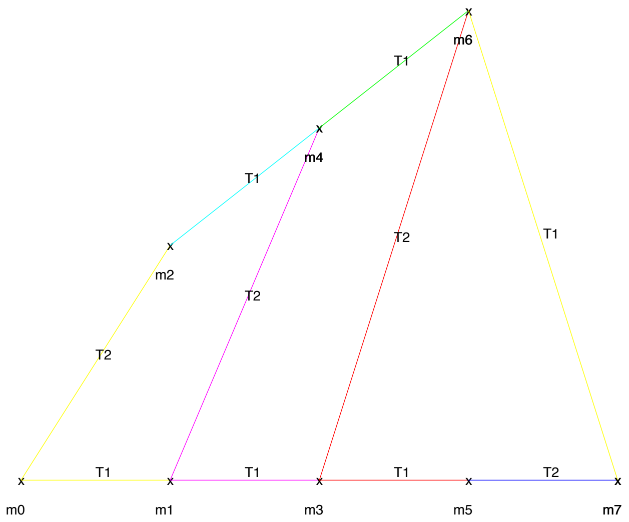


Figure 4.12: High level system with 8 states and $M_{hi}^0 = [3,1,1]$.

consistency. Also if the low-level initial condition M_{lo}^0 increases, resulting in an increase in M_{hi}^0 since $M_{hi}(v) = \alpha(v, M_{lo,v})$, still consistency is satisfied. The results for $M_{hi}^0 = [6,2,2]$ is shown in Figure 4.13.

Now suppose that the high-level system is modelled as in Figure 4.9. Starting from the same initial condition $M_{hi}^0 = [3,1,1]$, the output of the system is then obtained and shown in Figure 4.14. The black dotted lines represent a transition back to a previous state on the left, as l_1 and l_3 are in reverse directions. The high-level system is again consistent with the level system with respect to the initial condition M_{lo}^0.

Note that adding tokens to a place such as V_2 will still result in consistency, in which l_2 occurs only once in any string generated and l_1 can occur many times in any string generated by the system.

It can be observed that the high-level models are much easier to work with and generate much fewer states.

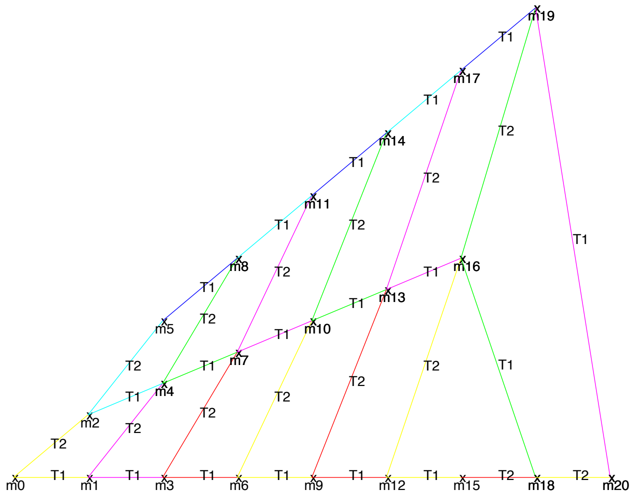


Figure 4.13: High level system with 21 states for $M_{hi}^{0} = [6,2,2]$.

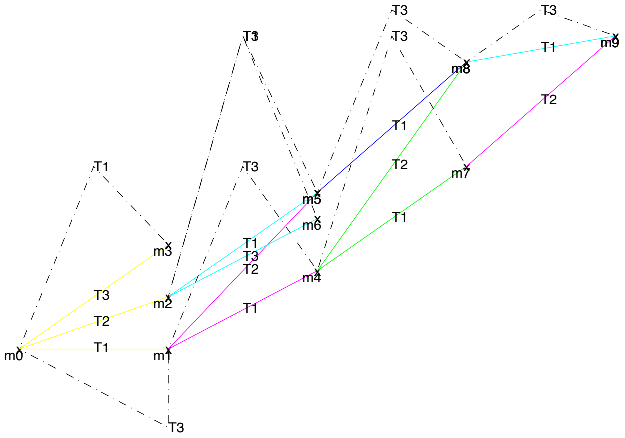


Figure 4.14: High level system with 10 states for $M_{hi}^{0} = [3,1,1]$.

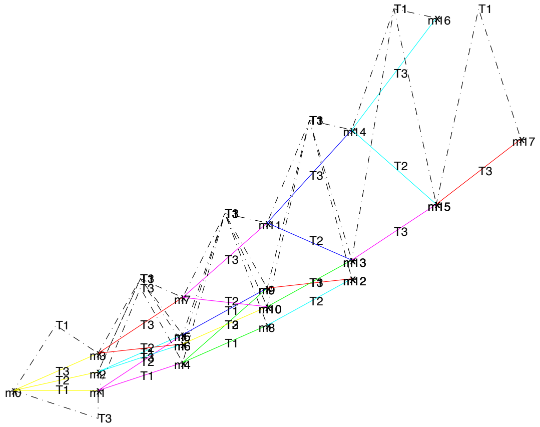


Figure 4.15: High level system with 18 states and $M_{hi}^{0} = [3,5,1]$.

4.3 Control Consistency

The main goal for the development of the abstract model was to avoid working on the low-level model directly. Therefor G_{hi} is the model that will be analyzed for the control design. However the ultimate control implementation will have to be in terms of individual protein interactions which is a language only recognized by G_{lo}. Some earlier work has been done on the subject of consistency in supervisory control of hierarchical models [122], however no research has been done on the subject of consistency in multi-layer non-hierarchical systems.

While the control specification (i.e. desired cell dynamics) is presented in terms of $L(G_{hi})$ (i.e. sequence of pathway activations), in order to be able to implement the control law on G_{lo} it is necessary to find at least one $t_i \in T_{lo}$ for each controllable $l_j \in T_{hi}$ such that disabling t_i will disable l_j and only l_j. Therefore, there should be at least one $v_i \in Pre_{hi}(l_j)$

where $M_{hi}(v_i) < F_{hi}(v_i, l_j)$ can be achieved, by disabling t_i. That is for all $s_i \in \ell(G_{lo})$ such that $C(L(t_i), s_i) = 0$, where $C(\sigma, S)$ counts the number of σ's in S, v_i should be bounded in G_{mi} and $M_{mi}(v_i) < F_{hi}(v_i, t_j)$. The development of control strategies for cell dynamics, is the motivation for the next chapter. This development is the subject for future work.

This condition will guarantee that, when the high-level controller is disabling an action, the control command can actually be implemented in the system.

4.4 Conclusion

In the post genomic era, the most important area of research in systems biology is to design suitable models and tools that can efficiently use the available knowledge and come up with useful observations and decisions for treatment of various diseases. Modelling the cell as a discrete event system helps reduce the complexity of the analysis and synthesis (control) problems, while capturing the essential knowledge. However, due to many factors such as the size of the problem and the uncertainty in the knowledge, the existing techniques for the study of discrete-event system need to be extended. In this chapter, a method is presented to verify whether an abstract model can replace a large low-level model (for the purpose of studying pathway sequences) by checking a modelling consistency property. These models may be used for developing control strategies for altering cell dynamics.

Chapter 5

An Efficient Framework for Sensitivity Analysis in Petri Net Representation of Biological Systems

Emergence of knowledge-intensive models in various different fields such as finance, sociology and biology has resulted in new challenges for modelling, analysis and control of such data-driven systems. The complexity and size of such systems are unprecedented and requires new tools and frameworks. In particular, new drug design, diagnosis and prognosis of disease in biological systems is a subject of research that can well be studied using methodologies in control theory. In this chapter sensitivity analysis in a Petri net representation of biological systems is studied. A theory is developed for efficient calculation of a rigidity measure as an indication of sensitivity. The proposed method is refined to avoid reachability analysis as much as possible in order to be computationally efficient. It is also shown using simulation that the measure accurately achieves identifying the best candidate for drug design target in a real example of Caspase cascade as part of Apoptosis pathway.

5.1 Introduction

Systems biology is a growing inter-disciplinary field of study that focuses on the complex interactions within biological systems. One of the main concerns of systems biology is to discover how such interactions function and affect the behaviour of biological systems (e.g. enzymes in metabolic networks) by using theoretical, analytical or computational modelling tools. The goal is to propose specific testable hypotheses about a biological system, validate them experimentally, and then refine the computational model or theory. With such a goal, various experimental and analysis techniques from different disciplines are employed to advance the research in systems biology [52, 57, 123].

One of the main focuses of systems biology concentrates on modelling protein pathways and using systems theories and tools to study the emergent behaviour and properties of such pathways. Petri nets are one of the powerful mathematical modelling languages that are widely used for studying biological pathways due to their graph-based structure and the fact that they allow formal and transparent representation of the system based on their firm mathematical foundation. One of the reasons that makes Petri nets popular in systems biology is that they provide a balance between modelling power and analyzability: many properties can be automatically determined for Petri nets, some of which are very expensive to determine in the general cases, such as reachability, boundedness and liveness. [16, 44, 50, 51, 113].

Biological models do not necessarily provide a straightforward understanding of the relationship between input (the variable whose change leads to a change in another dependent variable, i.e. the output) and output (the dependent variable) of the model due to the complexity of their structures. Understanding how the model behaves in response to changes in its inputs is of fundamental importance [124]. One of the important tools used in analysis and design optimization of a system is sensitivity analysis. Sensitivity analysis is defined as how changes in the output of a model (system) can be affected by changes in inputs of the model. [49, 57, 125].

Considering the importance of biological processes such as Apoptosis, it is of great value to model and analyze their underlying pathways and try to understand the behaviours that arise from the structural properties of such pathways. This information can later be used in design of control mechanisms with the aim of finding drugs to regulate the behaviour of these processes.

A central aim of systems biology is to study the complex dynamic structure of biological systems. One of the main endeavours in this area is to construct models and theories that are dynamically explanatory, rather than purely descriptive. It is the intention of this thesis to propose a method to enable calculating the sensitivity of a specific quantity of a biological network regarded as the output of the system to other different parameters of the network regarded as inputs of the system. Using this information, one might be able to design a control mechanism to influence the behaviour of the system using the input that has the highest effect on the output.

The main concerns in choosing a method for computing sensitivities are its accuracy, computational expense, and ease of implementation [126] .

In this research, Petri net modelling framework has been used to calculate a sensitivity measure for biological systems. The rest of this chapter is organized as follows.

The proposed sensitivity measure and the logic behind it, is introduced in section 5.2. The computational results corresponding to the application of sensitivity analysis to the Apoptotic pathway is presented and compared with previous results in section 5.3. Finally, in section 5.4, the chapter concludes by a summary of the points and an outlook on future research directions.

5.2 Sensitivity

One of the main questions that needs to be answered after providing a Petri net model for a biological system, is how would it be possible to influence the system to achieve a desired behaviour. In other words how can a control strategy be implemented. In order to answer this question, it is necessary to first define inputs and outputs so that control objective can be described.

The biological systems are naturally stable systems. States are always bounded, and multiple feedback loops regulate the whole system and ensure that biological entities maintain homeostasis. In case of a malfunction of one or a series of such regulatory mechanisms a disease occurs, which would require a certain drug to force the system back to the normal behaviour.

In order to help design such drug, it is required to know what node in the protein network can have a higher influence on the desired output protein or proteins. Hence the input and outputs of the system will be proteins or places in the Petri net model. The question of a better influence translates into a sensitivity analysis in such networks. The researcher would like to know that perturbing which protein can result in a better outcome in the desired output protein concentration. This will offer a possible target or targets for drug design.

In the example studied here, it is desired to control the rate of Caspase cascade. More precisely the activation rate of the cascade must be kept in healthy levels by controlling the rate of its consumption. Therefore the problem is to find which molecule in the network influences casp3 or casp3* the most. That protein will be the target of drug design. This drug will control the Caspase cascade and hence satisfy the control objective. Note that the design of the controller is a completely different task. For the purpose of this research, sensitivity is defined as the range over which the tokens representing a protein would fluctuate given certain excitation at the initial condition at which the Petri net starts. Note that this measure is relative, so for each desired output, the measure depends on which places in the initial condition are excited.

Starting from any given initial condition M_0, the Petri net G will create a reachability set such as $R_G(M_0)$. The desired output p_o will have different token values at different states in $R_G(M_0)$. Suppose that a counter is observing these values, and each time p_o reaches a value j, it adds one to the count of value j.

