UNIVERSITY OF SEVILLE
DEPARTMENT OF COMPUTER SCIENCE
AND ARTIFICIAL INTELLIGENCE

Doctoral Thesis

# P Systems, a Computational Modelling Framework for Systems Biology

Francisco José Romero Campero

Supervisor:

Dr. Mario J. Pérez Jiménez

Dr. Marian Gheorghe

A toda mi familia, especialmente a
mis padres, José y Loli, y
a mi hermana, Mariani. También
recordar a mi abuela y tita que me
incentivaban a subirme al limonero en
lugar de reñirme. A mi abuelo, por hacerme
observar al trigo dentro de la espiga.

# Acknowledgments

I would like to thank the myriad of people who made this thesis possible. Their time, effort and support were invaluable to the work presented here. Many people contributed time, ideas, and inspiration without which I would never have succeeded. Here, I would like to thank this army of indispensable people.

I would primarily like to thank my supervisors, Mario J. Pérez-Jiménez and Marian Gheorghe, whose guidance and contributions made this thesis possible. Under their supervision, I have learned so much about how to do research. Without their council and encouragement I would never have made it this far and I would not have gained as much potential for the future.

I would additionally like to acknowledge all of my co-authors whose work has been indispensable. Without the following people, the work presented here would not have been possible: John M. Auld, Francesco Bernardini, Daniela Besozzi, Luca Bianco, Miguel Cámara, Paolo Cazzaniga, Rodica Ceterchi, Smitha Cheruku, Gabriel Ciobanu, Miguel Ángel Gutiérrez-Naranjo, Natalio Krasnogor, Ravie C. Muniyandi, Isabel Nepomuceno-Chamorro, Juan A. Nepomuceno-Chamorro, Andrei Păun, Gheorghe Păun, Dario Pescini, Agustín Riscos-Núñez, Peter Siepmann and Sergey Verlan.

I would also like to thank Edgar Goodaire and Brian Sleeman for introducing me for the first time to the fascinating field of Mathematical/Computational Systems Biology during the summer school AARMS2004 held in Memorial University of Newfoundland, St. John's, Newfoundland (Canada). Special thanks to Andrew Hughes for reading this text and suggesting some changes which allowed to improve its readability in English.

Additionally, I would like to thank specially all of my family and friends whose love and support were essential in making it through all these years. I would especially like to thank my parents, José Romero-Turrillo and María Dolores Campero-Ortiz, and my sister Mariana Romero-Campero whose unwavering confidence in my abilities gave me the courage to always keep trying.

Finally, I would like to acknowledge that the research presented here was supported by Fundación Cámara, II Plan Propio de la Universidad de Sevilla, an FPU grant from the Spanish Ministry of Education and Science, projects TIC-2002-04220-C03-01, TIN-2005-09345-C04-01 and TIN-2006-13425 from the Spanish Ministry of Education and Science and project TIC-581 from the Junta de Andalucía.

# Contents

# Part I.

# Introduction

# 1. Introduction

The young field of *Systems biology*[1] aims to deepen the understanding of cellular level dynamics arising over time from the interactions between different cellular systems. In this respect a systematic approach is taken. In contrast, the classical study of cellular processes consisted of a reductionist approach which intended to understand the functioning of cellular dynamics by identifying and characterising each one of their molecular constituents. This approach did not produce the expected understanding uncovering the fact that the functioning of cellular systems arises as an emergent process from the interactions between their different components. Due to the complexity of cellular processes and to the huge amount of data produced by experimentalists computational/mathematical modelling, simulation and analysis are essential techniques in this field. Systems biology is presented in recent textbooks [3, 87] and article collections [25, 70, 120].

Systems biology constitutes a purely interdisciplinary field aiming to merge classical biology, computer science and mathematics. Ideally it will produce a new generation of scientists able to understand and apply concepts, techniques and sources of inspirations coming from any of the three classical fields enumerated above into any of the others. In this respect Systems biology does not consists of the application of computational and mathematical techniques to biology, as it is the case in *Bioinformatics*, it rather aims at the merging of all these disciplines.

Systems biology is closely related to the also young and growing research field of *synthetic biology* [15, 98]. The final goal when modelling a cellular system is to obtain the necessary understanding to be able to control its functioning. A deeper knowledge of the organisation and functioning of cellular processes will allow us to engineer our own cellular systems exhibiting a desired behaviour or producing a prefixed output. This constitutes the main aim in synthetic biology. This opens a challenging range of applications in pharmacology and novel treatments to important diseases like cancer.

Ordinary differential equations (ODEs) are the most widely used modelling ap-

[1] This term was coined by [59] and is not yet fully established.

proach in systems biology so far. However, it has been reported that the macroscopic, deterministic and continuous approach followed by ODEs is questionable in cellular systems with low number of molecules, slow reactions and heterogeneous structures [8, 41]. These conditions are common is some cellular systems. For these cases mesoscopic, stochastic and discrete approaches have been proposed. Due to the complexity of cellular systems these latter approaches have been implemented in formal and high level computational frameworks with the aim of producing relevant, understandable, computable and extensible models. Some of the computational frameworks used to model cellular systems are Petri nets [46], process algebra ($\pi$-calculus [101], bioambients [100], brane calculus [26], $\kappa$-calculus [30], etc.), state charts [52], agent based systems [54], etc.

Although each of these computational frameworks captures some of the information regarding cellular systems and their components, none fully integrates the dynamics and structural details of the systems. One of the main points which is neglected in most computational approaches is the key role played by membranes and compartmentalisation in the structure and functioning of living cells. There have been several attempts in specifying and simulating membranes and compartments in cellular systems, for example the process algebra bioambients [100] and brane calculus [26]. Nevertheless, it has been discussed that the models developed using process algebra can be obscure, non intuitive and difficult to understand [101].

In this work we present P systems as a high level computational modelling framework which integrates the structural and dynamical aspects of cellular systems in a comprehensive and relevant way while providing the required formalisation to perform mathematical and computational analysis. All the modelling approaches mentioned above were well established formalisms coming from different sources of inspiration before being applied to model cellular systems. For instance, the $\pi$-calculus was introduced to specify mobile concurrent processes that interact through communication channels [80]. In contrast, P systems are an unconventional model of computation inspired directly from the functioning and structure of the living cell [91]. Therefore, the concepts in P systems are more similar to those used in molecular cell biology than the abstractions of other formalisms.

Roughly speaking, the three essential components of a P system are a cell-like membrane structure containing a number of membranes arranged in an hierarchical way and delimiting regions or compartments, multisets of objects and strings placed inside the compartments delimited by membranes and rewriting rules associated with specific compartments describing the evolution of the objects and strings placed inside these compartments.

Rather than being an alternative to more classical modelling frameworks, like ODEs, P systems constitute a complementary approach to be used when the classical modelling approaches fail to specify and simulate cellular systems correctly. In contrast to differential equations, P systems explicitly represent the discrete character of the quantity of components of a cellular system by using rewriting rules on multisets of objects which represent molecules, and strings which describe the organisation of genes on the genome. The inherent stochasticity, external noise and uncertainty in cellular systems is captured by using stochastic strategies based on Gillespie's theory of stochastic kinetics [40, 41, 42, 43, 44].

The key differential feature of P systems is the so called *membrane structure* which represents the compartmentalisation in the structural organisation of living cells. In this work we will show how by using membrane structures one can take into account the key role played by membranes and compartmentalisation in the functioning of cellular systems. For instance, in chapter 6 selective uptake of molecules from the environment will be studied; in chapter 7 signalling at the cell surface will be specified and simulated; and, finally, in chapter 8 colonies of interacting bacteria which communicate by sending and receiving diffusing signals will be investigated. In all these case studies membranes in P system specifications will specify the relevant regions of the corresponding cellular system under study.

## 1.1. Organisation of this thesis

This text is organised into four parts covering different aspects of the work developed by the author using P systems as a computational framework for the specification and simulation of cellular systems.

- **Part I: Introduction**

  The first part of this text constitutes the introduction to this work. It consists of two chapters.

  This first chapter stands as an extended abstract of the thesis together with a description of the organisation of the text, a brief summary of each chapter and the outcomes of the work presented in this text.

  The second chapter presents a brief overview of modelling in cellular systems within the framework of systems biology. The general concept of a model and the modelling process in cellular systems are introduced in sections 2.1 and 2.2. In this chapter, several modelling approaches are discussed. Namely, differential equations in section 2.5, stochastic modelling in section 2.6 and computational models in section 2.7. More precisely, in this last point, a short discussion on Petri nets and $\pi$-calculus is given.

- **Part II: Modelling Framework**

  The second part of this text constitutes the core of the thesis. In this part a computational modelling framework based on P systems is discussed in three chapters.

  In chapter 3, P systems are presented as a modelling approach to cellular systems fulfilling the requirements of a *good* modelling framework: relevance, understandability, extensibility and computational/mathematical tractability. The specific variant of P systems and the main definitions used in this work are presented in section 3.2. The main concepts are P system specifications and families of P system models. In section 3.3 we introduce an extension of the well known Gillespie algorithm to the compartmentalised structure of P systems, the *Multicompartmental Gillespie algorithm.* Finally, a methodology to perform probabilistic model checking on P system models using PRISM is discussed in section 3.4.

  Chapter 4 presents the specification principles in P systems. The chapter starts by discussing how compartments are described using P system specifications. Section 4.2 presents an enumeration of the protein-protein interactions that can be specified using P system rewriting rules. The description of transcription networks is discussed in the following section offering two possibilities. The processes involving proteins, genes and mRNA can be represented using either rewriting rules on multisets of objects (section 4.3.1) or using rewriting rules on multisets of objects and strings (section 4.3.2). Finally, section 4.4 introduces a method to translate into P systems cellular models specified using SBML (Systems Biology Markup Language), a machine-readable language, derived from XML, for representing models of biochemical reaction networks.

  In chapter 5 we discuss the concept of a P system module to describe cellular modules. In section 5.1 this concept is introduced to mimic a network of interacting molecules performing a specific task and with a relatively independence of the rest of the system. A basic set of P system modules representing the most important elementary subsystems in cellular systems are described in section 5.2. Finally, section 5.3 uses autoregulation in transcription networks to illustrate how P system modules can be combine to produce more complex modules.

- **Part III: Case Studies**

  In the third part of this thesis three different case studies are presented in order to illustrate the modelling principles and techniques discussed in Part II, namely prokaryotic gene regulation in chapter 6, signal transduction in chapter 7 and quorum sensing in the bacterium Vibrio fischeri in chapter 8. These case studies also show the suitability of P systems as a computational modelling framework.

In chapter 6 we describe the basic modelling principles for prokaryotic gene regulation within the P system modelling framework using the Lac operon regulation system as a case study. A brief description of the gene expression control in the Lac operon is presented in section 6.1. According to this description a P system specification and a family of P system models are developed in section 6.2. Finally, in section 6.3, an analysis of the behaviour of the Lac operon system under different environmental conditions is discussed.

Chapter 7 presents a brief discussion of the general principles of signal transduction systems. A deterministic version of the *Multicompartmental Gillespie's algorithm*, referred to as the *Deterministic Waiting Times Algorithm*, is introduced in section 3.3 and will be used in the rest of the chapter. Two different signal transduction systems are studied here: the Epidermal Growth Factor signalling cascade, section 7.3, and FAS induced apoptosis, section 7.4. A P system specification of each of these systems is developed as well as a family of P system models which will allow us to study the robustness of the system in the case of the EGFR signalling cascade, and to check the validity of various hypotheses about different protein-protein interactions in the case of the FAS-induced apoptosis.