Definition 9. *$V(M_0, p_o)$ or $V_{p_o}^{M_0}$ is a vector $[v_0, ..., v_k]$, where v_j is the number of times p_o has j tokens starting from the initial condition M_0, and is called the frequency vector or signal of output p_o.*

Definition 9 provides a count on distinct values of the number of tokens in place p_o representing some protein A in $R_G(M_0)$. If a change in the initial condition to M_0' would have little effect on the desired output place, then in the reachability graph, there will be many states with the same value for the output, and $V_{p_o}^{M_0}$ and $V_{p_o}^{M_0'}$ would be very similar. This will result in a narrow frequency vector, (i.e. $V_{p_o}^{M_0}$ will have few very large elements and all other elements will be very small). For the purpose of sensitivity analysis it is

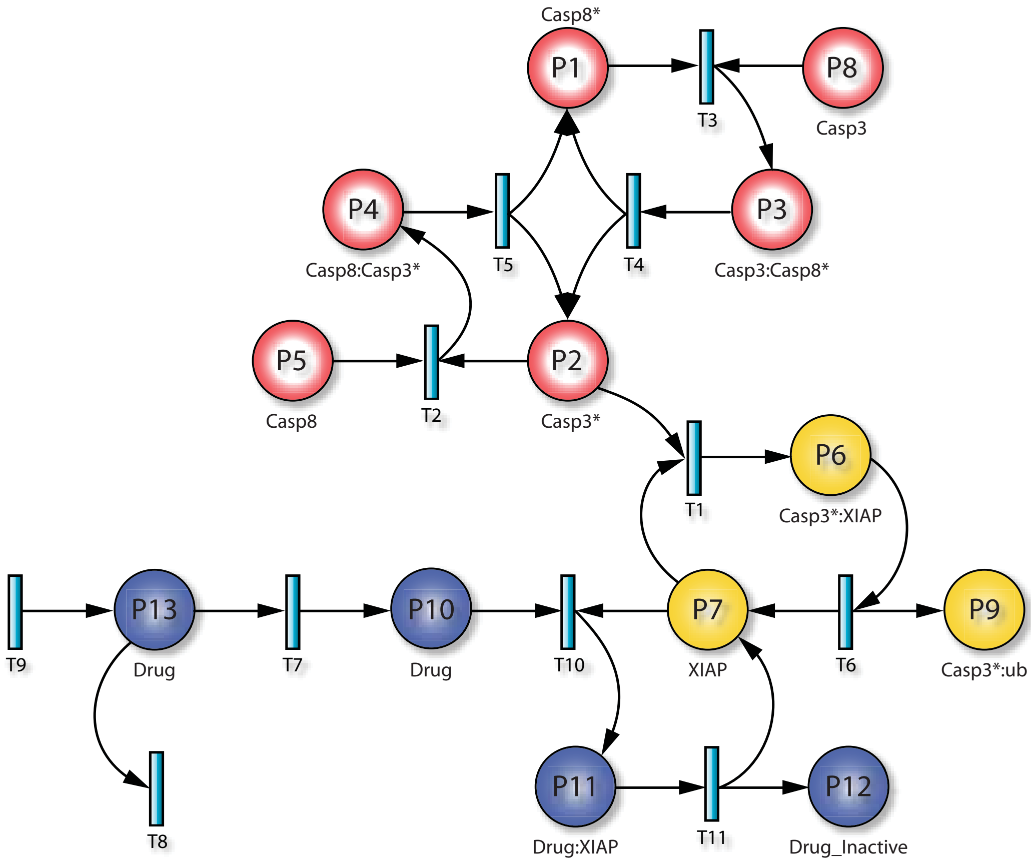


Figure 5.1: Part of Apoptosis pathway, and drug delivery mechanism.

desirable that the frequencies would be distributed more widely or with a larger variance, which would mean the input has had a better influence on the output. While a uniform distribution of frequencies shows that the output has changed more often, a narrow distribution would mean that changes in the input have not resulted into output changes, (i.e. the output has stayed unchanged). For simplicity use $V(l)$ as the l-th element of vector $V_{p_o}^{M_0}$, also $V(M_0, p_o) > V(M'_0, p_o)$ when M_0 results in at least as many elements in the frequency vector as M'_0, and the amplitude of each frequency is higher.

Definition 10. *$\rho(M_0, p_o)$ called the rigidity measure of a place p_o, is calculated as the l^2-norm of the vector $V_{p_o}^{M_0}$, which is the frequency of the token numbers in the place p_o when the initial condition is M_0.*

Using Definition 10 it is possible to drive a measure on how perturbance in the initial value of an input place would result in changes at the given output place. A lower rigidity measure means that the perturbed input has a better influence on the output and therefore is a better target (i.e. more sensitive) for control purposes.

Given an initial marking M_0, Petri net G can generate a reachability set $R_G(M_0)$. If M'_0 is obtained by adding a single token in a place p_j in M_0, then $R_G(M'_0)$ will either have the same size as $R_G(M_0)$ or have more elements. It can not shrink as the dynamics of the Petri net is not blocked by more tokens. Hence $V(M'_0, p_o) = V(M_0, p_o) + \delta$, where $\delta \geq 0$ is the change in frequency.

Case 1: ($\delta = 0$) If $R_G(M_0)$ would have the same size as $R_G(M'_0)$, the conclusion is that, the chosen input place is saturated and can no more use extra tokens, hence some tokens will never leave that place.

Case 2: ($\delta > 0$) If the size of $R_G(M_0)$ grows as a result of new tokens, then the conclusion is that, new states have been created.

Definition 11. *For a given initial condition M_0 and desired input place p_i, saturation limit $M_0^s(p_i)$ is the maximum number of tokens before p_i will be saturated.*

If the number of tokens in a place p_i are less than $M_0^s(p_i)$, then all tokens in p_i will eventually be consumed.

Definition 12. *$R_G(M_0)$, the reachability set of Petri net G starting from initial condition M_0 has a deeper reach than $R_G(M_0')$ and is written as $R_G(M_0) \geq R_G(M_0')$ when there exists a one to one map τ from $R_G(M_0')$ to a subset of $R_G(M_0)$ in which for every $M' \in R_G(M_0')$ there exists an $M \in R_G(M_0)$ with $M' \leq M$ element-wise.*

Definition 12 implies that if $R_G(M_0) \geq R_G(M_0')$ then for every reachable state M' in $R_G(M_0')$ there exists a similar reachable state M in $R_G(M_0)$ in which all enabled transitions in M' can fire.

In order to achieve sensitivity analysis using the rigidity measure the following algorithm has to be used.

Algorithm 3. *Given a Petri net G, output place p_o and a set of inputs $\mathbb{P} = \{P_1, ..., P_k\}$, find a place $p_i \in \mathbb{P}$, such that p_o is least rigid (i.e. most sensitive) to p_i.*

Step 1. *Choose an initial condition M_0, also choose a sufficiently large $M = M_0 + \Delta M$ as an upper bound for the initial condition. Choose $j = 1$, and set $M_t = M_0$.*

Step 2. *Calculate the reachability graph $R_G(M_t)$.*

Step 3. *Calculate $V_{p_o}^{M_t}$, the frequency of output, and take an l^2-norm, to find the rigidity measure $\rho(M_t, p_o)$, save this value in a set $\mathbb{S}$.*

Step 4. *If $M_t < M$ add one token to the j-th place in M_t and go to step 2.*

Step 5. *If $j < k$, then set $j = j + 1$, $M_t = M_0$ and go to step 4, Otherwise go to step 6.*

Step 6. *Using $\mathbb{S}$, the set of rigidity measures for each input in $\mathbb{P}$, find the index i representing smallest values. $p_i \in \mathbb{P}$ is the desired place.*

Remark 6. *The upper bound M in step 1 of algorithm 3 can be chosen arbitrarily large, as long as each element in M is larger than the largest element of M_0, so all elements of ΔM are positive and non zero. Also no place should be saturated, meaning that the newly added token should be consumed. Ideally this range should result in enough points for comparison of all rigidity measures. Note that usually sensitivity is studied in a small neighbourhood of operating point.*

As can be observed from algorithm 3 in order to calculate this measure, first the reachability set for Petri net has to be calculated. A breadth-first search (BFS) method can be used that finds a list of all possible transition firings and fires them all, and adds all the new states to the reachability set, and the process is repeated until no new states can be constructed. Note that there are other algorithms that could be more efficient than BFS, however the performance should be evaluated based on each specific case.

In general the calculation of the reachability set is very time consuming and exponentially complex. Furthermore this algorithm requires that this calculation be repeated many times, to find the best target. Therefore it is desired to find a better solution for the sensitivity analysis.

Lemma 1. *If $M_0' \leq M_0$ then $\rho(M_0', p_o) \leq \rho(M_0, p_o)$. The Petri net rigidity measure of a given output place for an input as defined in Definition 10 is strictly increasing with respect* *to the initial condition.*

Proof: By assumption M_0' is coverable by M_0, then $R_G(M_0)$ has a deeper reach than $R_G(M_0')$, or $R_G(M_0) \geq R_G(M_0')$. Therefore $V(M_0, p_o) \geq V(M_0', p_o)$, in other words the number of times that the Petri net visits a certain state would only increase as the initial condition increases. Since the rigidity measure is an l^2-norm of the frequency of the output at a given place (i.e. output) therefore the rigidity measure is a monotonically increasing function with respect to the initial condition. □

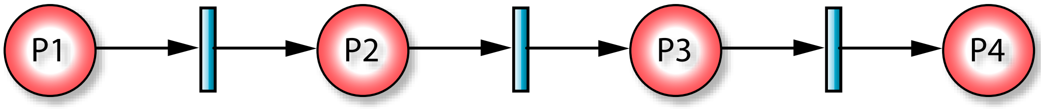


Figure 5.2: Petri net with 4 places, P_3 is assumed to be the output.

Once it is shown that the rigidity measure will only increase with respect to initial condition (as results are shown in figure 5.4 for a simple example depicted in figure 5.2), it is desired to know if the rate at which it increases has a specific behaviour. The following assumptions will be used to clarify the process of proof, however they can be easily relaxed as will be shown later.

The Petri net model is trying to capture all possible combinations of proteins and produce a reachability set that is covering as many states as possible. However when the norm of the initial condition (e.g. infinity norm or l^2-norm) is smaller than some $\mathbb{D}$ it is possible that the network would get blocked by lack of tokens, and hence the reachability graph would not be able to show all possible dynamics of the system. Therefore it is useful

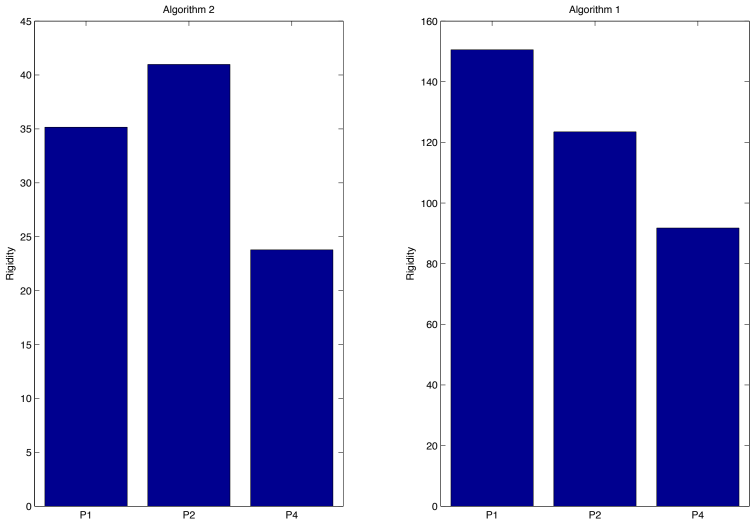


Figure 5.3: Rigidity measure for the system depicted in Figure 5.2.

to assume that there are enough tokens in the net initially that would result in a system which can exhibit all possible behaviour.

Assumption 3. *For any $\mathbb{E} > 0$ there exists a $\mathbb{D} > 0$ for which, if the l^2-norm of the initial condition is larger than $\mathbb{D}$ then all transitions in the system can fire more than $\mathbb{E}$ times.*

Assumption 3 says that our system shows a rather smooth behaviour, and that all transitions get to fire enough, to demonstrate the full dynamic range of the system.

Remark 7. *In modelling biological systems, it is known that the system will not exhibit unbounded behaviour. Furthermore, the system is globally fair. It can also be assumed that the weight of all arcs are equal to one. Firstly the chemical reactions of Caspase cascade in Apoptosis will result in such weights, and secondly even if a system results in larger weights for arcs, without loss of generality the framework presented here can be expanded to cover those cases.*

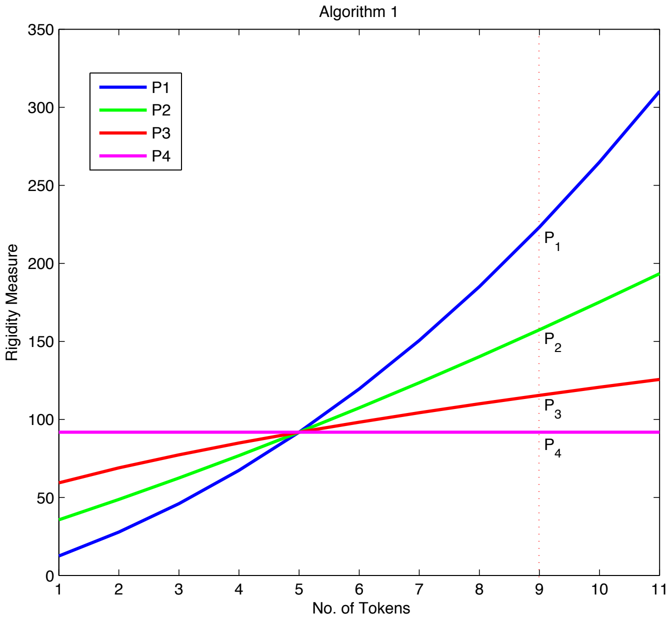


Figure 5.4: Rigidity measure calculated for a linear Petri net with four places shown in Figure 5.2, P_3 is assumed to be the output.

In order to be able to compare rigidity of an output p_o with respect to two different input places p_1 and p_2, it is also necessary that the two measures would have same origin so that they are comparable. In other words the initial condition at p_1 and p_2 has to be the same.

Assumption 4. *For comparing rigidity measures of p_1 and p_2 the reachability set is calculated starting from the same initial condition M_0.*

It follows immediately from Assumption 4 that the reachability set for two initial conditions in p_1 and p_2 is equal initially and hence the rigidity measure is equal at the first point.

We further assume that the two places to compare have different structural significance on the output and hence different sensitivities.

Assumption 5. *The subnets starting from either of the two places p_1 and p_2 and ending in the output place p_o are structurally such that, equal perturbation in p_1 and p_2 results in different rigidity measures at p_o.*

Assumption 5 requires that neither the subnets from p_1 and p_2 to p_o would be isomorphic by a mapping of transition and place names, nor any abstracted versions of them be isomorphic. As a result they will have different sensitivities. It is now possible to prove the second lemma, that the two rigidity measure signals which are always increasing and smooth are not going to cross in a point other than their initial condition.

Assumption 6. *The system being studied does not have any sources of new tokens and no sinks. Furthermore the biological systems are naturally bounded.*

Assumption 7. *When comparing sensitivity of p_1 and p_2, if all initial tokens in p_1 or p_2 are consumed, and the number of tokens reaches zero, it will not go to one again.*

Lemma 2. *For every sequence s, where $M_{k+1} \xrightarrow{s} M'_{k+1}$ and $M_{k+1}(p_i) = M_k(p_i) + 1 > 0$ and $M'_{k+1}(p_i) = 0$, there exists an equivalent state in $R_G(M_{k+1})$ which does not exist in $R_G(M_k)$*

Proof: If a sequence such as $s = \{s_1, s_2, \cdots, s_j\}$ would take $M_{k+1}(p_i)$ to zero, it means that at least one s_i consumes the last token in p_i, which is impossible when the initial marking at p_i has one less token. □

Lemma 3. *The frequency changes of the output with respect to the changes in the input can be calculated and is only dependant on the structure of the Petri net.*

Proof: In Petri net G, using the state equation $M' = M + Ax$ will result in $A[i]x = M'[i] - M[i]$, where the i index represents the i-th row of the matrix. Suppose that $M'[i] = 0$, which means, the solutions $\{x_j\}$ of the state equation which are different firing sequences, will result in all tokens in the input place to be consumed, hence according to Lemma 2 creating new states. For each solution x_j, a new state is added to $R_G(M)$. Now suppose that initial condition M_k has been used and M_{k+1} has been obtained by adding one token to place p_i, and M'_{k+1} is a marking of the system when all tokens of p_i have been consumed (i.e. $M'_{k+1}[p_i] = 0$).

$$\begin{tikzcd}
M_k \arrow[r] \arrow[d, "\textcircled{+1}"'] & R_G(M_k) \\
M_{k+1} \arrow[r] & R_G(M_{k+1})
\end{tikzcd}$$

Figure 5.5: Perturbation

$$\begin{aligned} M'_{k+1} &= M_{k+1} + Ax \\ A[p_i]x &= -m_{k+1} \end{aligned} \tag{5.1}$$

where $A(p_i)$ is a vector equal to the row in matrix A that corresponds to place p_i. The diophantine equation 5.1 can have many solutions. Let $X^{p_i}_{k+1} = \{\bar{x}\}$ be the set of all such solutions. Now for $X^{p_i}_k$ and $X^{p_i}_{k+1}$, and $x \in X^{p_i}_k$ and $x' \in X^{p_i}_{k+1}$ one can write,

$$\begin{aligned} A[p_i]x &= -m_k \\ A[p_i]x' &= -m_{k+1} \\ &= -(m_k + 1) \end{aligned} \tag{5.2}$$

x' can be written as $x_1 + x_2$, where $x_1 = x$ and $A(p_i)(x_1 + x_2) = -m_k - 1$, hence

$$A[p_i]x_2 = -1 \tag{5.3}$$

Now suppose that the row corresponding to the output or $A(p_o)$ is used to calculate the marking after the same sequence of firings (i.e. x' the solution obtained previously),

$$\begin{aligned} M'_{k+1} &= M_{k+1} + A(p_o)x' \\ &= M_{k+1} + A[p_o](x_1 + x_2) \\ &= M_{k+1} + A[p_o]x_1 + A[p_o]x_2 \end{aligned} \tag{5.4}$$

Calculating the frequencies, knowing that with slight abuse of notation $V(M_{k+1}, p_0) = V(M_k, p_0) + \delta_k$,

$$\begin{aligned} \delta_{k+1} &= \delta_k + \delta' \\ &= \delta_0 + k\delta' \end{aligned} \tag{5.5}$$

It can then be written that,

$$\begin{aligned} V_{k+1} &= V_k + \delta_k \\ &= V_0 + \sum_{i=1}^{k} \delta_i \\ &= V_0 + \sum_{i=1}^{k} (k\delta' + \delta_0) \\ V_{k+1} &= V_0 + k\delta_0 + \frac{k(k+1)\delta'}{2} \end{aligned} \tag{5.6}$$

□

Lemma 4. *For a given common output p_o the rigidity measure for increasing initial conditions smaller than the saturation limit at p_1 and p_2 will only be equal when the initial conditions are equal.*

Proof: Following Assumption 4 it is known that the two rigidity measures for p_1 and p_2 are equal at the initial condition M_0. Suppose that one token is added once to p_1 and once to p_2, to arrive at M_1^1 and M_1^2 for calculating the rigidity measure at p_1 and p_2 respectively. Assumption 5 will result that the rigidity measures at M_1^1 and M_1^2 are not equal. Without loss of generality, assume that the rigidity calculated for p_1 is smaller than p_2. In order to use strong induction, assume that for a given k, M_k^1 and M_k^2 are the initial conditions for p_1 and p_2, obtained by adding k tokens to p_1 and p_2 respectively in M_0, and that the rigidity calculated at p_1 is smaller than p_2. It is now desired to show that the relationship will hold at $k+1$ as well, and therefore the two signals will never cross again after M_0.

Using Lemma 3 the frequency of the output can be calculated, and will result in,

$$V_{P_o}^{M_{k+1}} = V_{P_o}^{M_0} + k\delta_0 + \frac{k(k+1)\delta'}{2} \tag{5.7}$$

Therefore,

$$\exists c_i > 0 \text{ such that, } l^2(V_{P_o}^{M_{k+1}}) \leq c_i.l^2(V_{P_o}^{M_k}). \tag{5.8}$$

The positive constant c_i is called the rigidity norm. Using equation 5.7 and 5.8, the rigidity norm can be calculated, and for each input-output pair would be among a set of constants.

Since the behaviour of the system is smooth and it is already known that the rigidity at p_1 is smaller than in p_2, adding one token to p_1 and p_2 will preserve the order of the rigidity measures. □

Remark 8. *Note that using Lemma 3 one can calculate the rigidity measure without the need of extensive reachability analysis.*

Now that it is shown the two rigidity measures are only increasing and after initial condition will never cross again, it can be concluded that for the purpose of identifying the most sensitive place among a given set of places, it is not necessary to calculate the rigidity measure at all points, it suffices to calculate at one single point that satisfies the assumptions. Therefore it is not necessary to calculate the reachability graph at different initial conditions.

Theorem 2. *Given a set $\mathbb{P} = \{p_1, p_2\}$ of places, in order to find the most sensitive place with respect to a place p_o it is sufficient to calculate the rigidity measure at M_k^1 and M_k^2 such that $M^1(p_1) = k$, $M^2(p_2) = k$ and $M^1(p_2) = M^2(p_1) = 1$, where k satisfies assumption 3.*

Proof: The proof follows from Lemma 1 and Lemma 4. □

It is obvious from Theorem 2 that for any given set of places with more than two members, the same deduction can be obtained.

Remark 9. *If Assumption 3 does not hold, it means that there are parts of the system that are blocked by lack of tokens, as long as the output place is not among the blocked places, one can assume that the Petri net does not have those places. Also assumption 4 does not really pose a limitation on the system, it only provides a framework to calculate and compare the measure. Regarding assumption 5, if there exists an exact isomorphism* *between all possible paths from two input places to the desired output place, then their rigidity will be equal.*

Using the proposed measure is useful for sensitivity analysis, however there are two factors that need to be considered. First, it would be interesting to use various other functions or norms to achieve different measures other than sensitivity as presented here. Second, It is desirable to further reduce the need to calculate reachability graph at a single point k. In the main example studied here (Example 6), for which the simulation results are presented in section 5.3, the reachability graph will exceed 74000 states, even using relatively small initial conditions. Therefore a more efficient and flexible approach would be very useful. In the rest of this section, a method is proposed to calculate rigidity with less computational expense.

Definition 13. *The frequency difference from l_k to l_{k-1} for $k = 1, 2, \cdots$ is defined as,*

$$V_{P_o}^{M_0}(l_{k-1}, l_k) = V_{P_o}^{M_0}(l_k) - V_{P_o}^{M_0}(l_{k-1}) \tag{5.9}$$

Assumption 8. *The random variables $V_{P_o}^{M_0}(l_1)$, $V_{P_o}^{M_0}(l_1, l_2)$, $V_{P_o}^{M_0}(l_2)$ are mutually independent for any $l_1 < l_2$*

Assumption 9. *$P[V_{P_o}^{M_0}(l_{k-1}, l_k) = n]$, for $n = 0, 1, \cdots$ may depend on the interval length (l_{k-1}, l_k), but it is independent of l_{k-1} and l_k for any, $k = 1, 2, \cdots$.*

Definition 14. *$P_i(l)$ is the probability that the output place of the Petri net $G(M_0)$ would have l tokens, exactly i times in its reachability graph.*

Definition 14 can be written as,

$$P_i(l) \equiv P[V_{P_o}^{M_0}(l) = i]. \tag{5.10}$$

By definition take $P_0(0) = 1$, also since tokens arrive only on integer values, for any $h < 1$, it can be seen that $P_0(h) = 1$.

Lemma 5. *For any sufficiently small h and for Petri net G with assumption 3 it can be written $P_0(t+h) = P_0(t).P_0(h)$.*

Proof: By definition,

$$\begin{aligned} P_0(t+h) &\equiv P[V_{P_o}^{M_0}(t+h) = 0] \\ P[V_{P_o}^{M_0}(t+h) = 0] &= P[V_{P_o}^{M_0}(t) + V_{P_o}^{M_0}(t,t+h) = 0] \\ \xrightarrow{V(l)\geq 0} &= P[V_{P_o}^{M_0}(t) = 0 \cap V_{P_o}^{M_0}(t,t+h) = 0] \\ \xrightarrow{Assumption\ 9} &= P[V_{P_o}^{M_0}(t) = 0 \cap V_{P_o}^{M_0}(0,h) = 0] \\ \xrightarrow{Assumption\ 8} &= P[V_{P_o}^{M_0}(t) = 0].P[V_{P_o}^{M_0}(h) - V_{P_o}^{M_0}(0) = 0] \\ \xrightarrow{P_0(0)=1} &= P[V_{P_o}^{M_0}(t) = 0].P[V_{P_o}^{M_0}(h) = 0] \\ &= P_0(t).P_0(h). \end{aligned} \tag{5.11}$$

□

Lemma 6. *The probability of first token arrival is exponentially distributed.*

Proof: Assume that $P_1(h)$ is differentiable at $h = 0$, and the derivative is a positive value such as λ,

$$P_1'(0) = \lambda \tag{5.12}$$

Having $P_{\geq 2}(h) = \sum_{i=2}^{\infty} P_i(h)$, since P is a probability distribution and also since the tokens in the system are bounded, without loss of generality one can take $P'_{\geq 2}(0) = 0$ and hence,

$$\begin{aligned} P_0(h) + P_1(h) + P_{\geq 2}(h) &= 1 \\ P'_0(0) + P'_1(0) + P'_{\geq 2}(0) &= 0 \\ P'_0(0) &= -\lambda \end{aligned} \tag{5.13}$$

From lemma 5 it follows that,

$$\begin{aligned} P_0(t+h) &= P_0(t).P_0(h) \\ \frac{P_0(t+h) - P_0(t)}{h} &= P_0(t)\frac{P_0(h) - 1}{h} \\ &= P_0(t)\frac{P_0(h) - P_0(0)}{h} \end{aligned} \tag{5.14}$$

Taking a very small h equation 5.14 will lead, $P'_0(t) = -\lambda P_0(t)$, which can be solved and will result in $P_0(t) = e^{-\lambda t}$ □

Theorem 3. *The frequency vector as defined in definition 9 follows a Poisson distribution.*

Proof: The proof follows from Lemma 5 and Lemma 6. From lemma 5, and using polynomial expansion of the exponential, it follows that,

$$\begin{aligned}
&P[V_{P_o}^{M_0}(t+h)=n] \\
&\quad =\sum_{j=0}^{n} P[V_{P_o}^{M_0}(t,t+h)=n-j \cap V_{P_o}^{M_0}(t)=j] \\
&\quad =\sum_{j=0}^{n} P[V_{P_o}^{M_0}(t,t+h)=n-j].P[V_{P_o}^{M_0}(t)=j] \\
&\quad =\sum_{j=0}^{n} P[V_{P_o}^{M_0}(h)=n-j].P[V_{P_o}^{M_0}(t)=j] \\
&\quad =P[V_{P_o}^{M_0}(h)=0].P[V_{P_o}^{M_0}(t)=n] \\
&\qquad +P[V_{P_o}^{M_0}(h)=1].P[V_{P_o}^{M_0}(t)=n-1] \\
&\qquad +\sum_{j=0}^{n-2} P[V_{P_o}^{M_0}(h)=n-j].P[V_{P_o}^{M_0}(t)=j] \\
&\quad \simeq(1-\lambda h)P[V_{P_o}^{M_0}(t)=n]+\lambda hP[V_{P_o}^{M_0}(t)=n-1]
\end{aligned} \tag{5.15}$$

therefore

$$\begin{aligned}
P_n(t+h) &=(1-\lambda h)P_n(t)+\lambda hP_{n-1}(t) \\
P_n(t+h)-P_n(t) &=-\lambda hP_n(t)+\lambda hP_{n-1}(t) \\
\frac{P_n(t+h)-P_n(t)}{h} &=-\lambda P_n(t)+\lambda P_{n-1}(t)
\end{aligned} \tag{5.16}$$

hence

$$P_1(t)=(\lambda t)e^{-\lambda t} \tag{5.17}$$

continuing to use the same logic will result in the following distribution for $V(l)$ which is a Poisson distribution.