The last case study is presented in chapter 8. Here we will study a communication mechanism in colonies of bacteria termed *quorum sensing*. A brief description of the quorum sensing system in the marine bacterium Vibrio fischeri is introduced in section 8.1. In section 8.2 some principles for the development of specifications and models for colonies of bacteria within the P system modelling framework are presented in general, and, in particular, for the quorum sensing system in Vibrio fischeri. Finally, in section 8.3 various P system models describing colonies of different sizes will be analysed.

- **Part IV: Conclusions**

The last part of the text constitutes the conclusions of this work. Here we discuss some achievements and limitations of the use of P systems as a computational modelling framework in systems biology. Some future work and directions are also presented.

## 1.2. Outcomes of this work

Most of the technical contributions in this thesis are published in international journals. The unpublished parts have been submitted recently or will be submitted shortly to relevant journals in the field. Various further publications report early ideas, and approaches related to the use of P systems as a computational modelling framework for systems biology.

Most of the simulations performed in this thesis use an extension of Gillespie algorithm called Multicompartmental Gillespie algorithm. The use of Gillespie's stochastic kinetic theory in P systems was first presented in the following paper published in *Transactions on Computational Systems Biology.*

- Pérez-Jiménez, M.J., Romero-Campero, F.J. (2006) P Systems, a new computationl modelling tool for systems biology, *Transactions on Computational Systems Biology VI, LNBI*, **4220**, 176 – 197.

Related to the adaption of P systems to the modelling of cellular systems, we enumerate below two workshop papers which collect some preliminary investigations on the use of bounded parallelism instead of the classical strategies based on maximal parallelism in P systems.

- Bernardini, F., Romero-Campero, F.J., Gheorghe, M., Pérez-Jiménez M.J. (2006) A Modeling Approach Based on P Systems with Bounded Parallelism, *Lecture Notes in Computer Science*, **4361**, 49 – 65.
- Bernardini, F., Gheorghe, M., Krasnogor, N., Muniyandi, R.C., Pérez-Jiménez, M.J., Romero-Campero, F.J. (2005) On P Systems as a Modelling Tool for Biological Systems, *Lecture Notes in Computer Science*, **3850**, 114 – 133.

An investigation towards the use of probabilistic and symbolic model checking on P system models was reported in the following workshop paper.

- Romero-Campero, F.J., Gheorghe, M., Bianco, L., Pescini, D., Pérez-Jiménez, M.J., Ceterchi, R. (2006) Towards Probabilistic Model Checking on P Systems Using PRISM, *Lecture Notes in Computer Science*, **4361**, 477 – 495.

The previous four papers are mostly covered in chapter 3 when presenting the P system modelling framework.

A comparison with other computatinal modelling frameworks in sytems biology, more specifically $\pi$-calculus was discussed in the next paper published in *Progress in Natural Science.*

- Romero-Campero, F.J., Gheorghe, M., Ciobanu, G., Auld, J., Pérez-Jiménez, M.J. (2007) Cellular modelling using P systems and process algebra, *Progress in Natural Science*, **17** (4), 375 – 383.

The specification principles for protein-protein interactions and transcription networks were first presented in the following paper published in *Biosystems*.

- Romero-Campero, F.J., Pérez-Jiménez, M.J. Modelling Gene Expression Control Using P Systems: The Lac Operon, A Case Study, *Biosystems* in press.

The discussions in chapter 4 follow the ideas presented in the previous paper. Chapter 6 describes the gene expression control in the lac operon. This system was used as a case study to illustrate the specification principles introduced in the paper above.

The two case studies presented in chapter 7 illustrating how signal transduction systems can be specified and simulated using P systems were discussed in the two following papers. The first paper is published in *Progress in Natural Science* and the second one is a workshop paper.

- Chereku, S., Paun, A., Romero-Campero, F.J., Pérez-Jiménez, M.J., Ibarra, O.H. (2007) Simulating Fas-induced apoptosis by using P systems, *Progress in Natural Science*, **17** (4), 424 – 431.
- Paun, A., Pérez-Jiménez, M.J., Romero-Campero, F.J. (2006) Modeling Signal Transduction Using P Systems, *Lecture Notes in Computer Science*, **4361**, 100 – 122.

Finally, the next paper published in *Artificial Life* studies the use of P systems to model interactions in colonies of bacteria and, in particular, covers the case study presented in chapter 8 consisting of the quorum sensing system in the bacterium Vibrio fischeri.

- Romero-Campero, F.J., Pérez-Jiménez, M.J. (2008) A Model of the Quorum Sensing System in Vibrio Fischeri Using P Systems, *Artificial Life*, **14**, in press.

# 2. Cellular Modelling

*Without a recognition of logical fictions, without a comparison of reality with the purely imagined world of the absolute and immutable, without a constant counterfeiting of the world by means of numbers, man could not live. To recognise untruth as a condition of life; that is certainly to impugn the traditional ideas of value in a dangerous manner, and a philosophy which ventures to do so, has thereby alone placed itself beyond good and evil.*

*Friedrich Nietzsche, Beyond Good and Evil*

## 2.1. Reality and Models

The use of models is intrinsic to any scientific activity. Scientists regularly use abstractions of the reality such as diagrams, graphs, plots, relationships, laws, etc. with the aim to describe and understand the reality they are examining. Indeed, our theories and hypotheses about biological objects and systems are in one sense just models [124].

Generally, in the modelling process one can discriminate between the domain of ideas, thoughts, or other mental constructs and the domain of observations or data. Nonetheless, these two fields are linked in an iterative cycle. Our mental picture and understanding of the world is first suggested by experimental data, then we improve our hypotheses, theories and models of the world by carrying out experiments that produce data. This data is used to generate new hypotheses or mental constructions that in turn suggest the design of the next set of experiments to perform [64, 65].

Although biologists are familiar with modelling, quantitative computational mathematical models have lain outside the mainstream due to the lack of techniques from both experimental and theoretical/computational sides. Nonetheless, at the end of the last century extraordinary advances were achieved in both computer science and biology. For example the development of high-throughput technology like the ability to effect high-quality and quasi continuous optical images of cells and the application of machine learning to estimate parameters of stochastic models. The progress in molecular cell biology along with the development of computer science has reached the

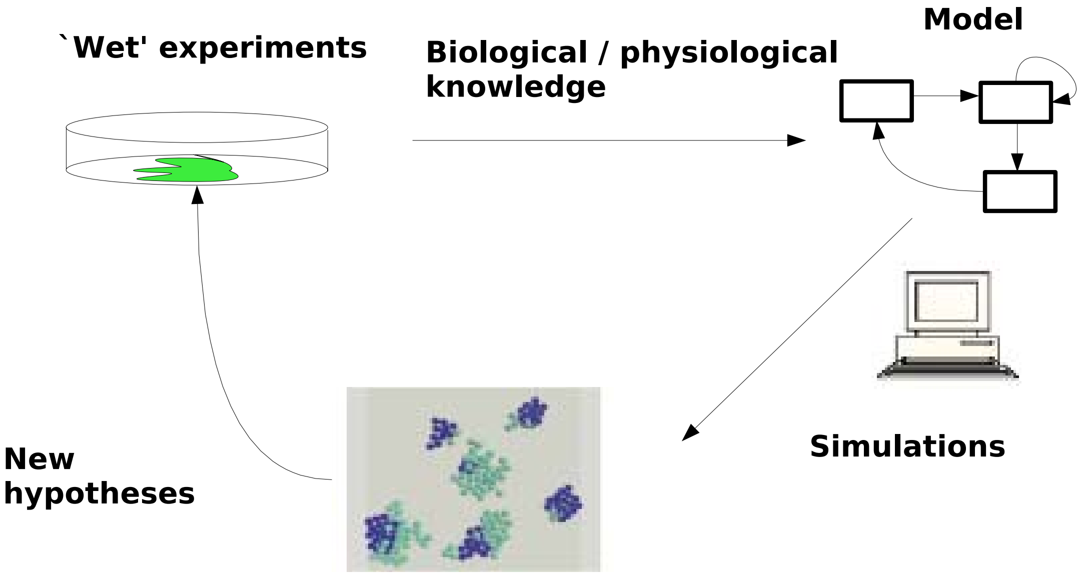

Figure 2.1.: The iterative relationship between the domain of ideas and data

point where each one can benefit from the other one, integrating computer science and biology: computational systems biology [25].

These new advances have made possible the enumeration of the components of cellular systems on a large scale. Nonetheless, computational systems biology is not focused on the enumeration of the components themselves, but rather on the nature of the links that connect them and the functional states of the networks that result from the assembly of such links. The complexity and apparent messiness of molecular interactions in cellular systems makes necessary the development of models able to provide a better understanding of the dynamics and properties of the systems. In this respect, the language of cells is much richer than we had supposed, and we are now well placed to decode it.

A model, an abstraction of the real-world onto a mathematical/computational domain, highlights some key features while ignoring others that are assumed to be irrelevant. Therefore, a model should not be seen or presented as representations of the *truth*, but instead as a statement of *our* current knowledge of the phenomenon under research. A model is often more useful when proved to be wrong, since it shows that our current understanding of the phenomenon studied does not match the reality. Thus, it helps experimentalists as a way to decide which experiments are necessary to advance understanding [25].

In a cellular model it is desirable to have at least four properties: relevance, understandability, extensibility and computability [101].

- **Relevance:** A model must be relevant capturing the essential features of the phenomenon investigated. It should present a unifying specification of the different components that constitute the system, the interaction between them, their dynamic behaviour as well as the physical structure of the system itself.

- **Understandability:** The abstract formalisms used to model cellular systems should correspond well to the informal concepts and ideas from molecular biology. A model should provide a better and integrated understanding of the real cellular system instead of producing a complicated and hard to decipher formalism.

- **Extensibility:** In a cellular model we should be able to identify easily its different components so they can be rearranged, duplicated, composed, etc. in an easy way to produce other models. Models of cellular systems should also be extensible to higher levels of organisations, like tissues, organs, organism, etc. Our knowledge of cellular systems continues to expand and change. In order to handle this continuous supply of new discoveries a model should be adapted to incorporate new information without having to develop a completely new model from scratch.

- **Computability and Mathematical tractability:** It should be possible to implement a model in a computer so that we can run simulations to study the dynamics of the system by manipulating experimental conditions in the model without having to perform complex and costly experiments. The computability of the model also allows us to perform model checking and similar techniques to infer and study qualitative and quantitative properties of the system in an automatic way. In this respect, the model should be mathematically tractable. That is, it should be possible to perform mathematical analysis on it.

There are primarily three uses of models in science:

- **Understanding:** Models are used as a formal framework for summarising and synthesising large quantities of data. This allows us to organise and integrate partial empirical research with the aim of identifying a lack of knowledge in specific areas.

- **Prediction:** Having a model able to make an experimentally testable prediction about the dynamical behaviour of the systems under investigation is one the main goals of modelling itself.
- **Control:** The understanding and capability of prediction acquired using models lead to the possibility to build, constrain or manipulate real cellular systems so that they produce a desirable output or behaviour.