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \tag{5.18}$$

□

Now that $V_{P_o}^{M_0}$ is known to follow Poisson, the rigidity measure can be defined as higher moments of $V_{P_o}^{M_0}$ and hence obtained without having to calculate the reachability tree. The rigidity norm c_i therefore would be calculated from the probability distribution function.

Remark 10. *The biological nature of the system being studied guarantees that assumptions 8 and 9 are satisfied.*

Definition 15. $\sigma_{P_o}^{M_0}$ *called the refined rigidity measure of a protein p_o, is calculated as the second moment around the mean for the distribution of $V_{P_o}^{M_0}$, which is the frequency of the tokens in the place p_o when the initial condition is M_0.*

The moment will be calculated using

$$E(V) = (\lambda t)e^{-\lambda t} \sum_{m=0}^{\infty} \frac{(\lambda t)^m}{m!} = \lambda t \tag{5.19}$$

In order to achieve sensitivity analysis using the refined measure the following algorithm is used.

Algorithm 4. *Given a Petri net G, output place p_o and set of inputs $\mathbb{P} = \{P_1, ..., P_k\}$ find a place $P_i \in \mathbb{P}$, such that p_o is most sensitive to P_i.*

Step 1. *Choose an initial condition M_0, also choose $m > \max(M_0)$. Choose $j = 1$.*

Step 2. *For each $P_j \in \mathbb{P}$, substitute the corresponding element in M_0 with m, and go to step 3.*

Step 3. *Generate partial reachability for $G(R_0)$, and estimate mean $\hat{\lambda}_j$ by fitting a poisson distribution to the data.*

Step 4. *Using $\hat{\lambda}_j$, calculate the second moment as c_j and save it in $\mathbb{S}$.*

Step 5. *If every $P \in \mathbb{P}$ has been used go to next step, otherwise go to step 2.*

Step 6. *Using $\mathbb{S}$, the set of rigidity measures for each input in $\mathbb{P}$, find the index i representing smallest values. $P_i \in \mathbb{P}$ is the desired place.*

Remark 11. *Step 3 of algorithm 4 needs just enough data to be able to estimate the mean of a poisson distribution. Since Poisson does not depend on the where the sample is taken, the sampling can be taken very easily.*

Remark 12. *As can be observed algorithm 4 is much more efficient than algorithm 3. First it does not repeat the calculation for multiple points, and second, as it does not require full reachability analysis. As shown in the simulations it will yield in similar results.*

Remark 13. *In any application where assumption 8 or 9 would not hold and hence the Poisson distribution could not be used, still the appraoch used in algorithm 4 could be used by estimating the mean of the appropriate distribution.*

5.3 Simulation

The theoretical results of section 5.2 are first validated using a simple artificial example.

Example 5. *Suppose that a simple Petri net with three places is given as shown in Figure 5.6, and P_3 is the output, where sensitivity to other elements needs to be studied. Starting from a common initial condition $M_0 = \begin{bmatrix} 5 & 5 & 5 \end{bmatrix}^T$, the initial conditions of each place is changed from* 1 *to* 11*, and rigidity measure is calculated for each point. As can be seen in Figure 5.7, P_2 has a lower rigidity measure than P_1 and therefore is a better target to influence P_3 levels, which is also expected since it can be observed that P_1 is farther to the output P_3, furthermore the measure does not change at P_3 since all tokens are finally deposited at P_3 and when the initial condition is perturbed at P_3 no new behaviour will be observed. Figure 5.8 shows the results for both Algorithm 3 on the right and 4 on the left. It can be observed that the results here are not consistent. This is due to the fact that place P_1 is depleted from tokens without any source and hence does not satisfy the condition of the assumption 8. As the network gets more complex, it can be seen that the two algorithms will provide the same results, except at the input place P_1.*

Suppose that another place is added to the Petri net, such as P_4 in Figure 5.9, notice that P_4 is creating a parallel path and it's consumption is also regulating P_3. The simulation is now repeated for different initial conditions, and the resulting rigidity measures are shown in Figure 5.10.

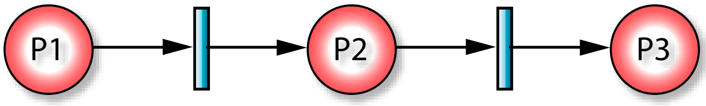


Figure 5.6: Petri net example with three places

As expected P_4 has the lowest rigidity measure and hence the best target to influence the output in P_3. Note that the rigidity measure for P_3 is still the lowest (i.e. highest influence

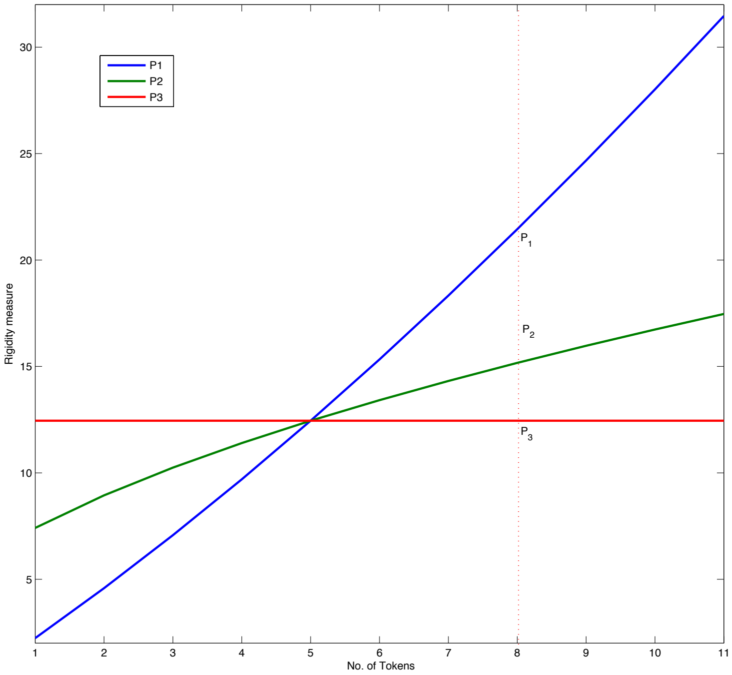


Figure 5.7: output rigidity measure for 3 places

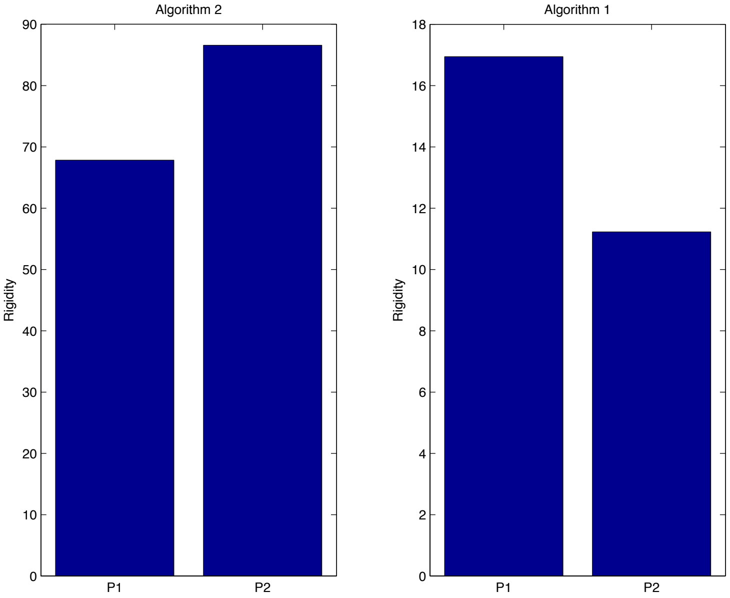


Figure 5.8: rigidity with three places

on itself) however since the tokens are being consumed by a transition, it does not stay constant.

The results of the two algorithms is shown in Figure 5.11, where Algorithm 3 is on the right and Algorithm 4 is on the left. The results are exactly as expected, where P4 has the lowest rigidity and P_1 which does not satisfy the condition does not stay consistent.

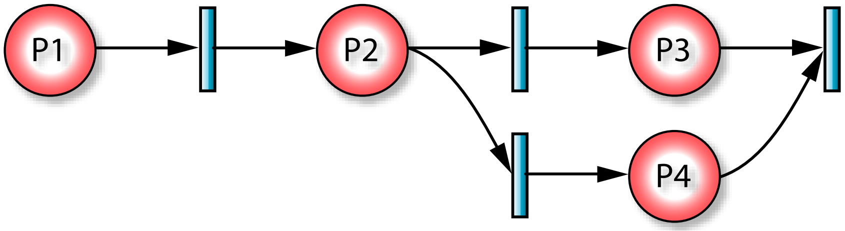


Figure 5.9: Petri net example with four places

Next a new place P_5 is added, and the transitions create a backward loop in the structure of the system, as depicted in Figure 5.12.

The initial conditions are changed as before and the results of rigidity measure is shown in Figure 5.13, As can be observed from Figure 5.13, P_4 has the lowest rigidity. The results of Algorithm 3 and 4 are shown in Figure 5.14.

Next the Theorem 2 is verified using the Caspase cascade pathway as introduced in Chapter 2, also the results match those of the continuous analysis done using SimBiology in MATLAB [28].

Example 6. *In the Caspase cascade as one of the most important parts of Apoptosis pathway it is desired to identify the best drug target that would influence the casp3* concentrations. The continuous analysis done in MATLAB verifies that XIAP is the best choice. The*

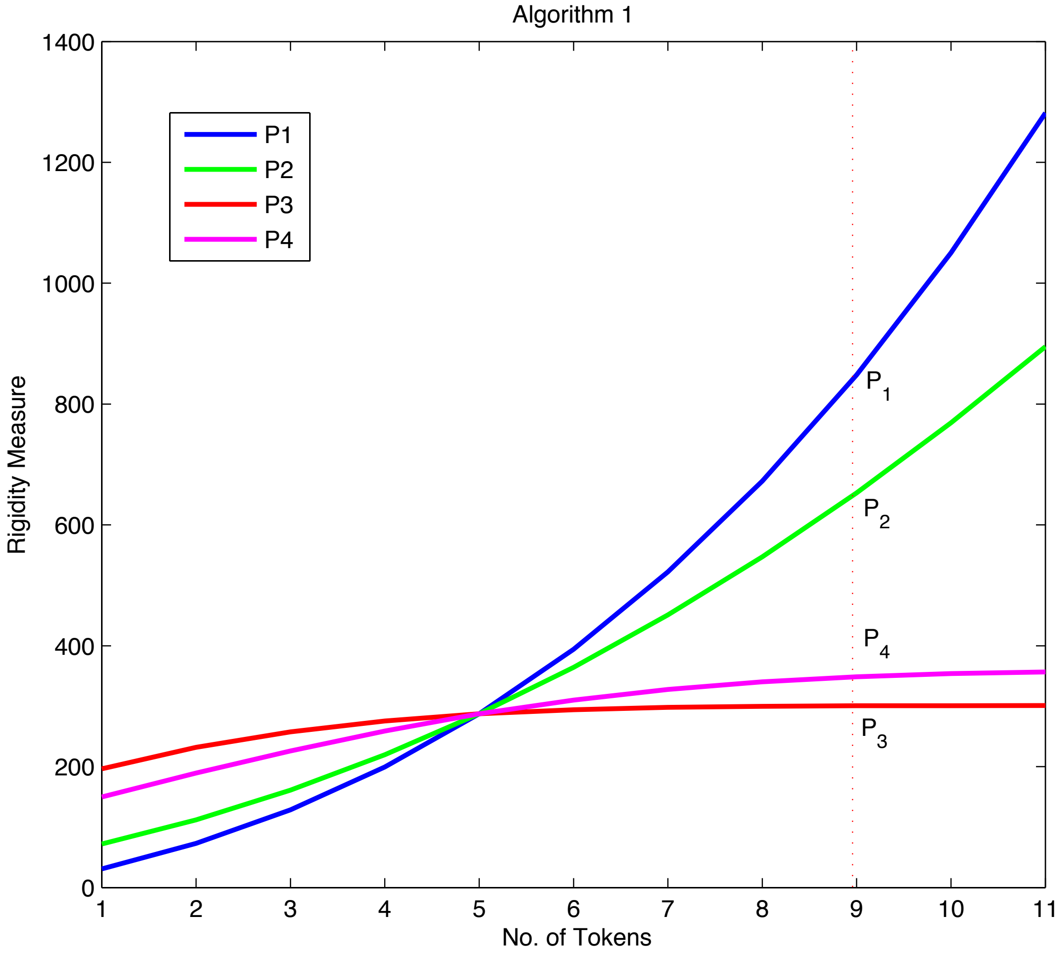


Figure 5.10: output rigidity measure for 4 places shown in figure 5.9

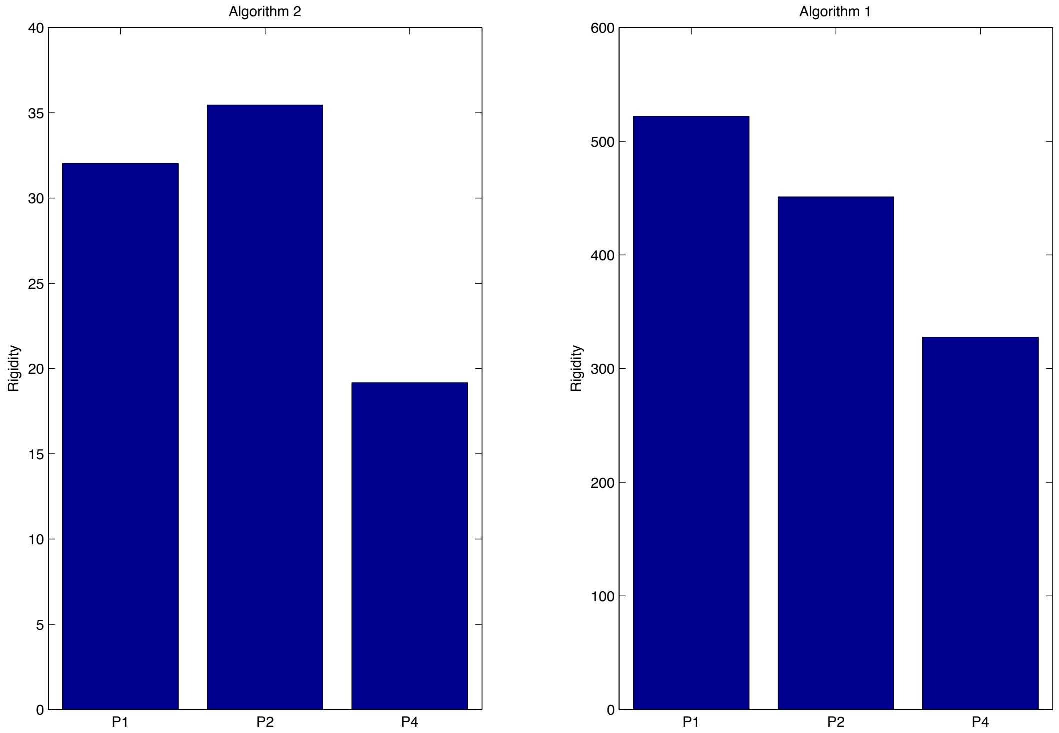


Figure 5.11: rigidity with four places

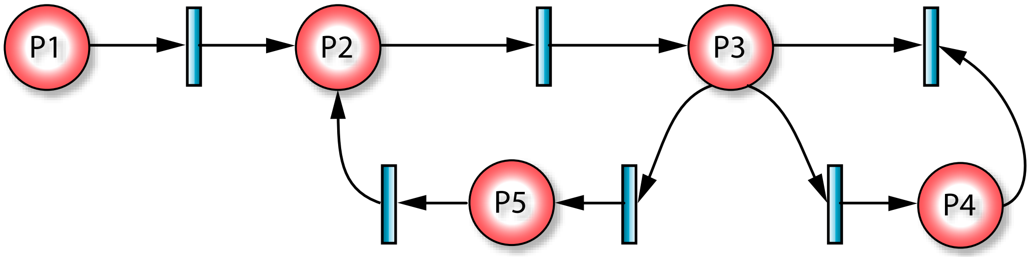


Figure 5.12: Petri net example with five places and loop.

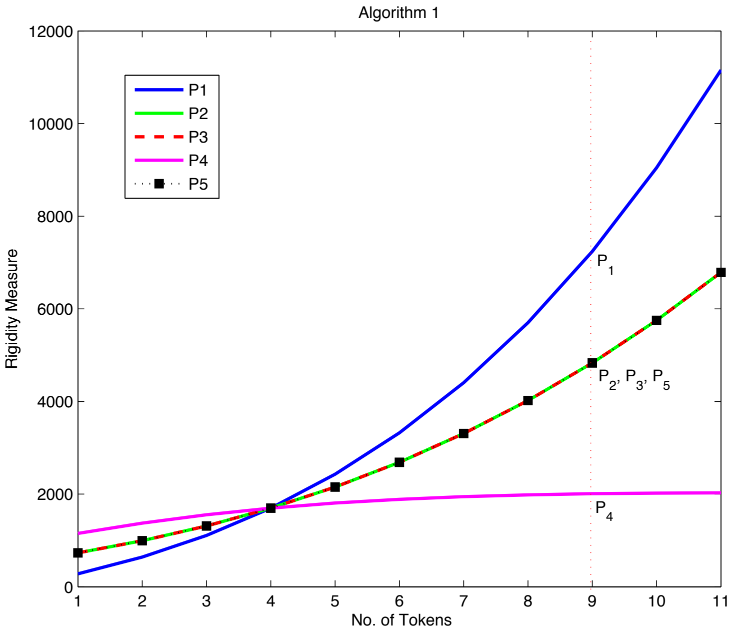


Figure 5.13: output rigidity measure for 5 places with loop.

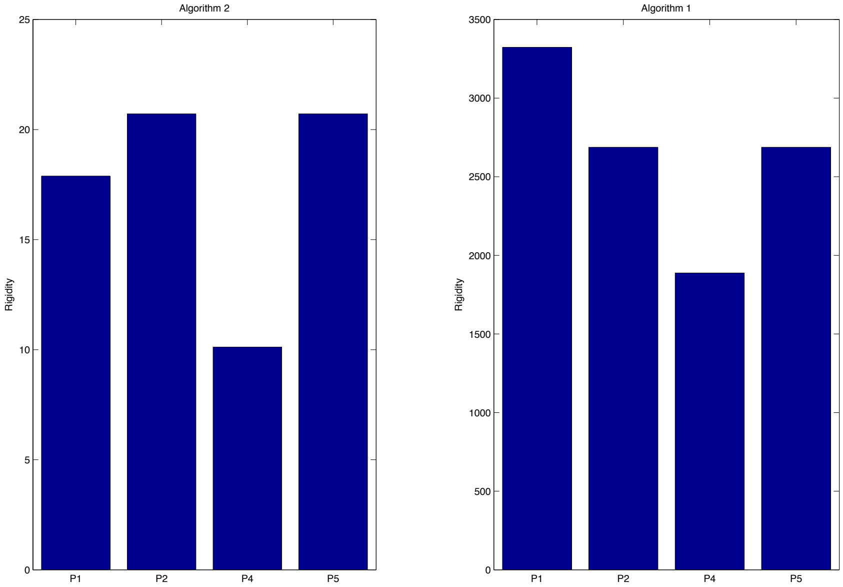


Figure 5.14: rigidity with five places.

Petri net of the model is constructed as depicted in Figure 5.1. The initial conditions are then perturbed for various molecules in the network and the rigidity measure is calculated as shown in Figure 5.15.

As can be observed the lowest rigidity measure belongs to XIAP, which matches the actual results in the biology literature [77] as well as continuous sensitivity analysis in SimBiology tool for MATLAB, as seen in Figure 5.16, where XIAP has the highest sensitivity and Casp 8 has the lowest, which matches the results obtained from Petri net analysis. Note

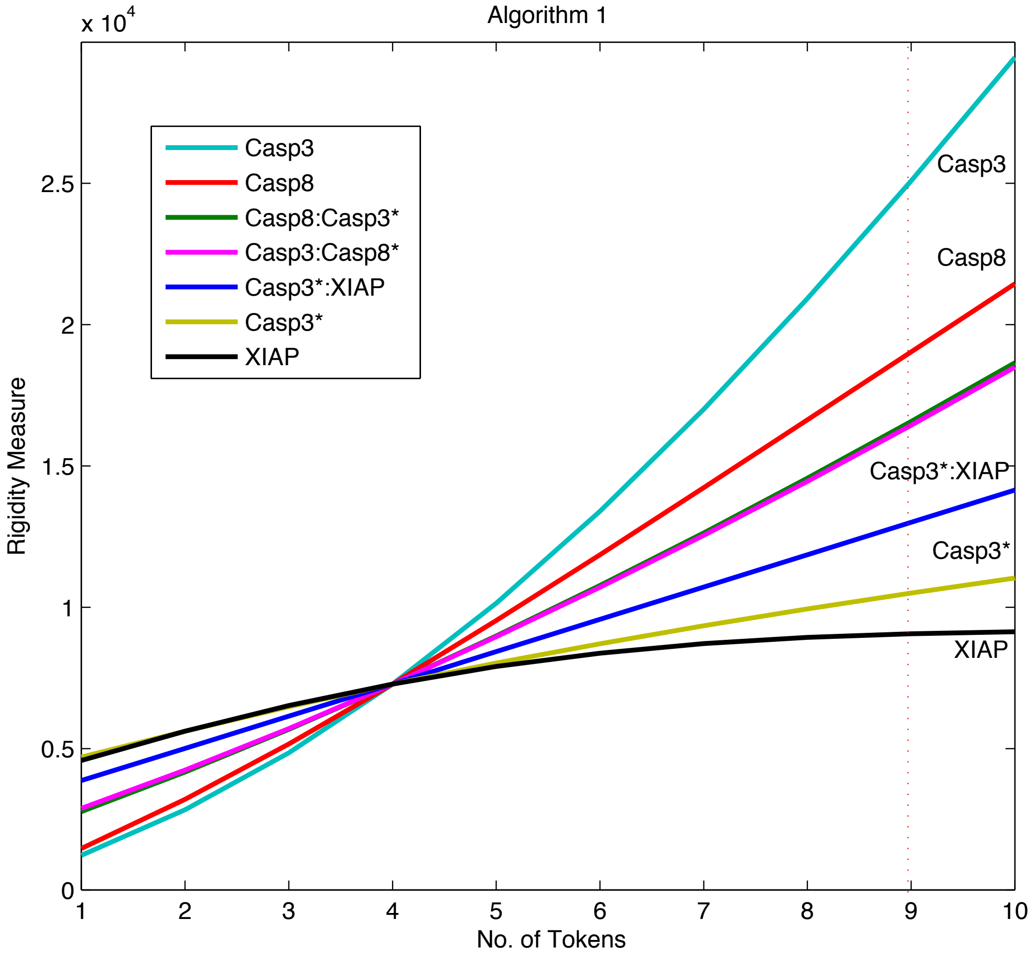


Figure 5.15: Caspase cascade rigidity measures.

that the definition of sensitivity used here is not exactly replicating that of SimBiology and hence does not result in exact same values.

Figure 5.17 shows the results of the Algorithm 3, where XIAP has the lowest rigidity.

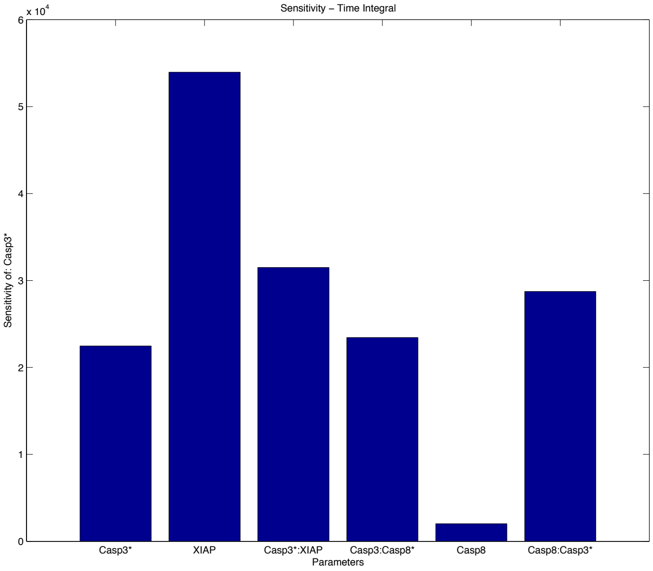


Figure 5.16: Sensitivity calculated using continuous model with SimBiology in MATLAB

Figure 5.19 depicts the results of Algorithm 4, XIAP still has the lowest value, and the order of all places is also the same. Note that as expected all rigidity measures are equal at the point in which the initial conditions have been equal. Figure 5.20 shows the output distribution for six different proteins in the network, the x axis represents the number of tokens in the output protein, while the y axis is the number of times at which that value has been observed. Multiple lines show increasing values on initial condition in respective molecule.