## 2.2. The Modelling Process

Modelling is real world problem solving and so the development of a model is a hard process prone to failure where one has to reconsider many times the assumptions, simplifications, etc. made at different points. The modelling process is a semiformal set of rules that guides us to produce a model, formulate it in a formal language, implement it on a computer and derive properties of the system under research.

- **System identification and delimitation:** The first step of the modelling process consists of stating the specific part of the system one wants to model, the objectives to achieve, the questions to be addressed and how the model will be validated and analysed. It is not trivial to decide what part of the whole phenomenon we are interested in.

  Many cellular systems are very complex with many interactions and links, not fully understood and difficult to delimit. This step involves making some simplifying assumptions to have a clear image of what we want to study and the question we want to answer. In this respect this is a crucial step in the process since we can hardly design a good model without a clear delimitation of the phenomenon.

- **Definition and formal formulation:** Once we have delimited the system to be modelled it is necessary to specify or translate the *informal* description of the components of the system, and the question to be addressed into a formal language. This step requires that vague concepts and loose relations be made definite in a formal language.

  There are many formal frameworks for the specification of cellular systems ranging from graphs, to differential equations, to stochastic processes, to computational approaches, etc. Each one has its own set of advantages and disadvantages being more suitable to be applied to model different systems. Depending on the system to be modelled and the type of questions one wants to answer some formalisms are more suitable than others. There does not yet exist so far any unique unifying modelling framework that can be used in any cellular system. Therefore this step implies taking the crucial decision of which formal approach is going to be used. This decision may have to be revised and changed later on as some approaches are inappropriate in certain circumstances and can influence the predictions of the model.

- **Implementation:** This step consists of the activities in which the formal model is implemented using computer code. At this stage one must verify that the computer algorithms and code are correct with respect to the formal model. Since models of cellular systems can easily involve many parameters, variables and structures this is not an easy step. Most problems that arise at this point are a concern of software engineering.

- **Validation and calibration:** As mentioned earlier cellular systems depend frequently on many parameters. Some of these parameters can be obtained using experiments, nonetheless, others can not be measured in the lab or are very expensive to estimate. Therefore before simulations can be performed, we need to *calibrate* our model by obtaining estimates for missing parameters and validate it against the expected behaviour of the system. This requires comparing trial results or simulations obtained using the model with real data coming from the lab or with data generated using *trustable* methods. This stage consist of an iterative process in which a candidate set of parameters is proposed, some simulations are generated and on the basis of some metric of closeness to the desired behaviour a new set of parameters is tested. If a satisfactory set of parameters is not found some previous assumptions like the structure of the model, the modelling approach, etc. must be reconsidered.

- **Analisis and testing:** Once an accurate set of parameters has been found and our model has been validated one can address the questions that motivated the study of the system by using the model. There exists different possible analysis methods depending on the type of model developed which can range from simple generation of simulations by running a computer program to complicated statistical and mathematical argument and model checking.

## 2.3. Systems Biology

Life is one of the most complex phenomena in the universe and biology, the science of life, is one of the most challenging research fields ever. For a long time biologists have

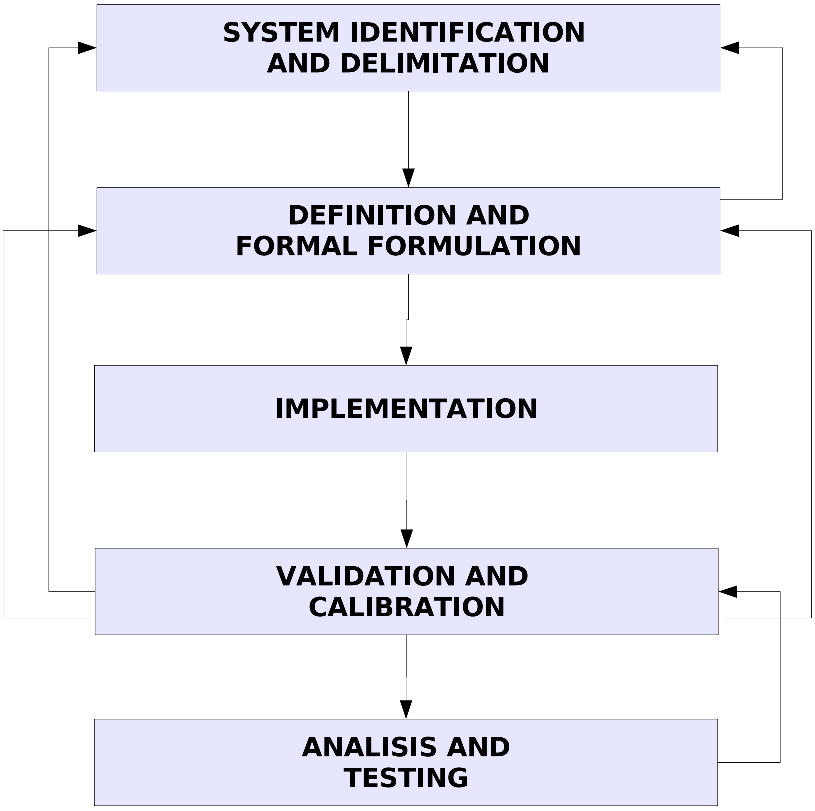
SYSTEM IDENTIFICATION AND DELIMITATION
DEFINITION AND FORMAL FORMULATION
IMPLEMENTATION
VALIDATION AND CALIBRATION
ANALISIS AND TESTING

Figure 2.2.: The Modelling Process

assumed a reductionist approach in biology and have tried to understand life by looking at one little part of the cell at a time. For instance, they have elucidated the structure of proteins, the structure of DNA and RNA, the principles of DNA replication as well as transcription and translation and the structure and functioning of cell membranes.

There has been also a revolution in the experimental techniques and methodologies developing new high-throughput techniques. These new methodologies allow us to measure the expression level of all genes of a cell at the same time with a reasonable temporal resolution. Fluorescence labeling and sophisticated microscopic techniques allow tracing individual molecules within a single cell. Although these techniques are these days still very expensive they have produced fine-grained knowledge of cellular components and processes in time and space which is an important prerequisite for the further elucidation of cellular regulation, see Figure 2.3.

Nevertheless, all these advances have proved wrong the initial hope that an understanding of the individual cellular components would be enough to understand how cells work. It turned out that the whole cell functioning is more than just the sum of the functionality of their components. In this respect, recently it has been proposed that to really understand life it is not sufficient to look at a single protein or gene at a time. It is necessary to use a systematic approach which integrates the functioning of thousands of proteins, genes and membranes in a dynamically changing environment.

Besides, the new techniques and methodologies produce a massive amount of data which is impossible to analyse without the aid of very powerful computers. The huge advances in computer science in the last century has made available computers able to process this information. Advances in mathematical methodologies has also made it possible to infer new insights into cellular systems.

Summing up, the scientific community is posed now to develop a systematic research of cells, tissue, organs and organisms and of mainly cellular processes such as cellular communication, cell division, homeostasis, and adaptation. The approach has been termed Systems Biology and it relies on the integration of biology, mathematics and computer science.

Systems biology is the quantitative study of biological systems, aided by technological advances that both permit molecular observations on far more inclusive scales than possible even fifteen years ago, and permit computational analysis of such observations.

Systems biology is experimentally driven, computational/mathematical driven, and knowledge driven. It is experimentally driven because the complexity of biological systems is difficult to penetrate without large-scale coverage of the molecular underpinnings, it is computationally/mathematically driven because the data obtained from experimental investigations of complex systems need extensive quantitative analysis to be informative; and it is knowledge driven because it is not computationally feasible to analyse the data without incorporating all that is already known about the biology in question.

Modelling is central in Systems Biology. Having a model allows one to analyse it in a variety of ways, but a chief one is to establish those parts of the model that are most important for determining the behaviour in which one is particularly interested. Techniques such as sensitivity analysis are designed for this, and thus indicate to the experimenter which parameters must be known with the highest precision and should be the focus of experimental endeavor. When modelling can be applied effectively it is far cheaper than wet biology and, as well as its use in metabolic engineering, can reduce the reliance on in vivo animal/human experimentation (a factor of significant importance in the pharmaceutical industry). Modelling is also used as a source of hypothesis generation and testing. We may have existing experimental data with which the model is inconsistent, and it is desirable to explore different models to see which changes to them might best reproduce the experimental data. In biology this might, for example, allow the experimenter to test for the presence of an interaction or kinetic property that might be proposed. In a more general or high-level sense, we may use such models to seek evidence that existing hypotheses are wrong, that the model is inadequate, that hidden variables need to be involved, that existing data are inadequate, or that new theories are needed. In kinetic modelling this is often the case with inverse problems in which one seeks to find a model that best explains a time series of experimental data. In this respect Systems Biology uses theoretical models that are only representations of their biological counterpart. Nevertheless, although models are simplifications of the reality they can elucidate possible networks properties and help to discover possible design principles in cellular systems.

Systems Biology is mainly driven by the high potential of its applications. There are many applications in health care, for instance in drug development and validation. A good example is the research carried on these days on the epidermal growth factor receptor which is targeted by a new generation of cancer drugs. Using virtual screening, in which the ability of different synthetic drugs to bind to receptors is tested in silico using appropriate models, the most promising candidate can be determined. Then, it can be synthesised and tested in the lab. The huge advantage in this approach is the enormous speed and favorable economics (scalability) of this approach based on modelling over the actual wet experiments.

## 2.4. Modelling Approaches

There are multiple ways of modelling cellular systems. A pathway diagram as displayed in standard biochemistry textbooks can be consider as the simplest one. Nevertheless, this type of models lack many of the properties desirable in a good model, as described in section 2.1. This representation does not capture relevant features of the system like rate parameters, number of molecules involved in the reactions, etc. Models of this form are not easily extended or combined and they do not allow much mathematical or computational analysis. Therefore, the necessity of formal mathematical/computational modelling approaches is evident. In the next three sections the most widely used formalisms to model cellular systems are discussed. Here we cannot present an exhaustive classification of the different modelling approaches and will talk only about categories depending on the space scale, type of analysis, level of modelling, etc.

According to the space scale in which the modelling approach is focused models are classified into *macroscopic*, *microscopic* and *mesoscopic* [125].

- **Macroscopic:** In a *macroscopic approach* the system is observed and modelled as a whole; the individual parts of the systems are poorly represented and almost no mechanistic information of how they interact is provided.

- **Microscopic:** In a *microscopic approach* each individual part of the system is represented with a high level of detail. In the case of cellular systems, each molecule is taken into account specifying its position and momenta. This approach turns out to be computationally intractable in most cases.

- **Mesoscopic:** In a *mesoscopic approach* the most relevant individual parts of the system are taken into account but details like position and momenta are neglected. Instead, one focuses on the number of individual components of the system, the statistics of the events and how often they take place. The mesoscopic approach is more tractable than the microscopic one while keeping more relevant information than the macroscopic approach.

Depending on the knowledge available about a system, the analysis that can be performed on them and on the data provided by models they can be classified into *quantitative* and *qualitative* models.

- **Quantitative:** The advent of high-throughput technology allows us to obtain quantitative data on a large scale for cellular systems. This data can be used to estimate parameters making possible the development of models which produce in turn quantitative data and information. The three modelling approaches discussed in the following sections are examples of this type of models.

- **Qualitative:** In spite of the recent advances in high-throughput technology, for many cellular systems only incomplete, non-quantitative, uncertain or unreliable data is available. In this case qualitative models can still provide relevant insights of the dynamical behaviour of the system that is independent of parametric information. Examples of this type of models are *discrete abstractions of the dynamics of differential equations* [13, 14], *boolean networks* [116], *graphs* [12, 83], etc.

Based on the type of quantitative data generated by models and on the character of the specification of the components of the system, one can talk about *discrete* and *continuous* models.

- **Discrete:** In a discrete model the components of the systems are specified as individual and discrete entities; in the case of cellular models one talks about a positive integer number of molecules. The data generated is also discrete.

- **Continuous:** In a continuous model the components of the system are represented by continuous variables; for example in cellular models one talks about the concentration of the different chemical species. The data generated in this case is also continuous.

Models can be classified into *deterministic* or *stochastic* according to the dynamics they exhibit.

- **Deterministic:** Given a state of a deterministic model it is possible to determine unequivocally the next state. Therefore there exists a single possible evolution of the model and just one run of the model is enough to obtain all the necessary information about the dynamics of the model.
- **Stochastic:** In a stochastic model, given a state there exists a set of possible next states. The model can move to one of these states following a probability distribution. Therefore there exists many possible evolutions of the model and in order to get an idea of the statistics of the model one must perform a sufficient number of simulations.

Depending on the starting point of the development of a model, low level or high level, and how the detailed information of the system is incorporated into it, models can be referred to as *top-down* or *bottom-up* models.

- **Top-down models:** In a top-down approach a general view of the system is first formulated. Low-level subsystems are specified as black boxes. Then if necessary and possible each subsystem is specified in greater detail, possibly using black boxes for lower level subsystems. This process is iterative and it finishes when the entire specification is reduced to first-level elements or when a predefined level of detail is reached.
- **Bottom-up models:** In a bottom-up approach the first-level components of the system are first specified in great detail. Then these components are combined to form higher levels subsystems until the complete top-level system is achieved.

The classification presented here is far from being exhaustive. We have only presented a general description of the most relevant classes of models overlooking combinations of the different approaches to produce hybrid models.

## 2.5. Modelling based on Differential Equations

Nowadays ordinary differential equations (ODEs) constitute the most widely used approach in modelling molecular interaction networks in cellular systems. They have been used successfully to model kinetics of conventional macroscopic chemical reactions [55, 115]. Nevertheless the realisation of a reaction network as a system of ODEs is based on two assumptions.

**1.** First, cells are assumed to be well stirred and homogeneous volumes so that concentrations do not change with respect to space. Whether or not this is a good approximation depends on the time and space scales involved. In bacteria it has been shown that molecular diffusion is sufficiently fast to mix compounds. The time needed for a protein to diffuse throughout a bacterium size volume is a few seconds. Therefore if we are interested in transcription/translation processes (minutes), cell cycle (hours), circadian rhythms (one day), etc. in bacteria the well stirred volume assumption is justified. This is not the case in eukaryotic cells where the volume is considerably bigger and it is structured in different compartments like nucleus, mitochondria, golgi body, etc.

**2.** The second basic assumption is that chemical concentrations vary continuously over time in a deterministic way. This assumption is valid if the number of molecules of each specifies in the reaction volume (the cell or the subcellular compartment) are sufficiently large and the reactions are fast. A sufficient large number of molecules is considered to be at least thousands of molecules. This is a common scenario in some eukaryotic cellular systems whereas in bacterial system hundreds and fewer molecules are more common and the continuous variation character of the system does not hold anymore.

Writing and solving numerically a system of ODEs, such as the one in (2.3) describing reaction networks can be largely automated. Each species $i$ is assigned a single variable $X_i(t)$ which represents the concentration of the species at time $t$.

For each molecular species, a differential equation is written to describe its concentration change over time due to the reactions with other species in the system. The rate of each reaction is represented using a *kinetic law*, which commonly depends on one or more rate constants, these kinetic laws are represented by the functions $F_i(X_1, \ldots, X_n)$. In this sense models based on ODEs are referred to as *macroscopic* since they do not represent mechanistic aspects of molecules.

In order to solve a system of ODEs one must impose a set of initial conditions representing the initial concentrations of the different species involved in the system. The combination of differential equations with initial conditions is called a *well-posed initial value problem.* The existence and uniqueness of solution for such a system of ODEs is guaranteed under very weak conditions on the smoothness of the functions $F_i(X_1, \ldots, X_n)$.

| **Property** | **E. Coli** | **Yeast** | **Mammalian Cell** |
|---|---|---|---|
| Cell Volume | $\sim 1 \mu m^3$ | $\sim 1000 \mu m^3$ | $\sim 10000 \mu m^3$ |
| Proteins per cell | $\sim 4 \times 10^6$ | $\sim 4 \times 10^9$ | $\sim 4 \times 10^{10}$ |
| Diffusion time of proteins across the cell volume | $\sim$ 0.1 sec | $\sim$ 10 sec | $\sim$ 100 sec |
| Diffusion time of small molecules across the cell volume | $\sim$ 1 msec | $\sim$ 10 msec | $\sim$ 0.1 sec |
| Size of genome | $4.6 \times 10^6$ *bp* | $1.3 \times 10^7$ *bp* | $3 \times 10^9$ *bp* |
| Size of: Regulator binding site | $\sim$ 10 *bp* | $\sim$ 10 *bp* | $\sim$ 10 *bp* |
| Promoter | $\sim$ 100 *bp* | $\sim$ 1000 *bp* | $\sim 10^4$ to $10^5$ *bp* |
| Gene | $\sim$ 1000 *bp* | $\sim$ 1000 *bp* | $\sim 10^4$ to $10^6$ *bp* |
| Time to transcribe a gene | $\sim$ 2 min | $\sim$ 2 min | $\sim$ 30 min |
| Time to translate a protein | $\sim$ 2 min | $\sim$ 2 min | $\sim$ 30 min |
| Timescale for equilibrium biding of small molecule to protein (diffusion limited) | $\sim 1 \mu sec$<br>($1 \mu M$ affinity) | $\sim 1 sec$<br>($1 nM$ affinity) | $\sim 1 sec$<br>($1 nM$ affinity) |
| Typical mRNA lifetime | $\sim$ 2 - 5 min | $\sim$ 10 min to over 1 h | $\sim$ 10 min to over 10 h |
| Ribosomes per cell | $\sim 10^4$ | $\sim 10^7$ | $\sim 10^8$ |
| Cell generation time | $\sim$ 30 min to several hours | $\sim$ 2 h to several hours | $\sim$ 20 h to nondividing |

Figure 2.3.: Typical parameter values for E. Coli, Yeast and a Mammalian cell [3]

$$\begin{cases} \dfrac{dX_1}{dt} = F_1(X_1, \ldots, X_n) \\ \qquad\quad \ldots \\ \dfrac{dX_n}{dt} = F_n(X_1, \ldots, X_n) \\ \\ X_1(0) = x_0^1, \ldots, X_n(0) = x_0^n \end{cases} \tag{2.1}$$

*Exponential decay law*, *mass action law* and *Michaelis-Menten dynamics* are the most widely used kinetic laws:

- **Exponential decay law** is used to represent degradation, transformation, complex dissociation and similar processes with a single reactant species or first order reactions. This law assumes that the rate of the reaction is proportional to the concentration of the reactant. More specifically, if $X(t)$ represents the species involved in a first order reaction, which follows an exponential decay law, the rate of such a reaction is $k \cdot X(t)$.

- **Mass action law** is used to model reactions involving the collision of two molecules to form a complex. In this case the rate is assumed to be proportional to the product of the concentrations of the reactants. More specifically, if $X_1(t)$ and $X_2(t)$ represents the species involved in the reaction following the mass action law then its rate is $k \cdot X_1(t) \cdot X_2(t)$.

- **Michaelis and Menten** (1913) first explored the elementary reaction mechanism in which an enzyme $E$ binds to a molecule $X$, called substrate, to produce the product $P$ while keeping intact the enzyme $E$, see (2.2). They showed that the rate of the enzyme-catalysed reaction can be described using the term $\dfrac{k_pEX}{K_m + X}$, where $E$ and $X$ represent the concentration of the enzyme and substrate, $K_m = \dfrac{k_r + k_p}{k_f}$ is called Michaelis-Menten constant, and $k_f, k_r, k_p$ are the kinetic constants associated with the binding, dissociation and production reaction, respectively.

$$X + E \underset{k_r}{\overset{k_f}{\rightleftarrows}} X.E \xrightarrow{k_p} P + E \tag{2.2}$$

Once a *well-posed initial value problem* has been formulated one can think of three different possible ways of "solving" it:

- **Analytical:** Under rather restrictive conditions the solution of a set of ODEs can be expressed in terms of elementary functions like exponentials and harmonic functions. The conditions in which this can be done are so restrictive that the dynamics obtained are very poor.

- **Numerical:** One can always approximate the solution of a well-posed initial value problem in a computer using one of the well known numerical integration methods. These methods can be very complicated and always produce an error, meaning that the time series obtained constitutes an approximation of the *real* solution.

- **Qualitative:** The previous possible ways of *solving* a system of ODEs produce quantitative information about the dynamics of the system, whereas sometimes one is only interested in qualitative information. There are very powerful and developed techniques for the analysis of steady states, stability, bifurcation, etc. like, for example, phase planes, vector fields, nullclines, etc.

In order to illustrate the different modelling approaches presented in this chapter we will use the example described graphically in Figure 2.4. Our example consists of two genes, *geneI* and *geneR* encoding a signal, $S$, and a cytoplasmic receptor protein $R$ respectively. The protein $R$ is able to interact with the signal $S$ to produce a complex $C$ which in turn activates the production of signals by direct binding to *geneI*.

The following system of ODE models our example:

$$\begin{cases} \dfrac{dS}{dt} = k_1 - k_3 \cdot S \cdot R + k_4 \cdot C - k_5 \cdot S + \dfrac{k_p \cdot C}{K_m + C} \\[2ex] \dfrac{dR}{dt} = k_2 - k_3 \cdot S \cdot R + k_4 \cdot C - k_6 \cdot R \\[2ex] \dfrac{dC}{dt} = k_3 \cdot S \cdot R - k_4 C \\[2ex] S(0) = 0, R(0) = 0, C(0) = 0 \end{cases} \tag{2.3}$$

The production of signals $S$ and proteins $R$ are assumed to take place at a constant rate $k_1$ and $k_2$ respectively. All the first order reactions are modelled using the *exponential decay law* as in the case of degradation of the signal and protein, $k_5 \cdot S$ and $k_6 \cdot R$, and complex dissociation $k_4 \cdot C$. Complex formation is described using the *mass action law*, $k_3 \cdot S \cdot R$. The increase of signal production by the complex $C$ is modelled using *Michaellis-Menten dynamics*, $\dfrac{k_p \cdot C}{K_m + C}$.