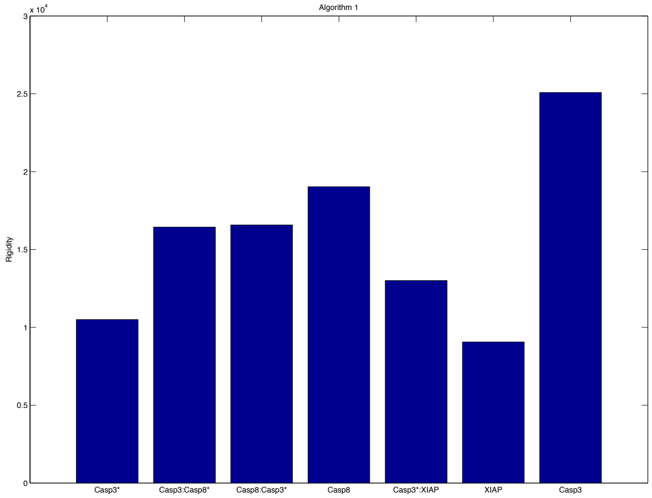


Figure 5.17: Rigidity for apoptosis calculated using Algorithm 3

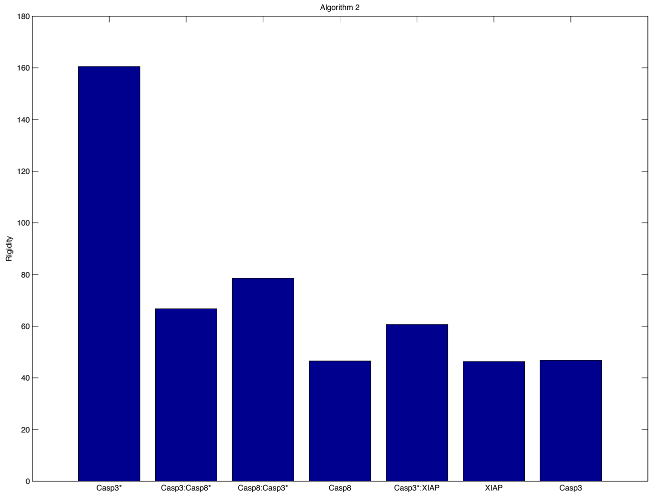


Figure 5.18: Refined rigidity for apoptosis calculated using Algorithm 4 with maximum 5000 states calculated in reachability

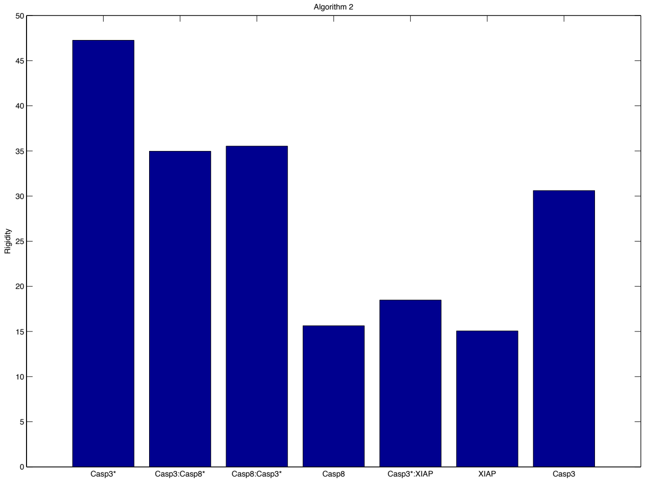


Figure 5.19: Refined rigidity for apoptosis calculated using poisson estimates on full reachability

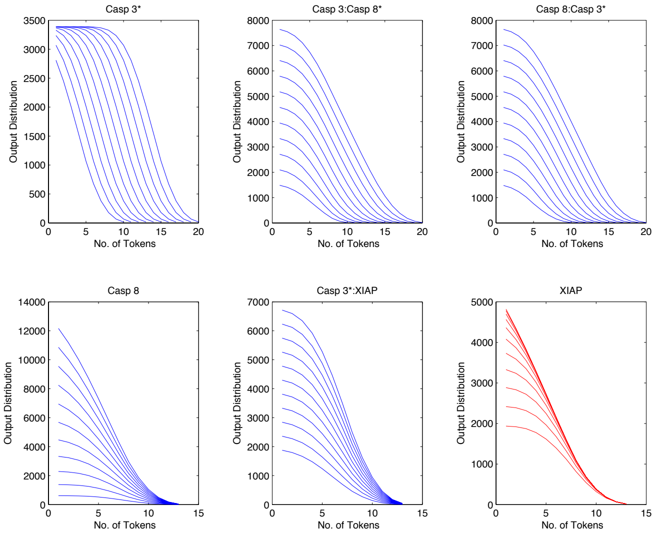


Figure 5.20: Distribution of different outputs

5.4 Conclusion

Knowledge-intensive systems such as biological systems are the new frontiers of research for engineers, who wish to bring the power of their tools to the modern problems that have been formed due to huge amounts of data generated each day. Discrete modelling using Petri nets has been proposed for biological systems, however control analysis for such systems requires new tools. In this work a framework has been proposed for efficiently calculating sensitivity of a desired output with respect to different protein concentrations. It is shown using a simple example as well as a real biological example that the measure introduced can accurately identify the desired targets. The mathematical proof is provided to show that sensitivity analysis can be done without structural analysis of the Petri net and by looking at the behaviour of the Petri net for various initial conditions. It would be interesting to study the rigidity norm as defined in this work, which could help relate the same sensitivity analysis directly to the structure of the Petri net. Also for different applications one might be interested in defining other norms than l^2-norm, such as a weighted norm, which would put more emphasis on certain values of the output. This could be regarded as a very important step in control design for the biological systems and hence a useful tool in drug design.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The use of discrete event modelling tools for analysis of biological systems has been studied in this thesis. The first goal is to establish a useful framework in capturing the knowledge of such systems using Petri nets. As a result of the size and inherent complexity of the system, a mechanism is introduced for consistency analysis between Petri nets obtained from two knowledge sources. This framework can be very well applied to other knowledge-intensive systems such as financial systems or social systems.

A case study is conducted to demonstrate the performance and success of the framework. The next goal is to develop tools and measures necessary for designing a control strategy which in the case of biological systems would translate to drug design. In order to achieve such analysis, a measure for the rigidity of the Petri net places is defined that can be used for identifying the proteins to which a desired output is most sensitive.

Sensitivity analysis of the petri net will help identify the best candidates for designing a drug to alter cell behaviour (i.e. control cell dynamics). It is shown that it is possible to define and evaluate a rigidity measure without the need for extensive reachability analysis on various initial markings. These models may be used for developing and implementing control strategies for altering cell dynamics.

The results of this analysis are then compared with the similar results obtained from the continuous model as performed in the SimBiology library of MATLAB. The proposed algorithm based on the discrete-event model results in the same identification of protein as in the continuous case.

With the rapid generation of data in various systems, the challenges are growing faster than ever to develop and utilize methods and frameworks that can efficiently and effectively analyze such systems, while this thesis does not find a cure for cancer, it can be regarded as an step towards such modelling frameworks.

6.2 Future research

This area of research is very new and has a lot of interesting subjects that can be studied. Some of the immediate topics that will follow this work include the study of quantization effect on consistency and sensitivity. The choice of quantization of protein concentrations will inevitably have an effect on the dynamics of the system. As a quantization function is used to assign discrete tokens to each place of the Petri net. It is interesting to study effects of different quantization methods.

Also when calculating the sensitivity measure, the focus of this work has been on l^2-norm, however depending on the application one could use weighted norms which would assign more weights to certain places.

The problem of sensitivity can also be studied in the framework of consistency. This would mean that the sensitivity at a high level model would be equivalent to the sensitivity for the low level model under certain conditions.

Similar to the methodology used for modelling consistency, a measure can be developed that would describe the control consistency of the two models. A refinement algorithm can then iterate on the model to create other possibilities for the control to be implemented. Eventually the algorithm will create a controllable language, based on the biological constraints on disabling events.

It is also desirable to see how the Assumption 1 and Assumption 2 can be relaxed, and what would be their impact on the general notion of consistency between the high-level and low-level models.

In the supervisory control of discrete event systems, it is normal to assume, that the alphabet of the system is divided into two disjoint sets, of controllable and uncontrollable events, and the supervisor would have the capacity to disable any controllable events at any time, without changing the nature of the events. In biological systems, the definitions of controllable and uncontrollable events will be different. While it is still true that some of the events can be disabled by increasing concentrations of other proteins, yet some events can be facilitated by such increase. Enabling or disabling of one event can potentially change the controllability of some other events. Also excessive increase in certain concentrations can have variable effect on disabling of events.

Mutations of different genes, can result in different sets of responses in the system. Two mutations are synthetically lethal if cells with either of the single mutations are viable but cells with both mutations are inviable. In other words, it is not just a single event that needs to be disabled but a string needs to be disabled in order for a certain behaviour to appear.

Furthermore, it is possible to force some of the events to happen while, a supervisor will only have the capacity of disabling events not triggering them. Hence, the control design will be different from the conventional Ramadge-Wonham supervisory control design. These issues could be subject of future studies.

Finally the study of supervisory control of Petri nets should be applied to the models of biological systems, using the framework developed in this thesis.

A control strategy that is designed based on the high-level (pathway level) model will be implemented on the low-level (protein interaction) model. The designed supervisor (controller) could essentially be an autonomous molecular computer [127] that would be used for anti-viral therapies, in which systematic gene delivery is applied to target and kill metastases is applied using genetically modified viruses or stem cells [23]. As a topic for future work, it would be interesting to consider uncertainty in the model and also investigate some combination of linear and discrete behaviour as a hybrid model.

Bibliography

[1] S. Nasr, “Islamic science: an illustrated study.,” *Islamic science: an illustrated study., by Nasr, SH. London (UK): World of Islam Festival Publishing, 14+ 273 p.*, vol. 1, 1976.

[2] L. Goyal, “Cell death inhibition: keeping caspases in check.,” *Cell*, vol. 104, no. 6, p. 805, 2001.

[3] G. D. Bader and C. W. V. Hogue, “Analyzing yeast protein-protein interaction data obtained from different sources,” *Nature Biotechnology*, vol. 20, no. 10, pp. 991 – 997, Oct. 2002.

[4] N. Przulj, D. Wigle, and I. Jurisica, “Functional topology in a network of protein interactions,” *Bioinformatics*, vol. 20, no. 3, pp. 340 – 348, 2004.

[5] E. Sontag, “Molecular systems biology and control,” *European Journal of Control*, vol. 11, no. 4, pp. 1 – 40, Jun. 2005.

[6] H. De Jong, J. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann, “Qualitative simulation of genetic regulatory networks using piecewise-linear models,” *Bulletin of Mathematical Biology*, vol. 66, no. 2, pp. 301–340, 2004.

[7] D. Gilbert and M. Heiner, "From petri nets to differential equations–an integrative approach for biochemical network analysis," *Petri Nets and Other Models of Concurrency-ICATPN 2006*, pp. 181–200, 2006.

[8] R. Schlatter, H. Conzelmann, E. Gilles, O. Sawodny, and T. Sauter, "Analysis of an apoptotic core model focused on experimental design using artificial data," *Systems Biology, IET*, vol. 3, no. 4, pp. 255–265, 2009.

[9] M. Johnston and D. Siegel, "Linear conjugacy of chemical reaction networks," *Journal of mathematical chemistry*, vol. 49, no. 7, pp. 1263–1282, 2011.

[10] N. El-Farra, A. Gani, and D. Christofides, "A switched systems approach for the analysis and control of mode transitions in biological networks," *in Proc. of American Contr. Conf.*, vol. 5, pp. 3247 – 3252, Jun. 2005.

[11] A. A. Julius, A. Halasz, V. Kumar, and G. J. Pappas, "Finite state abstraction of a stochastic model of the lactose regulation system of escherichia coli," *in Proc. of 45th IEEE Conf. on Decision and Control*, pp. 19 – 24, Dec. 2006.

[12] F. Hinkelmann, M. Brandon, B. Guang, R. McNeill, G. Blekherman, A. Veliz-Cuba, and R. Laubenbacher, "Adam: analysis of discrete models of biological systems using computer algebra," *BMC bioinformatics*, vol. 12, no. 1, p. 295, 2011.

[13] S. Ware and R. Malik, "Conflict-preserving abstraction of discrete event systems using annotated automata," *Discrete Event Dynamic Systems*, pp. 1–27, 2012.

[14] D. Degenring, M. Rohl, and A. Uhrmacher, "Discrete event, multi-level simulation of metabolite channeling," *BioSystems*, pp. 29–41, 2004.

[15] M. Chen and R. Hofestädt, “A medical bioinformatics approach for metabolic disorders: biomedical data prediction, modeling, and systematic analysis,” *Journal of Biomedical Informatics*, vol. 39, no. 2, pp. 147–159, 2006.

[16] H. Genrich, R. Küffner, and K. Voss, “Executable petri net models for the analysis of metabolic pathways,” *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 3, no. 4, pp. 394 – 404, Sep. 2001.

[17] D.-Y. Lee, R. Zimmer, S. Y. Lee, and S. Park, “Colored petri net modeling and simulation of signal transduction pathways,” *Discrete Event Dynamic Systems*, vol. 8, no. 2, pp. 112 – 122, 2006.

[18] R. Ghosh and C. Tomlin, “Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-notch protein signalling,” *IEE Proceedings on Systems Biology*, vol. 1, no. 1, pp. 170 – 183, Jun. 2004.

[19] A. Halasz, V. Kumar, M. Imielinski, C. Belta, O. Sokolsky, S. Pathak, and H. Rubin, “Analysis of lactose metabolism in e.coli using reachability analysis of hybrid systems,” *IET Systems Biology*, vol. 1, no. 2, pp. 130 – 148, 2007.