As mentioned earlier homogeneity is a key assumption in the application of ODEs, nonetheless many biological systems display spatially inhomogeneous structures. This

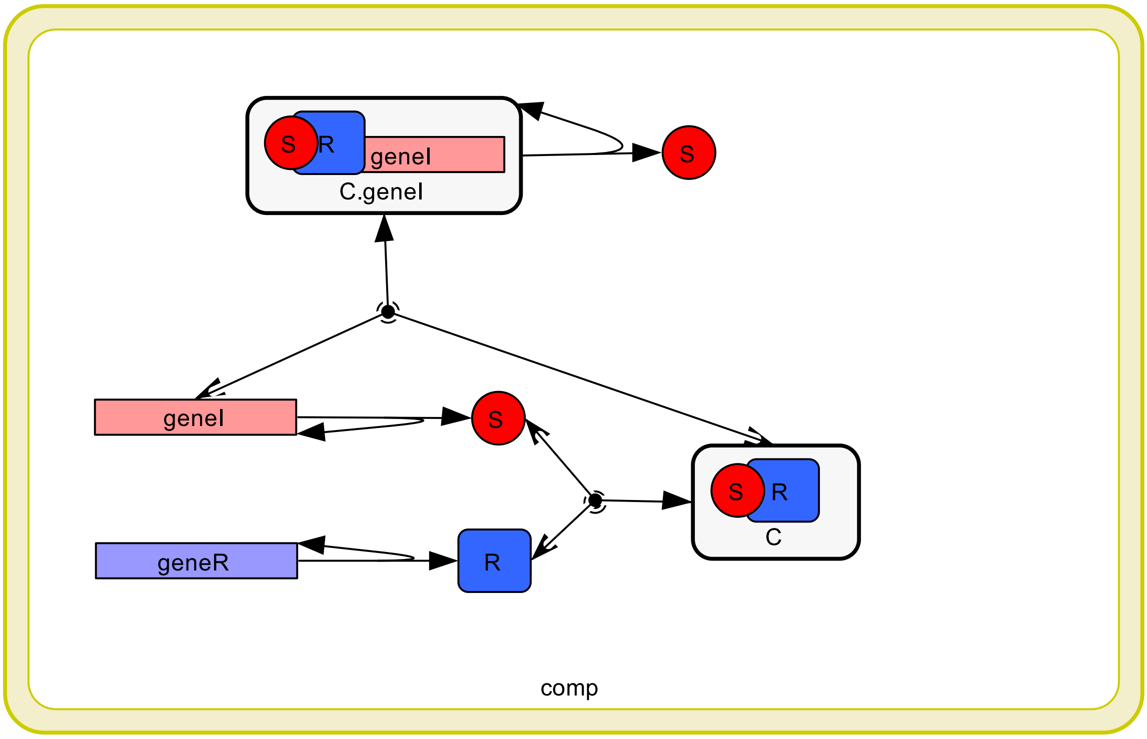

Figure 2.4.: Two genes, *geneI* and *geneR*, encoding a signal $S$ and a protein $R$ which interact to form a complex $C$. This complex in turn increases the production of signals by direct binding to *geneI*.

inhomogeneity is a product of processes in which the time for the movement of molecules across the whole system is long compared to typical reaction times. In these systems ODEs are not applicable anymore, instead partial differential equations (PDEs) appear as a reliable modelling framework. In PDEs the discrete nature of the single molecules forming the system is neglected. Therefore, as in the case of ODEs, PDEs are referred to as a *macroscopic* modelling framework.

The state of the system is given in terms of continuous functions called *fields*, that in this case, depend on space and time. Typically these functions represent densities of molecules, although they can specify other features of the molecules. The fields are linked to microscopic representations of the system state in terms of individual molecules by local averages. Local averages are performed over volume elements that are small compared to the length scales of the structure one is interested in, but large enough to contain a sufficient number of particles such that spatial fluctuations within a volume element are negligible.

Here we focus on the very important class of PDEs called *reaction-diffusion systems.*

Turing introduced the idea that the diffusion of particles together with chemical reactions can lead to the formation of spatiotemporal patterns [123]. The formation of compartments in Drosophila and calcium dynamics in cell aggregates as well as within cells have been successfully described using a reaction-diffusion approach [72]. The application of reaction-diffusion systems to model intracellular protein dynamics is a more recent development [57].

In a reaction diffusion system each field represents the density of one molecular species $i$ as a function of space and time, $X_i(\mathbf{r}, t)$.

$$\begin{cases} \frac{\partial}{\partial t} X_1(\mathbf{r}, t) = D_1 \nabla^2 X_1(\mathbf{r}, t) + F_1(X_1, \dots X_n) \\ \qquad\qquad \dots \qquad \dots \\ \frac{\partial}{\partial t} X_n(\mathbf{r}, t) = D_n \nabla^2 X_n(\mathbf{r}, t) + F_n(X_1, \dots X_n) \\ \\ X_1(0) = x_0^1, \dots X_n, (0) = x_0^n \end{cases} \tag{2.4}$$

The first terms on the right hand sides describe particle diffusion. The parameters $D_i$ are the respective diffusion constants and $\nabla^2$ is the Laplace-operator. In the three spatial dimensions, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ with $\frac{\partial^2}{\partial x^2}$ is the second partial derivative with respect to $x$. The functions $F_i$, as in the case of ODEs, represents the kinetic laws that model the reaction between the molecules of the system.

## 2.6. Stochastic Modelling of Cellular systems

As discussed earlier, continuous and deterministic descriptions are adequate only if the number of molecules are large. At the microscopic level of functioning of cellular processes the interactions between molecules follow the laws of physics. A fundamental result of theoretical statistical physics is the famous $\sqrt{n}$ law, which states that randomness or fluctuation level in a system are inversely proportional to the square root of the number of particles. As a result biochemical systems in living cells with small number of molecules of a few reactant species exhibit discrete and stochastic dynamical behavior, rather than continuous and deterministic [8, 33, 78, 79]. A rapidly growing body of experimental work shows that many cellular systems are controlled by a few molecules, for example, gene expression where there exists only one molecule of DNA a tens of molecules of transcription factor [33].

As a starting point we assume that the system is *well-stirred.* As mentioned before, this is adequate if the distribution of molecules equilibrates on a shorter time scale than the characteristic time scales of the processes under investigation.

The first step for stochastic descriptions of chemical reactions is to define a sufficiently complete set of state variables such that changes only depend on the current state. Assuming that the system is confined to a constant volume $V$, and in thermal equilibrium at some constant temperature it is enough to represent the state of a system using the number of molecules of each reacting species.

Let us consider a system of molecules of $N$ chemical species $\{S_1, \ldots, S_N\}$ interacting through $M$ chemical reaction channels $\{R_1, \ldots, R_M\}$. The state of the system at time $t$ is describe by the state vector $\mathbf{X}(t) = (X_1(t), \ldots, X_N(t))$. We want to study the evolution of the state vector $\mathbf{X}(t)$ given that the system was initially in some state $\mathbf{X}(t_0) = \mathbf{x}_0$.

Each reaction channel $R_j$ is characterised by a *state-change vector* $\mathbf{v}_j = (v_{1j}, \ldots, v_{Nj})$ and a *propensity* $a_j(\mathbf{x})$ in a given state $X(t) = \mathbf{x}$, see [40, 41, 42]. The components of the *state-change vector* $v_{ij}$ represents the change in the molecular population of the species $S_i$ caused by one $R_j$ reaction. Therefore if the system is in state $\mathbf{x}$ and a reaction $R_j$ occurs then the system will jump to the state $\mathbf{x} + \mathbf{v}_j$. The array $(v_{ij})$ is commonly known as the *stoichiometric matrix*.

The propensity is defined so that $a_j(\mathbf{x})dt$ represents the probability that one reaction $R_j$ will occur in the infinitesimal time interval $[t, t + dt)$ given that $\mathbf{X}(t) = \mathbf{x}$. In order to compute the propensity of each reaction channel $R_j$ a stochastic constant $c_j$ is associated with each reaction.

If $R_j$ is a first order reaction of the form $S_i \rightarrow$ *products* its propensity is $a_j(\mathbf{x}) = c_j X_i(t)$.

For a second order reaction of the form $S_i + S_{i'} \rightarrow$ *products* the propensity is computed as $a_j(\mathbf{x}) = c_j X_i(t) X_{i'}(t)$ if $i \neq i'$ or $a_j(\mathbf{x}) = c_j \frac{1}{2} X_i(t)(X_i(t) - 1)$ if $i = i'$.

Any higher order reaction is very unlikely to occur instead a series of second order reactions takes place.

The stochastic constant $c_j$ can be computed using the reaction rate constant $k_j$ used in conventional deterministic chemical kinetics like ODEs. It turns out that $c_j$ is equal to $k_j$ for first order reaction, while for second order reactions $c_j = k_j/V$ if the reactants are different species or $c_j = 2k_j/V$ if they are the same, see [40, 41, 42]. If the rate constants $k_j$ are meassured in concentration units, which is a common case, one has also to divide by Avogrado's number.

At this point, it is worth noting that this approach is referred to as *mesoscopic modelling* [125].

As discussed above here one wants to compute, $P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)$, the probability that the system is in state $\mathbf{X}(t) = \mathbf{x}$, given that $\mathbf{X}(t_0) = \mathbf{x}_0$. The time evolution equation for this probability can be obtained using the laws of probability to write $P(\mathbf{x}, t+dt \,|\, \mathbf{x}_0, t_0)$

as follows:

$$P(\mathbf{x}, t + dt | \mathbf{x}_0, t_0) \quad = \quad \overbrace{P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0) \times [1 - \sum_{j=1}^{M} a_j(\mathbf{x})dt]}^{\substack{\text{The system is already in state } \mathbf{x} \text{ at time } t \\ \text{and no reaction occurs in } [t, t+dt)}} \quad +$$

$$+ \quad \textstyle\sum_{j=1}^{M} \overbrace{P(\mathbf{x} - \mathbf{v}_j, t \,|\, \mathbf{x}_0, t_0) \times a_j(\mathbf{x} - \mathbf{v}_j)dt}^{\substack{\text{The system is one } R_j \text{ reaction removed from state } \mathbf{x} \\ \text{at time } t \text{ and one } R_j \text{ reaction occurs in } [t, t+dt)}}$$

Subtracting $P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)$ from both sides, dividing through by $dt$, and taking the limit $dt \to 0$, we obtain the *chemical master equation (CME)*:

$$\frac{dP(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)}{dt} = \sum_{j=1}^{M} [a_j(\mathbf{x} - \mathbf{v}_j)P(\mathbf{x} - \mathbf{v}_j, t \,|\, \mathbf{x}_0, t_0) - a_j(\mathbf{x})P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)] \qquad (2.5)$$

The CME completely determines the probabilities of the system, but it really is a system of nearly as many coupled ordinary differential equations as there are combinations of molecules that can exist in the system. Therefore it can be solved analytically only in a very few simple systems. Instead of solving the CME one can always construct *numerical realisations* of $\mathbf{X(t)}$, that is, generate trajectories of $\mathbf{X(t)}$ using a Monte Carlo algorithm. Here we present the *Gillespie algorithm* or *stochastic simulation algorithm (SSA)*.

There is a change in the point of view in the SSA with respect to CME. Here one does not focus on computing $P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)$, the probability of the system being in state $\mathbf{x}$ at time $t$, instead one wants to compute $p(\tau, j \,|\, \mathbf{x}, t)$ the probability that when the system is in state $\mathbf{x}$ at time $t$ the next reaction to occur in the infinitesimal interval $[t + \tau, t + \tau + d\tau)$ will be an $R_j$ reaction.

If we denote as $P_0(\tau \,|\, \mathbf{x}, t)$ the probability that given $\mathbf{X}(t) = \mathbf{x}$ no reaction occurs in the time interval $[t, t + \tau)$ then the laws of probability imply:

$$p(\tau, j \,|\, \mathbf{x}, t)d\tau = P_0(\tau \,|\, \mathbf{x}, t) \times a_j(\mathbf{x})d\tau \qquad (2.6)$$

The laws of probability also imply:

$$P_0(\tau + d\tau \,|\, \mathbf{x}, t) = P_0(\tau \,|\, \mathbf{x}, t) \times (1 - \sum_{j=1}^{M} a_j(\mathbf{x})d\tau) \qquad (2.7)$$

An algebraic rearrangement of this last equation and passage to the limit $d\tau \to 0$ results in a differential equation whose solution is:

$$P_0(\tau \mid \mathbf{x}, t) = \exp\left(-a_0(\mathbf{x})\tau\right) \quad \text{where} \quad a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x}) \tag{2.8}$$

Plugging 2.8 in 2.6 we get

$$p(\tau, j \mid \mathbf{x}, t) = a_j(\mathbf{x}) \exp\left(-a_0(\mathbf{x})\tau\right) \tag{2.9}$$

This last equation implies that the joint density function of $\tau$ and $j$ can be written as the product of the $\tau$-density function, $a_0(\mathbf{x}) \exp\left(-a_0(\mathbf{x})\tau\right)$, and the $j$-density, $a_j(\mathbf{x})/a_0(\mathbf{x})$. Now by using the inversion method of Monte Carlo theory [42] one can generate random samples from these two densities to arrive at the following version of the *stochastic simulation algorithm (SSA)*, see [40, 41, 42]:

1. Initialise the time $t = t_0$ and the system's state $\mathbf{X}(t_0) = \mathbf{x_0}$.

2. With the system in state $\mathbf{x}$ at time t, compute all $a_j(\mathbf{x})$ and their sum $a_0(\mathbf{x})$.

3. Draw two random numbers $r_1$ and $r_2$ from the uniform distribution in the unit-interval, and select $\tau$ and $j$ according to

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right) \tag{2.10}$$

$$j = \text{the smallest integer satisfying } \sum_{j'=1}^{j} a_{j'}(\mathbf{x}) > r_2 a_0(\mathbf{x}) \tag{2.11}$$

4. Effect the next reaction by replacing $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}_j$.

5. Record $(t, \mathbf{x})$ as desired. Go to step 2 or stop simulation.

Note that the time step $\tau$ in the SSA is *exact* and is not a finite approximation to some infinitesimal $dt$ as is the case in most numerical solvers for differential equations. The SSA does not solve the CME numerically as it does not produce the probability density function of $\mathbf{X}(t)$. Nonetheless, much information can be achieved about this density function by either histogramming or averaging the results of many realisations of the SSA.

The main drawback of the SSA is its high computational cost due to the simulation of every reaction event one at a time. This is particularly inefficient in systems with many molecules where $a_0(\mathbf{x})$ is very large and therefore the generated $\tau s$ are very small.

There are some improvements of the SSA that give up some of the exactness to get more simulation speed [43, 44].

So far it has been assumed that the system volume $V$ is well-mixed, that is, the spatial distribution of molecules equilibrates on a shorter time scale than the characteristic time scales for changes in the state variables. However, if the molecules do not have time to diffuse through the whole reaction volume between their reactions, heterogeneity becomes apparent and the *well-mixed* volume assumption is not adequate anymore. The condition for homogeneity by diffusion is that

$$T_i >> L^2/D_i \quad \text{for all species } i \tag{2.12}$$

where $T_i$ is the average time between two reactions involving species $i$, $D_i$ is its diffusion constant and $L$ is the linear size of the system [9, 38]. When this condition is not satisfied one way to model spatial heterogeneity is to divide the volume $V$ into $n$ artificial subvolumes such that the length of the subvolumes $l$ is chosen so that (2.12) holds when replacing $L$ with $l$. Therefore each subvolume can be considered *well-mixed*.

In this case to represent the state of the system at time $t$, $\mathbf{X(t)}$, one needs one variable $X_{i,k}(t)$ to represent the number of molecules of species $S_i$, $1 \leq i \leq N$ in the subvolume $V_k$, $1 \leq k \leq n$. The state of the system can be changed by chemical reactions within the subvolumes and diffusion events between the subvolumes. The chemical reactions have different propensities in the different subvolumes as they depend on the local number of molecules. Diffusion is modelled as a memory lacking random walk in discrete space. A diffusion rate constant is associated with each diffusion reaction, $d_i^{\lambda\mu} = d_i^{\mu\lambda} = D_i/l^2$, for two neighboring subvolumes and zero otherwise. $D_i$ represents the diffusion rate of the species $S_i$ and $l$ is the linear size of the subvolume. The propensity of the diffusion of a species $S_i$ from $\lambda$ to $\mu$, two neighboring subvolumes, is computed as $d_i^{\lambda\mu} X_{i,\lambda}(t)$.

Now we can derive the so call *reaction-diffusion master equation (RDME)* similarly to the derivation of the CME.

$$
\begin{aligned}
\frac{dP(\mathbf{x},t\,|\,\mathbf{x}_0,t_0)}{dt} \quad = & \\
& \sum_{k=1}^{n}\sum_{j=1}^{M}[a_j(\mathbf{x}-\mathbf{v}_j)P(\mathbf{x}-\mathbf{v}_j,t\,|\,\mathbf{x}_0,t_0)-a_j(\mathbf{x})P(\mathbf{x},t\,|\,\mathbf{x}_0,t_0)]+ \\
& \sum_{\mu=1}^{n}\sum_{\lambda\neq\mu}\sum_{i=1}^{N}\{d_i^{\lambda\mu}(x_{i,\lambda}+1)P([\ldots x_{i,\lambda}+1\ldots x_{i,\mu}-1\ldots]\,|\,\mathbf{x}_0,t_0)- \\
& \qquad -d_i^{\lambda\mu}(x_{i,\lambda})P(\mathbf{x},t\,|\,\mathbf{x}_0,t_0)\}
\end{aligned} \tag{2.13}
$$

The first term is similar to the CME and it represents the evolution due to the reactions $j = 1, \ldots, M$ in the subvolumes $k = 1, \ldots, n$. The second term contains the state transitions rates due to diffusion between neighbouring subvolumes.

As in the case of CME, the *reaction diffusion equation* is too complicated for analytical analysis; instead, one can sample the Markov process one event at a time using an appropriate Monte Carlo method. The *next subvolume method* [31] is a recent adaptation of the SSA [40] to the situation where spatial homogeneity is questionable. In this case the space is partitioned into subvolumes where homogeneity can be assumed. Starting with an initial distribution of molecules, in each volume $k$ the time for the next reaction $\tau_k$ is sampled from an exponential distributed with an average of $1/a_0^k$ where $a_0^k = \sum a_i^k$ with $a_i^k$ the propensities of the reaction and diffusion events $i$ in the volume $k$. Then the events to take place in the different subvolumes are sorted according to the waiting time $\tau_k$. On the one hand, when a reaction event takes place in a subvolume the new reaction or diffusion event to take place is recomputed in that subvolume. On the other hand, when a diffusion event takes places it produces a change in two subvolumes and therefore new events to take place in both subvolumes are recomputed.

## 2.7. Computational Modelling

The complexity of cellular systems makes the use of computers necessary to help understand the underlying mechanisms. Nevertheless, until recently the majority of computational models in biology were implemented in custom programs and published as statements of the underlying mathematical model. No computational formalism was explicitly used to model and simulate cellular systems. Nevertheless, to be useful a computational model must be presented within a well defined, consistent and formal framework.

Recently many computational frameworks has been used to model cellular systems like Petri nets, process algebra, cellular automata, state charts, agents, etc. Here we will only discuss briefly the first two.

### 2.7.1. Petri nets

Petri Nets are a mathematical and computational tool for modelling and analysis of discrete event systems typically with concurrent behaviour. They offer a formal way to represent the structure of the interactions in a discrete event system, simulate its behaviour, and prove certain properties of the system [99]. Petri nets have applications in many fields of system engineering and computer science. Here, we focus only on a specific class of Petri nets called *place-transition net* or *PT-net*, for short.

A PT-net is a directed graph formed by two kinds of nodes called places $P$ and transitions $T$ respectively. Directed edges, called arcs, connect places to transitions, and transitions to places, each arc $(x, y)$ having a weight $W(x, y)$. Thus, for each transition, one identifies a set of input places, the places which have at least one arc directed to that transition, and a set of output places, the places which the outgoing arcs of that transition are directed to. A non-negative integer number of tokens is assigned to each place; the numbers of tokens defines the state of the PT-net, which is also called the *marking* of the PT-net. PT-nets are usually represented by diagrams where places are drawn as circles, transitions are drawn as squares, and an arc $(x, y)$ is added between $x$ and $y$ if $W(x, y) \geq 1$. These arcs are then annotated with their weight if this is 2 or more.

In a PT-net, a transition is enabled when the number of tokens in each input place is greater than or equal to the weight of the arc connecting that place to the transition. An enabled transition can fire by consuming tokens from its input places and producing tokens in its output places; the number of tokens produced and consumed are determined by the weight of the arcs involved. The firing of a transition can be understood as the movement of tokens from some input places to some output places.

In order to handle PT-nets a matrix representation turns out to be convenient. A marking is represented using a vector $M$ of size $|P|$ whose elements correspond to the number of tokens present in each place of the PT-net. The *incidence matrix* of a PT-net is the $|T| \times |P|$ matrix $A$ such that, every element $a_{ij}$ of $A$, denotes the change in the number of tokens in place $j$ due to the firing of transition $i$. A *control vector* $u$ is a vector of size $|T|$ containing 1 in position $i$ to denote the firing of transition $i$, 0 otherwise.

Then if the current marking of a PT-net is $M_k$ and the firing of a transition $i$ is represented by the control vector $u$ the next marking $M_{k+1}$ can be computed as

| **Biochemistry** | **PT-net** |
|---|---|
| Molecule | Place |
| Molecular Population | Marking |
| Biochemical Transformation | Transition |
| Reactant | Input Place |
| Product | Output Place |

Table 2.1.: Modelling Principles in PT-nets.

$$M_{k+1} = M_k + A'u$$

Thus, if a particular marking $M_n$ is reached from an initial marking $M_0$ through a firing sequence $u_1, u_2, \ldots, u_n$ of enabled transitions, we obtain

$$M_n = M_0 + A^T \cdot \sum_{k=1}^{n} u_k$$

which represents the *reachable-marking equation.*

A system of interacting molecules can be modelled as a discrete event system with concurrent behaviour using PT-nets as follows. Each molecular species is represented by a different place and each biochemial transformation as a different transition. The number of tokens inside a place is used to specified the number of molecules [99]. This modelling approach is summarised in Table 2.7.1. Within this framework only qualitative analysis can be performed. In order to be able to develop quantitative analysis *Stochastic Petri Nets (SPN)* were introduced in [46]. In SPNs enabled transitions fire with an exponentially distributed time delay according to the SSA described in the previous section. Each transition is associated with a rate parameter used to compute the propensity of each transition according to the weight of its input places.