[20] H. Achrekar, A. Gandhe, R. Lazarus, S. Yu, and B. Liu, “Predicting flu trends using twitter data,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pp. 702–707, IEEE, 2011.

[21] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D’Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews, *et al.*, “Reactome: a knowledgebase of

biological pathways," *Nucleic acids research*, vol. 33, no. suppl 1, pp. D428–D432, 2005.

[22] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH, 2005.

[23] M. Stanizzi and S. Hall, "Clinical experience with gene therapy for the treatment of prostate cancer," *Reviews in Urology*, vol. 9, pp. S20 – S28, 2007.

[24] T. Ran, S. Kaplan, and E. Shapiro, "Molecular implementation of simple logic programs," *Nature nanotechnology*, vol. 4, no. 10, pp. 642–648, 2009.

[25] K. Tarassov, V. Messier, C. Landry, S. Radinovic, M. Molina, I. Shames, Y. Malitskaya, J. Vogel, H. Bussey, and S. Michnick, "An in vivo map of the yeast protein interactome," *Science Signalling*, vol. 320, no. 5882, p. 1465, 2008.

[26] J. Hesselberth, X. Chen, Z. Zhang, P. Sabo, R. Sandstrom, A. Reynolds, R. Thurman, S. Neph, M. Kuehn, W. Noble, *et al.*, "Global mapping of protein-dna interactions in vivo by digital genomic footprinting," *Nature methods*, vol. 6, no. 4, pp. 283–289, 2009.

[27] T. MASUJIMA, "Live single-cell mass spectrometry," *Analytical Sciences*, vol. 25, no. 8, pp. 953–960, 2009.

[28] A. Samant, "Getting started with SimBiology: An online tutorial," *The MathWorks*, p. http://www.mathworks.com/company/events/webinars, 2008.

[29] S. Janowski, B. Kormeier, T. Töpel, K. Hippe, R. Hofestädt, N. Willassen, R. Friesen, S. Rubert, D. Borck, P. Haugen, *et al.*, "Modeling of cell-to-cell communication processes with petri nets using the example of quorum sensing," *In Silico Biology*, vol. 10, no. 1, pp. 27–48, 2010.

[30] R. Riefstahl, "The date and provenance of the automata miniatures," *Art Bulletin*, pp. 206–215, 1929.

[31] P. Ramadge and W. Wonham, "Supervisory control of discrete event processes," *Feedback Control of Linear and Nonlinear Systems*, pp. 202–214, 1982.

[32] P. Ramadge and W. Wonham, "Supervision of discrete event processes," in *Decision and Control, 1982 21st IEEE Conference on*, vol. 21, pp. 1228–1229, IEEE, 1982.

[33] S. Eilenberg, *Automata, languages, and machines*. Academic press, 1974.

[34] C. Petri, "Kommunikation mitautomaten," *Bonn: Institut für Instrumentelle Mathematik, Schriften des NM*, no. 3, 1962.

[35] T. Murata, "State equation, controllability, and maximal matchings of petri nets," *Automatic Control, IEEE Transactions on*, vol. 22, no. 3, pp. 412–416, 1977.

[36] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541 – 580, 1989.

[37] L. Holloway, B. Krogh, and A. Giua, "A survey of petri net methods for controlled discrete event systems," *Discrete Event Dynamic Systems*, vol. 7, no. 2, pp. 151–190, 1997.

[38] J. Moody and P. Antsaklis, “Supervisory control of discrete event systems using petri nets,” *Springer*, 1998.

[39] A. Ghaffari, N. Rezg, and X. Xie, “Design of a live and maximally permissive petri net controller using the theory of regions,” *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 137–141, 2003.

[40] I. Suzuki and T. Murata, “A method for stepwise refinement and abstraction of petri nets,” *Journal of Computer and System Sciences*, vol. 27, no. 1, pp. 51–76, 1983.

[41] H. Ammar and S. Islam, “Time scale decomposition of a class of generalized stochastic petri net models,” *Software Engineering, IEEE Transactions on*, vol. 15, no. 6, pp. 809–820, 1989.

[42] M. Di Benedetto, S. Di Gennaro, and A. D’Innocenzo, “Verification of hybrid automata diagnosability by abstraction,” *Automatic Control, IEEE Transactions on*, vol. 56, no. 9, pp. 2050–2061, 2011.

[43] R. Hofestädt, “A petri net application of metabolic processes,” *Journal Systems Analysis Modelling Simulation*, vol. 16, pp. 113 – 122, 1994.

[44] M. Peleg, I. Yeh, and R. B. Altman, “Modelling biological processes using workflow and Petri net models,” *Bioinformatics*, vol. 18, no. 6, pp. 825 –837, 2002.

[45] C. Chaouiya, “Petri net modelling of biological networks,” *Briefings in Bioinformatics*, vol. 8, no. 4, pp. 210–219, 2007.

[46] R. Ewald, C. Maus, A. Rolfs, and A. Uhrmacher, “Discrete event modelling and simulation in systems biology,” *Journal of Simulation*, vol. 1, no. 2, pp. 81 – 96, 2007.

[47] C. Chaouiya, E. Remy, and D. Thieffry, “Petri net modelling of biological regulatory networks,” *Journal of Discrete Algorithms*, vol. 6, no. 2, pp. 165–177, 2008.

[48] S. Grunwald, A. Speer, J. Ackermann, I. Koch, *et al.*, “Petri net modelling of gene regulation of the duchenne muscular dystrophy,” *Biosystems*, vol. 92, no. 2, p. 189, 2008.

[49] M. A. Savageau and R. Rosen, “Biochemical systems analysis: a study of function and design in molecular biology,” *Addison-Wesley Reading*, vol. 56, Mar 1976.

[50] V. N. Reddy, M. N. Liebman, and M. L. Mavrovouniotis, “Qualitative analysis of biochemical reaction systems,” *Computers in Biology and Medicine*, vol. 26, pp. 9 – 24, Jan 1996.

[51] R. Hofestadt and S. Thelen, “Quantitative modeling of biochemical networks,” *In Silico Biology*, vol. 1, pp. 39 –53, 1998.

[52] H. Kitano, “Systems biology: A brief overview,” *Science*, vol. 295, no. 5560, pp. 1662–1664, Mar 2002.

[53] I. Rigoutsos and G. Stephanopoulos, “Systems biology,” *Oxford University Press*, Sep 2006.

[54] W. Bosl, “Systems biology by the rules: hybrid intelligent systems for pathway modeling and discovery,” *Journal of Simulation*, vol. 1, no. 13, pp. 1 – 25, 2007.

[55] V. Wolf, “Modelling of biochemical reactions by stochastic automata networks,” *Electronic Notes in Theoretical Computer Science*, vol. 171, no. 2, pp. 197 – 208, 2007.

[56] M. Heiner, R. Donaldson, and D. Gilbert, “Petri nets for systems biology,” *Symbolic Systems Biology: Theory and Methods. Jones and Bartlett Publishers, Inc., USA (in Press, 2010)*, 2010.

[57] H. El-Samad and H. D. Madhani, “Can a systems perspective help us appreciate the biological meaning of small effects?,” *Developmental Cell*, vol. 21, no. 1, pp. 11–13, 2011.

[58] V. Reddy, M. Mavrovouniotis, M. Liebman, *et al.*, “Petri net representations in metabolic pathways,” in *Proc Int Conf Intell Syst Mol Biol*, vol. 1, pp. 328–336, 1993.

[59] M. Chen and R. Hofestädt, “Quantitative petri net model of gene regulated metabolic networks in the cell,” *In Silico Biology*, vol. 3, no. 3, pp. 347 – 365, 2003.

[60] S. Hardy and P. Robillard, “Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches,” *Journal Bioinformatic Computer Biology*, vol. 4, pp. 595 – 613, Mar 2004.

[61] H. Aliah and M. Zeti-Azura, “Simulation of a petri net-based model of the terpenoid biosynthesis pathway,” *BMC Bioinformatics*, vol. 11, no. 1, p. 83, 2010.

[62] E. Wingender, “Petri net applications in molecular biology,” *In Silico Biology*, vol. 10, no. 1, pp. 1–4, 2010.

[63] S. Hardy and R. Iyengar, "Analysis of dynamical models of signaling networks with petri nets and dynamic graphs," *Modeling in Systems Biology*, pp. 225–251, 2011.

[64] V. Spirin, M. Gelfand, A. Mironov, and L. Mirny, "A metabolic network in the evolutionary context: Multiscale structure and modularity," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8774 – 8779, Jun. 2006.

[65] D. Wild, Y. Ding, A. Sheth, L. Harland, E. Gifford, and M. Lajiness, "Systems chemical biology and the semantic web: what they mean for the future of drug discovery research," *Drug discovery today*, 2011.

[66] N. Meskin, H. Nounou, M. Nounou, A. Datta, and E. Dougherty, "Intervention in biological phenomena modeled by s-systems," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 5, pp. 1260–1267, 2011.

[67] A. H. Wyllie, "Apoptosis: An overview," *British Medical Bulletin*, vol. 53, no. 3, pp. 451 – 465, 1997.

[68] I. Low, Y. Yang, and H. Lin, "Validation of petri net apoptosis models using p-invariant analysis," in *Control and Automation, 2009. ICCA 2009. IEEE International Conference on*, pp. 416–421, IEEE, 2009.

[69] A. Levine, "p53 the cellular gatekeeper for growth and division," *Cell*, vol. 88, no. 3, pp. 323 – 331, Feb. 1997.

[70] M. Lamkanfi, N. Festjens, W. Declercq, T. V. Berghe, and P. Vandenabeele, "Caspases in cell survival, proliferation and differentiation," *Cell Death and Differentiation*, vol. 14, no. 1, pp. 44 – 55, Oct 2006.

[71] C. S. Duckett, V. E. Nava, R. W. Gedrich, R. J. Clem, J. L. V. Dongen, M. C. Gilfillan, H. Shiels, J. M. Hardwick, and C. B. Thompson, "A conserved family of cellular genes related to the baculovirus IAP gene and encoding apoptosis inhibitors," *The EBMO Journal*, vol. 15, no. 11, pp. 2685–2694, Jun 1996.

[72] C. S. Duckett, F. Li, Y. Wang, K. J. Tomaselli, C. B. Thompson, and R. C. Armstrong, "Human IAP-like protein regulates programmed cell death downstream of bcl-xl and cytochrome c," *Molecular and Cellular Biolgy*, vol. 18, no. 1, pp. 608–615, Jan 1998.

[73] M. J. Burek and R. W. Oppenheim, "Cellular interactions that regulate programmed cell death in the developing vertebrate nervous system,cell death and diseases of the nervous system," *Humana Press Inc Totowa*, pp. 145 – 179, 1999.

[74] Q. L. Deveraux and J. C. Reed, "IAP family proteins - suppressors of apoptosis," *Genes and Development*, vol. 13, no. 3, pp. 239–252, Feb 1999.

[75] M. Holcik and R. G. Korneluk, "Functional characterization of the X-linked inhibitor of apoptosis (XIAP) internal ribosome entry site element: Role of la autoantigen in XIAP translation," *Molecular and Cellular Biolgy*, vol. 20, no. 13, pp. 4648–4657, July 2000.

[76] J. C. Wilkinson, E. Cepero, L. H. Boise, and C. S. Duckett, "Upstream regulatory role for XIAP in receptor-mediated apoptosis," *Molecular and Cellular Biology*, vol. 24, no. 16, pp. 7003 – 7014, Aug 2004.

[77] A. D. Schimmer, S. Dalili, R. A. Batey, , and S. J. Riedl, "Targeting XIAP for the treatment of malignancy," *Cell Death and Differentiation*, vol. 13, no. 2, pp. 179–188, Dec 2005.

[78] R. W. G. Watson and J. M. Fitzpatrick, "Targeting apoptosis in prostate cancer: focus on caspases and inhibitors of apoptosis proteins," *British Journal of Urology International*, vol. 96, pp. 30–34, Dec 2005.

[79] A. Finkel, "The minimal coverability graph for petri nets," *Advances in Petri Nets 1993*, pp. 210–243, 1993.

[80] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad, "A symbolic reachability graph for coloured petri nets," *Theoretical Computer Science*, vol. 176, no. 1, pp. 39–65, 1997.

[81] P. Habermehl, R. Meyer, and H. Wimmel, "The downward-closure of petri net languages," *Automata, Languages and Programming*, pp. 466–477, 2010.

[82] E. Best, P. Darondeau, R. Gorrieri, *et al.*, "On the decidability of non interference over unbounded petri nets," in *SecCo: International Workshop on Security Issues in Concurrency*, pp. 16–33, 2011.

[83] P. Darondeau, S. Demri, R. Meyer, and C. Morvan, "Petri net reachability graphs: Decidability status of first order properties," *Logical Methods in Computer Science*, vol. Vol. 8, pp. 1–28, Oct 2012.

[84] R. Dickinson and R. Gelinas, "Sensitivity analysis of ordinary differential equation systems—a direct method," *Journal of computational physics*, vol. 21, no. 2, pp. 123–143, 1976.

[85] S. Marino, I. Hogue, C. Ray, and D. Kirschner, "A methodology for performing global uncertainty and sensitivity analysis in systems biology," *Journal of theoretical biology*, vol. 254, no. 1, p. 178, 2008.