Thus, a biochemical system is represented as a discrete event system whose structural properties, like *P-invariants*, *T-invariants*, boundedness, liveness, etc, are useful for drawing conclusions about the behaviour and structure of the original biochemical system. Here we refer to the current literature [45, 99, 103, 104, 105] for details regarding the theory and applications of Petri nets.

For instance, P-invariants determine the set of molecules whose total net concentrations remains unchanged during the application of certain biochemical transformations; T-invariants instead indicate the presence of cyclic reactions which lead to a condition where some reactions are in a state of continuous operation. Also, the property of liveness is useful to determine the absence of metabolic blocks which may hinder the progress of the biochemical system.

In order to illustrate the modelling principles in Petri nets we present a Petri net model of the cellular system presented in Figure 2.4. A graphical representation of our Petri net model is depicted in Figure 2.5. Each one of the six molecular entities of the cellular system, namely, *geneI*, *geneR*, *R*, *S*, *C*, and *C.geneI* is represented by a place. Each transition corresponds with a molecular interaction. For instance, transition $T2$ describes the formation of the complex $C$, represented by the output place of this transition, with $S$ and $R$, represented by the two input places of this transition. Finally, the initial marking consisting of two tokens in our Petri net, one in the place corresponding with the *geneI* and the other one in the place corresponding with the *geneR* representing the initial number of molecules in the system.

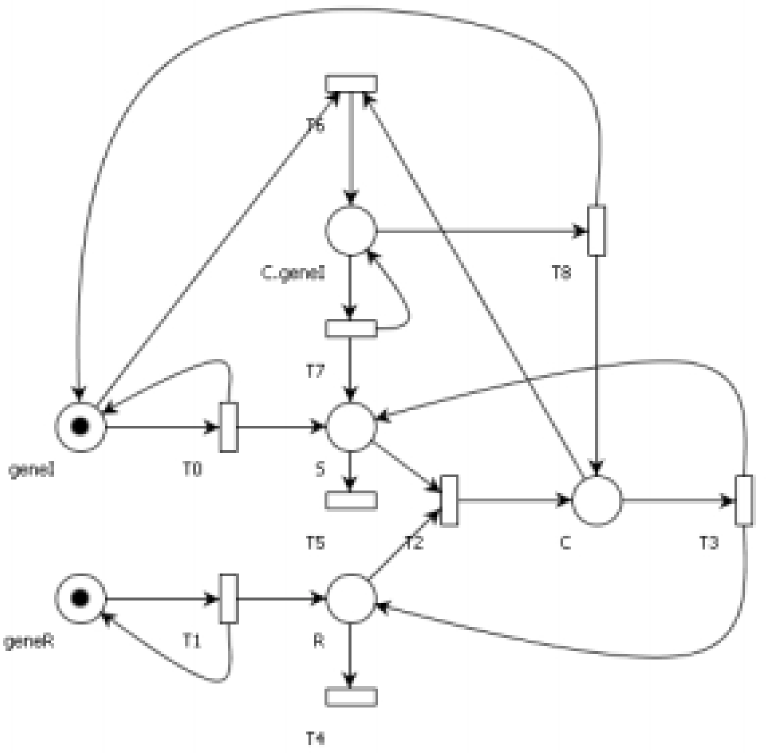

Figure 2.5.: A Petri net model of the cellular system introduced in Figure 2.4

## 2.7.2. $\pi$-calculus.

The $\pi$-calculus was introduced by Milner, Parrow, and Walker as a formal language to describe mobile concurrent processes that interact through communication channels [80]. It is now a widely accepted model for interacting systems with dynamically evolving communication topology. The $\pi$-calculus allows channels to be passed as data along other channels, and this fact provides channel mobility. This mobility is an important feature and increases the expressive power. The $\pi$-calculus has a simple semantics and a tractable algebraic theory. Starting with atomic actions and simpler processes, complex processes can be constructed in many ways. The process expressions are defined by guarded processes, parallel composition $P|Q$, nondeterministic choice $P+Q$, replication $!P$, and a restriction operator $(\nu x)P$ creating a local fresh channel $x$ for a process $P$. A structural congruence relation providing a static semantics is defined over the set of processes. The evolution of a process is described in $\pi$-calculus by a reduction relation over processes. This relation contains those transitions which can be inferred from a set of rules. Different variants have been used to model molecular interactions [102], gene networks, and to integrate molecular and gene networks [24].

In the $\pi-$calculus formalism a system of interacting molecular entities is modelled by a system of interacting processes that communicate through complementary communication channels. Each molecular species or domain is represented by a different process. The number of copies of each process is used to specify the number of molecules. Molecular interactions are described using complementary communication channels. Complexes and cellular compartments are represented by the scope of private communication channels. This modelling approach is summarised in Table 2.2.

Originally the semantics of the $\pi-$calculus were non-deterministic which is well suited for qualitative analysis. However for a more accurate quantitative modelling of biomolecular systems the *stochastic* $\pi-$*calculus* was introduced in [95]. To provide the $\pi-$calculus with a stochastic extension based on the SSA, described in the previous section, each communication channel is associated with a rate parameter. This constant is used to compute the propensity of that communication channels by taking into account the possible combinations of the processes trying to communicate through that specific channels.

| **Biochemistry** | **$\pi$-calculus** |
|---|---|
| Compartment | Private communication channel |
| Molecule | Process |
| Molecular Population | Systems of communicating processes |
| Biochemical Transformation | Communication channel |
| Compartment Translocation | Extrusion of a private channel's scope |

Table 2.2.: Modelling Principles in $\pi$-calculus

Finally, we present in Figure 2.6 a $\pi$-calculus model of the cellular system presented in Figure 2.4. Each molecular entity of the system is specified as a process. Molecular interactions are represented by communication channels. Therefore, every molecular interaction needs two processes to communicate through the communication channel representing it. In this respect, an auxiliary process *aux* has been introduced to model unimolecular reactions. For instance, the behaviour of the molecules $S$ is modelled using the process $S := cf?.C + sdeg?.0$. This process can communicate through the communication channel $cf$ [1] with the process $R := cf!.0 + rdeg?.0$ to represent a complex formation reaction or with the auxiliary process, *aux*, through the communication channel *sdeg* which corresponds with the unimolecular reactions consisting in the degradation of the molecules $S$. The initial number of molecules is specified in the initial processes, $geneI|geneR|aux$.

π-calculus model of gene expression

Initial processes: $geneI \mid geneR \mid aux$

Processes:

$geneI := prodI?.(geneI \mid S) + act?.CgeneI$

$geneR := prodR?.(geneR \mid R)$

$S := cf?.C + sdeg?.0$

$R := cf!.0 + rdeg?.0$

$C := cd?.(S \mid R) + act!.0$

$CgeneI := deact?.(geneI \mid C) + aprod?.(CgeneI \mid S)$

$aux := prodI!.aux + prodR!.aux + sdeg!.aux + rdeg!.aux + deact!.aux$

Figure 2.6.: A π-calculus model of the cellular system introduced in Figure 2.4

[1] Complementary communication channels are denoted using the symbols ? and !.

# Part II.

# Modelling Framework

*2.*

# 3. P Systems, a Modelling Framework

*The data are accumulating and the computers are humming, what we are lacking are the words, the grammar and the syntax of a new language.*

*Dennis Bray*

In chapter 1 we discussed different approaches to modelling cellular systems. In spite of the fact that each of these approaches captures some of the information regarding cellular systems and their components, none fully integrates dynamics and structural details of the different components. As mentioned in chapter 1 this should be done while keeping relevance, understandability, extensibility and computational/mathematical tractability. In this chapter P systems are presented as a modelling approach to cellular systems fulfilling all these requirements. In section 3.1 we briefly introduce P systems as a branch of Natural computing presenting their source of inspiration. The specific variant of P systems and the definitions used in this work are presented in section 3.2. An extension of the well known Gillespie algorithm, the *Multicompartmental Gillespie algorithm* will be introduced in section 3.3. Finally, a methodology to perform probabilistic model checking on P system models using PRISM is discussed in section 3.4.

## 3.1. P Systems, a Bio-inspired Model of Computation

Membrane Computing is an emergent branch of Natural Computing introduced by G. Păun in [89]. Since then it has received important attention from the scientific community. This new model of computation starts from the assumption that the processes taking place in the compartmental structure of a living cell can be interpreted as computations. Specifically, membrane computing starts from the observation that membranes play a fundamental role in the functioning of living cells. The cell membrane separates, and hence protects, the internal space of the cell from the external environment. The inner membranes define the structure of the cell, by identifying a number of internal compartments, and, in particular, they protect the nucleus which contains the genetic information. Membranes are essentially involved in many reactions taking place inside the various compartments of a cell, and they act as selective channels of communication between different compartments as well as between the cell and the environment. Membrane computing formalises and abstracts these features of living cells by introducing the notion of *P systems*, also called *membrane systems*.

Roughly speaking, the three essential components of a P system are:

1. A cell-like membrane structure containing a number of membranes arranged in an hierarchical way and delimiting regions or compartments. A compartment is the space between a membrane and the membranes (if any) directly included in it[1]. All membranes but one are included in a unique main membrane called *skin membrane* a membrane without any membrane inside is said to be *elementary*. The membrane structure can be represented formally, as a rooted tree, where the nodes are called membranes, the root is called skin, and the relationship of a membrane being inside another one is represented by the relationship of the node being the descendent of another one.

2. Multisets of objects and strings placed inside the compartments delimited by membranes.

3. Rewriting rules associated with specific compartments describing the evolution of the objects and strings placed inside the compartments.

Each compartment contains a multiset of objects and strings that evolve according to the rewriting rules assigned to the compartment. Specifically, when rules are applied they can consume/produce some objects, move objects from one compartment to another one, rewrite strings when specific objects are present, etc. Rules have a local scope since they are associated with compartments and when applied they only change the content of at most two compartments; the compartment they are assigned to and possibly a compartment located inside it or the compartment outside it.

In the original approach a *computation* in a membrane system is obtained by repeatedly applying the rules in a non-deterministic synchronous maximal parallel way: in each step, in each compartment, all the objects and strings that can evolve by means of any rule must evolve in parallel at the same time. Since the introduction of membrane computing in 1998, many variants of P systems have been proposed like P

[1] Note that there exists a one-to-one relationship between membranes and compartments and thus these terms will be used interchangeably through this work.