[86] Y.-C. Ho, "Performance evaluation and perturbation analysis of discrete event dynamic systems," *Automatic Control, IEEE Transactions on*, vol. 32, pp. 563 – 572, jul 1987.

[87] R. Suri, "Perturbation analysis: the state of the art and research issues explained via the gi/g/1 queue," *Proceedings of the IEEE*, vol. 77, pp. 114 –137, jan 1989.

[88] H. Choi, V. Mainkar, and K. Trivedi, "Sensitivity analysis of deterministic and stochastic petri nets," *SIMULATION SERIES*, vol. 25, pp. 271–271, 1993.

[89] N. Viswanadham and Y. Narahari, "Coloured petri net models for automated manufacturing systems," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 1985–1990, IEEE, 1987.

[90] C. Feng and A. Desrochers, "Sensitivity analysis of a deds by a petri net-based perturbation method," in *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pp. 650 –655 vol.1, oct 1992.

[91] C. Feng and A. Desrochers, "Sensitivity analysis of a fms by a petri net-based perturbation method," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 564 –569 vol.3, may 1993.

[92] M. Alison, *The Cancer Handbook 2 volume set*. Wiley-VCH, 2004.

[93] E. Cerutti, M. F. Campagnoli, M. Ferretti, E. Garelli, N. Crescenzio, A. Rosolen, A. Chiocchetti, M. J. Lenardo, U. Ramenghi, and U. Dianzani, "Co-inherited mutations of fas and caspase-10 in development of the autoimmune lymphoproliferative syndrome," *BMC Immunology*, vol. 8, no. 1, pp. 8–28, 2007.

[94] M. Hubeau, F. Ngadjeua, A. Puel, L. Israel, J. Feinberg, M. Chrabieh, K. Belani, C. Bodemer, I. F. A. Plebani, S. Boisson-Dupuis, C. Picard, A. Fischer, A. Israel, L. Abel, M. Veron, J. L. Casanova, F. Agou, and J. Bustamante, "New mechanism of x-linked anhidrotic ectodermal dysplasia with immunodeficiency: impairment of ubiquitin binding despite normal folding of NEMO protein," *Blood*, vol. 118, no. 4, pp. 926 – 935, 2011.

[95] N. N. Danial and S. J. Korsmeyer, "Cell death: Critical control points," *Cell*, vol. 116, no. 2, pp. 205–219, Jan 2004.

[96] P. Liston, N. Roy, K. Tamai, C. Lefebvre, S. Baird, G. Cherton-Horvat, R. Farahani, M. McLean, J. E. Ikeda, A. MacKenzie, and R. G. Korneluk, "Suppression of apoptosis in mammalian cells by NAIP and a related family of IAP genes," *Nature*, vol. 379, no. 6563, pp. 349–353, Jan 1996.

[97] B. P. Eckelman, G. S. Salvesen, and F. L. Scott, "Human inhibitor of apoptosis proteins: why XIAP is the black sheep of the family," *EMBO Reports*, vol. 7, no. 10, pp. 988–994, Oct 2006.

[98] Y. Yao, "A petri net model for temporal knowledge representation and reasoning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 9, pp. 1374 – 1382, Sep. 1994.

[99] S. Hardy and P. Robillard, “Phenomenological and molecular-level petri net modeling and simulation of long-term potentiation,” *BioSystems*, vol. 82, pp. 26 – 38, May 2005.

[100] E. D. Knuth, “The art of computer programming,” *Boston: Addison-Wesley*, vol. 1, 2006.

[101] K. I. Fukuda and T. Takagi, “A pathway editor for literature-based knowledge curation,” in *APBC’04: Proceedings of the second conference on Asia-Pacific bioinformatics*, (Darlinghurst, Australia, Australia), pp. 339–344, Australian Computer Society, Inc., 2004.

[102] A. Regev, W. Silverman, and E. Shapiro, “Representation and simulation of biochemical processes using the π-calculus process algebra,” *in Proc. Pac. Symp. Biocomputation*, vol. 14, no. 9, pp. 1027 – 1047, 2001.

[103] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez, “Representation and simulation of biochemical processes using the π-calculus process algebra,” *in Proc. Pac. Symp. Biocomputation*, pp. 400 – 412, Jan. 2002.

[104] N. Przulj, D. G. Corneil, and I. Jurisica, “Modeling interactome: scale-free or geometric?,” *Bioinformatics*, vol. 20, no. 18, pp. 3508 – 3515, July 2004.

[105] A. D. King, N. Przulj, and I. Jurisica, “Protein complex prediction via cost-based clustering,” *Bioinformatics*, vol. 20, no. 17, pp. 3013 – 3020, November 2004.

[106] S. Wuchty, Z. N. Oltvai, and A.-L. Barabasi, “Evolutionary conservation of motif constituents in the yeast protein interaction network,” *Nature Genetics*, vol. 35, no. 2, pp. 176 – 179, Oct. 2003.

[107] O. Mason and M. Verwoerd, “Graph theory and networks in biology,” *IET Systems Biology*, vol. 1, no. 2, pp. 89 – 119, 2007.

[108] T. Pawson and P. Nash, “Protein protein interactions define specificity in signal transduction,” *Genes and Developement*, vol. 14, no. 9, pp. 1027 – 1047, May 2000.

[109] C. Choi, T. Crass, A. Kel, O. Kel-Margoulis, M. Krull, S. Pistor, A. Potapov, N. Voss, and E. Wingender, “Consistent re-modeling of signaling pathways and its implementation in the transpath database,” *Genome Informatics*, vol. 15, no. 2, pp. 244 – 254, 2004.

[110] M. Poolman, B. Bonde, A. Gevorgyan, H. Patel, and D. Fell, “Challenges to be faced in the reconstruction of metabolic networks from public databases,” *IEE Proceedings on Systems Biology*, vol. 153, no. 5, pp. 379 – 384, 2006.

[111] D.-Y. Lee, R. Zimmer, S. Y. Lee, and S. Park, “Knowledge representation model for systems-level analysis of signal transduction networks,” *Genome Informatics*, vol. 15, no. 2, pp. 234 – 243, 2004.

[112] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano, “Biopathways representation and simulation on hybrid functional petri net,” *In Silico Biology*, vol. 3, no. 3, pp. 389–404, 2003.

[113] M. Heiner, I. Koch, and K. Voss, “Analysis and simulation of steady states in metabolic pathways with Petri nets,” *Journal Bioinformatic Computer Biology*, vol. 1, pp. 15 – 34, 2001.

[114] P. Ramadge and W. Wonham, “Supervisory control of a class of discrete event processes,” *SIAM journal on control and optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[115] W. Wonham and P. Ramadge, “On the supremal controllable sublanguage of a given language,” *SIAM Journal on Control and Optimization*, vol. 25, no. 3, pp. 637–659, 1987.

[116] W. Wonham, “Supervisory control of discrete event systems,” *Lecture Notes at University of Toronto*, 2007. http://www.control.utoronto.ca/people/profs/wonham/.

[117] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Diagnosability of discrete-event systems,” *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555 – 1575, Sep. 1995.

[118] S. Hashtrudi-Zad, R. Kwong, and W. Wonham, “Fault diagnosis in discrete-event systems: framework and model reduction,” *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199 – 1212, Jul. 2003.

[119] A. Saboori and S. Hashtrudi-Zad, “Robust nonblocking supervisory control of discrete-event systems under partial observation,” *Systems and Control Letters*, vol. 55, no. 10, pp. 839 – 848, Oct. 2006.

[120] R. P. Kurshan, “Analysis of discrete event coordination.,” *REX Workshop*, pp. 414–453, 1989.

[121] G. Barrett and S. Lafortune, “Using bisimulation to solve discrete event control problems,” *in Proc. of American Contr. Conf.*, vol. 4, pp. 2337 – 2341, Jun. 1997.

[122] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, “Fault diagnosis in discrete-event systems: Framework and model reduction,” *Automatic Control, IEEE Transactions on*, vol. 48, no. 7, pp. 1199–1212, 2003.

[123] H. Kitano, “Foundations of systems biology,” *MIT Press*, Mar 2001.

[124] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, “Global sensitivity analysis,” *The Primer, John Wiley and Sons*, 2008.

[125] N. Motee, B. Bamieh, and M. Khammash, “Stability analysis of a class of biological network models,” *American Control Conference (ACC)*, pp. 5936–5941, 2010.

[126] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso, “The complex-step derivative approximation,” *ACM Transactions on Mathematical Software (TMOS)*, vol. 29, no. 3, pp. 245 – 262, Sep 2003.

[127] Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro, “An autonomous molecular computer for logical control of gene expression,” *Nature*, vol. 429, pp. 423 – 429, May 2004.

Appendix A

Simulation Codes

A.1 MATLAB Codes

The simulations in this thesis have been prepared using MATLAB version 7.14.0.739 (R2012a) from Mathworks, Operating System: Mac OS X Version: 10.6.8 Java Version: Java 1.6.0_37-b06-434-10M3909 with Apple Inc. Java HotSpot(TM) 64-Bit Server VM mixed mode.

A.1.1 Connecting to the KEGG API Web Service

The following sample code can be used to connect to the KEGG web service API. the WSDL located at 'http://soap.genome.jp/KEGG.wsdl' is an XML that defines the parameters and data structures.

```
wsdlURL = 'http://soap.genome.jp/KEGG.wsdl';


className = createClassFromWsdl(wsdlURL);
```

```
dir @KEGG
methods(KEGG)
kegg = KEGG;
% You can confirm that kegg is an instance of the KEGG using the *class* com
mand classType = class(kegg)
kegg_ids_conv = bconv(kegg, ’ncbi-gi:10047086 ncbi-geneid:14751’)
organisms = list_organisms(kegg)
homo = organisms(1)
pathway_list = list_pathways(kegg, homo.entry_id);
num_pathways = length(pathway_list)
aemp_pathway = pathway_list(17)
aemp_genes = get_genes_by_pathway(kegg, aemp_pathway.entry_id);
num_genes = length(aemp_genes)

aemp_compounds= get_compounds_by_pathway(kegg, aemp_pathway.entry_id);
num_compounds = length(aemp_compounds)

aemp_enzymes = get_enzymes_by_pathway(kegg, aemp_pathway.entry_id);
num_enzymes = length(aemp_enzymes)

aemp_reactions = get_reactions_by_pathway(kegg, aemp_pathway.entry_id);
num_reactions = length(aemp_reactions)

pathway_by_genes = get_pathways_by_genes(kegg, aemp_genes(1:10))
```

```
pathway_by_react = get_pathways_by_reactions(kegg,
aemp_reactions(1:10))

obj_list = [aemp_genes(5:6); {’hsa:83875’}; aemp_compounds(2); aemp_com
pounds(32:33)]; fg_list  = {’#ff6600’, ’#0000ff’, ’#0000ff’, ’#0000ff’,
’#ff0000’, ’#ff 0000’}; bg_list  = {’#99ccff’, ’yellow’,  ’#ff6633’,
’#ff0000’, ’#ccffff’, ’#cc ffff’};

pathway_map_colored = color_pathway_by_objects(kegg,
aemp_pathway.entry_id,. ..
                        obj_list, fg_list, bg_list)

pathway_map_html = get_html_of_colored_pathway_by_objects(kegg,...
                    aemp_pathway.entry_id, obj_list, fg_list, bg_list)

%Displaying Pathway Maps
web(pathway_map_html)

[x,cmap] = imread(pathway_map_colored);

hfig = figure(’Colormap’, cmap);

hax = axes(’Parent’, hfig);
```

```
himg = image(x, 'Parent', hax);

set(hax, 'Visible', 'off') scaleimagefigure(hfig, hax, himg);
```

A.1.2 Petri net Reachability

The following sample code takes three inputs. *Pre* is the matrix representing the inbound arc weights, and *Post* is the matrix of outbound arc weights. Together these two matrices define the structure of the Petri net. Vector M_0 is the initial marking for the Petri net. The outputs of the code are the adjacency matrix, which defines all possible firings, as well as *RM*, which is the list of reachable markings. This code can be used for converting the Petri net into a generator.

```
function [A,RM] = reachability(Pre,Post,M0)

% function reachability finds a graph of reachable markings
% to given marked bounded Petri Net
% A -  Adjacency matrix
% (A(i,j) means oriented arc from vertex i to vertex j)
% value of A(i,j) means index of fired transition
% RM -  Matrix of reachable states
% (each column represents marking of one state)

RM=M0;
A=[0];
```

```
[nofp,noft]=size(Pre);

C=Post-Pre;
i=0;
while i < size(RM,2) && i < 2500
%main loop
%while (number of already inspected states)<(number of existing states)
i=i+1;
    %generation of vector of  enabled transitions
for k=1:noft
     x(k)=all(RM(:,i) >= Pre(:,k));   % x - enabled transition
end
fx=find(x);

for k=1:size(fx,2)
bb = RM(:,i)+C(:,fx(k));
mat_bb=[];
for j=1:size(RM,2)
mat_bb=[mat_bb,bb];
end;
v=all(mat_bb == RM);
j=find(v);
if size(j,2)>1
                    sprintf(’State is duplicated’)
```

```
end

if any(v) %state already exists

A(i,j)= fx(k);

else %state does not exist

RM=[RM,bb];

A(size(A,1)+1,size(A,2)+1)=0;

A(i,size(A,2))=fx(k);

end;

end;

end %main loop


max(RM(2,:))
```