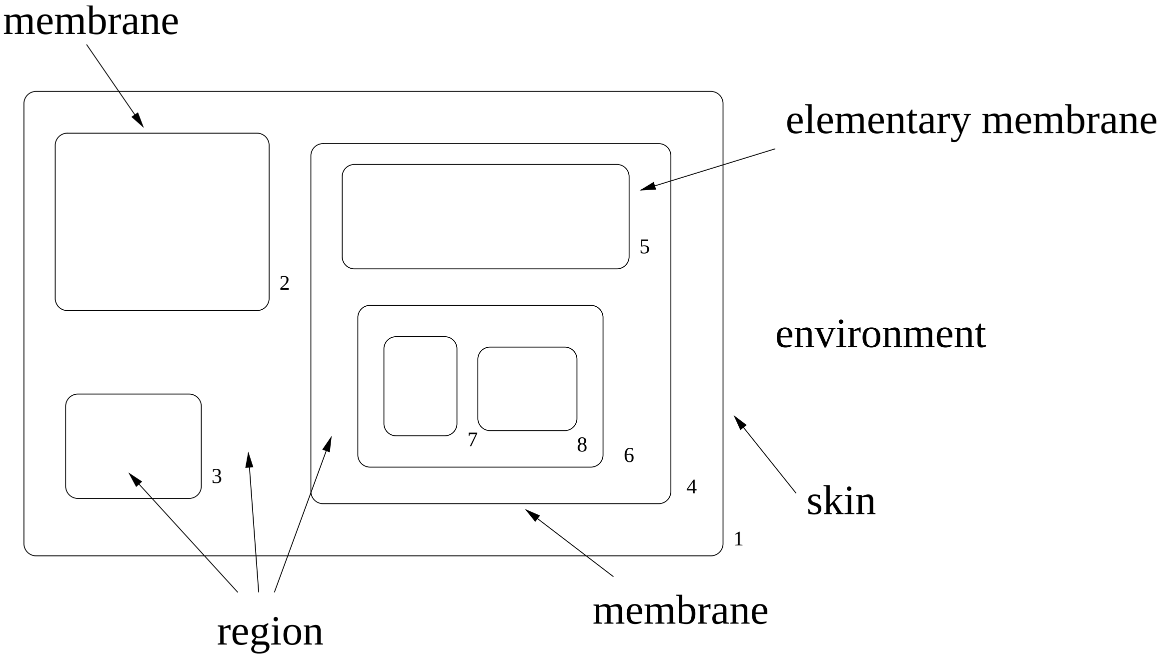

Figure 3.1.: A membrane structure containing eight membranes delimiting eight compartments.

systems with symport/antiport rules [90], with promoters/inhibitors [60], with active membranes, tissue P systems, Spiking Neural P systems [61], etc.

We must stress that P systems, according to the original motivation, were not intended to provide a comprehensive and accurate model of the living cell, rather, to explore the computational nature of various feature of biological membranes. Indeed, most variants of membrane systems have been proved to be computationally complete, that is equivalent in power to Turing machines, and computationally efficient, that is able to solve computationally hard problems in polynomial time by trading time with space. Although most research in P systems concentrates on computational powers, recently P systems have been used to model biological phenomena within the framework of computational systems biology presenting models of oscillatory systems [35], signal transduction [92], gene regulation control [107], quorum sensing [108] and metapopulations [93].

## 3.2. P system Specifications and P systems Models

In this section the main definitions used in this work are presented. First, we introduce *P system specifications* which will constitute the main structure used to analyse particular cell systems. A set of *parameters* is identified from the components of a P system specification. Then, the basic definition of P system specifications is extended to introduce *P system models*. Given a possible sets of values for the parameters of a P system specification, a P system model is obtained by instantiating the set of parameters using the given parameter values.

**Definition 3.1 (P system Specification).** A *P system specification* is a construct

$$\Pi = (O, L, \mu, M_1, M_2, \ldots, M_n, R_{l_1}, \ldots, R_{l_m})$$

where:

- $O$ is a finite alphabet of symbols representing objects;
- $L = \{l_1, \ldots, l_m\}$ is a finite alphabet of symbols representing labels for the compartments and identifying compartment types[2];
- $\mu$ is a membrane structure containing $n \geq 1$ membranes identified in a one to one manner with values in $\{1, \ldots, n\}$ and labelled with elements from $L$;
- $M_i = (l_i, w_i, s_i)$, for each $1 \leq i \leq n$, is the initial configuration of membrane $i$ with $l_i \in L$, the label of this membrane, $w_i \in O^*$ a finite multiset of objects and $s_i$ a finite set of strings over $O$;
- $R_{l_t} = \{r_1^{l_t}, \ldots, r_{k_{l_t}}^{l_t}\}$, for each $1 \leq t \leq m$, is a finite set of rewriting rules associated with compartments of the type represented by the label $l_t \in L$, of one of the following two forms:

  – Multiset rewriting rules:

$$r_j^{l_t} \,:\; obj_1\,[\,obj_2\,]_l \xrightarrow{c_j^{l_t}} obj_1'\,[\,obj_2'\,]_l \tag{3.1}$$

  with $obj_1, obj_2, obj_1', obj_2' \in O^*$ some finite multisets of objects and $l$ a label from $L$. A multiset of objects, $obj$ is represented as $obj = o_1 + o_2 + \cdots + o_m$ with $o_1, \ldots, o_m \in O$. The empty string, which represents the empty multiset, will be denoted by $\lambda$. For simplicity, we will write $o^n$ instead of $\overbrace{o + \cdots + o}^{n}$.

  These rules are multiset rewriting rules that operate on both sides of membranes, that is, a multiset $obj_1$ placed outside a membrane labelled by $l$ and a multiset $obj_2$ placed inside the same membrane can be simultaneously replaced with a multiset $obj_1'$ and a multiset $obj_2'$, respectively.

[2] Compartments with the same label will be considered of the same type and thus the same set of rules is associated with them.

– String rewriting rules:

$$r_j^{l_t} : [\, obj_1 + str_1; \ldots ; obj_p + str_p \,]_l \xrightarrow{c_j^{l_t}} [\, obj_1' + str_{1,1}' + \ldots str_{1,i_1}^{'}; \ldots ; obj_p' + str_{p,1}' + \ldots str_{p,i_p}' \,]_l \tag{3.2}$$

A string *str* is represented as follows $str = \langle s_1.s_2. \cdots .s_i \rangle$ where $s_1, \ldots, s_i \in O$. In this case each multiset of objects $obj_j$ and string $str_j$, $1 \leq j \leq p$, are replaced by a multiset of objects $obj_j'$ and strings $str_{j,1}' \ldots str_{j,i_j}'$.

Note that a constant $c_j^{l_t}$ is associated specifically with each rule. This constant will be referred to as *stochastic constant* and will be used to compute the propensity of the rule according to the current context of the compartment where the rule is located.

**Definition 3.2 (Parameters).** Given a P system specification $\Pi = (O, L, \mu, M_1, \ldots, M_n, R_{l_1}, \ldots, R_{l_m})$ the set of parameters of $\Pi$, $\mathcal{P}(\Pi) = (\mathcal{M}_0(\Pi), \mathcal{C}(\Pi))$, of consists of:

1. The initial multisets $\mathcal{M}_0(\Pi) = (M_1, \ldots, M_n)$ associated with the compartments.
2. The constants $\mathcal{C}(\Pi) = (r_j^{l_t})_{\substack{1 \leq j \leq k_{l_t} \\ 1 \leq t \leq m}}$, associated with the rewriting rules in $R_{l_1}, \ldots R_{l_m}$.

**Definition 3.3 (P system Model).** Let $\Pi$ be a P system specification with parameters $\mathcal{P}(\Pi) = (\mathcal{M}_0(\Pi), \mathcal{C}(\Pi))$, $\mathbb{M}_0$ and $\mathbb{C}$ a family of possible values for the initial mulitesets $\mathcal{M}_0(\Pi)$ and for the constants $\mathcal{C}(\Pi)$. A family of P system models, $\mathcal{F}(\Pi, \mathbb{M}_0, \mathbb{C})$, is obtained from $\Pi$, $\mathbb{M}_0$ and $\mathbb{C}$ by instantiating the parameters $\mathcal{P}(\Pi)$ using values from $\mathbb{M}_0$ and $\mathbb{C}$.

Hence given $\mathbb{M}_0$ and $\mathbb{C}$ sets of possible values for the parameters $\mathcal{P}(\Pi) = (\mathcal{M}_0(\Pi), \mathcal{C}(\Pi))$ of $\Pi$ and $(M_1, \ldots, M_n) \in \mathbb{M}_0$, $(r_j^{l_t})_{\substack{1 \leq j \leq k_{l_t} \\ 1 \leq t \leq m}} \in \mathbb{C}$ specific values, a P system model $(\Pi, (M_1, \ldots, M_n), (r_j^{l_t})_{\substack{1 \leq j \leq k_{l_t} \\ 1 \leq t \leq m}})$ is obtained by instantiating the parameters of $\Pi$ using the specific values from $(M_1, \ldots, M_n)$ and $(r_j^{l_t})_{\substack{1 \leq j \leq k_{l_t} \\ 1 \leq t \leq m}}$). In this way a family of P system models $\mathcal{F}(\Pi, (\mathbb{M}_0, \mathbb{C}))$ sharing the same P system specification $\Pi$ can be used to study the behaviour of a particular cell system specified by $\Pi$ under the different initial conditions collected in $\mathbb{M}_0$ and study the sensitivity of the system for the different rules constants in $\mathbb{C}$. Similar definitions were introduced in [93].

In order to illustrate the previous definitions we use the cell system described graphically in figure 3.2. Our example consists in a signal $S$ transported to the cytoplasm by a receptor $R$ located in the cell surface. Once in the cytoplasm $S$ interacts with a protein $Tf$ forming the complex $S$–$Tf$ which in turn activates by direct binding the genes *gene*–$Tf$ and *gene*–$R$, which codify the protein $Tf$ and the receptor $R$ respectively.

The following P system specification describes this system:

$$\Pi_0 = (O, L, \mu, M_1, M_2, M_3, R_{env}, R_{surf}, R_{cyto})$$

where:

- In the alphabet $O$ the molecular entities present in the system are represented:

  $O = \{S,\ R,\ S\text{–}R,\ Tf,\ S\text{–}Tf,\ gene\text{–}Tf,\ S\text{–}Tf\text{–}gene\text{–}Tf,\ gene\text{–}R,\ S\text{–}Tf\text{–}gene\text{–}R\}$

- The labels $L = \{env, surf, cyto\}$ specify the main regions of the system, namely, the environment, cell surface and cytoplasm.

- The membrane structure $\mu$ consists of three membrane defining the three compartments or regions involved in the system; the environment, the cell surface and the cytoplasm, identified respectively with the numbers 1, 2 and 3, and labelled with *env*, *surf* and *cyto*. A graphical representation of the membrane structure is given in figure 3.2.

- The initial multisets $M_1$, $M_2$ and $M_3$ represent the initial conditions of the compartments, environment, cell surface and cytoplasm, respectively. They are part of the parameters of the P system specification.

- The set of rewriting rules $R_{env}$, $R_{surf}$ and $R_{cyto}$ describe the molecular interactions taking place in the corresponding compartments.

  - The set of rules $R_{env} = \{r_1^{env}, r_2^{env}\}$ represents the binding of the signal $S$ to the receptor $R$, $r_1^{env}$ ; and the degradation of the signal $S$, $r_2^{env}$.

  $r_1^{env}$: $S\ [\ R\ ]_{env} \xrightarrow{c_1^{env}} [\ S\text{–}R\ ]_{env}$

  $r_2^{env}$: $[\ S\ ]_{env} \xrightarrow{c_2^{env}} [\ \ ]_{env}$

  - The set of rules $R_{surf} = \{r_1^{surf}, r_2^{surf}, r_3^{surf}\}$ specify the debinding of the signal $S$ from the receptor $R$, $r_1^{surf}$; the releasing of the signal $S$ to the cytoplasm, $r_2^{surf}$ and the internalisation or diffusion in of the receptor $R$ to the cytoplasm, $r_3^{surf}$.

  $r_1^{surf}$: $[\ S\text{–}R\ ]_{surf} \xrightarrow{c_1^{surf}} S\ [\ R\ ]_{surf}$

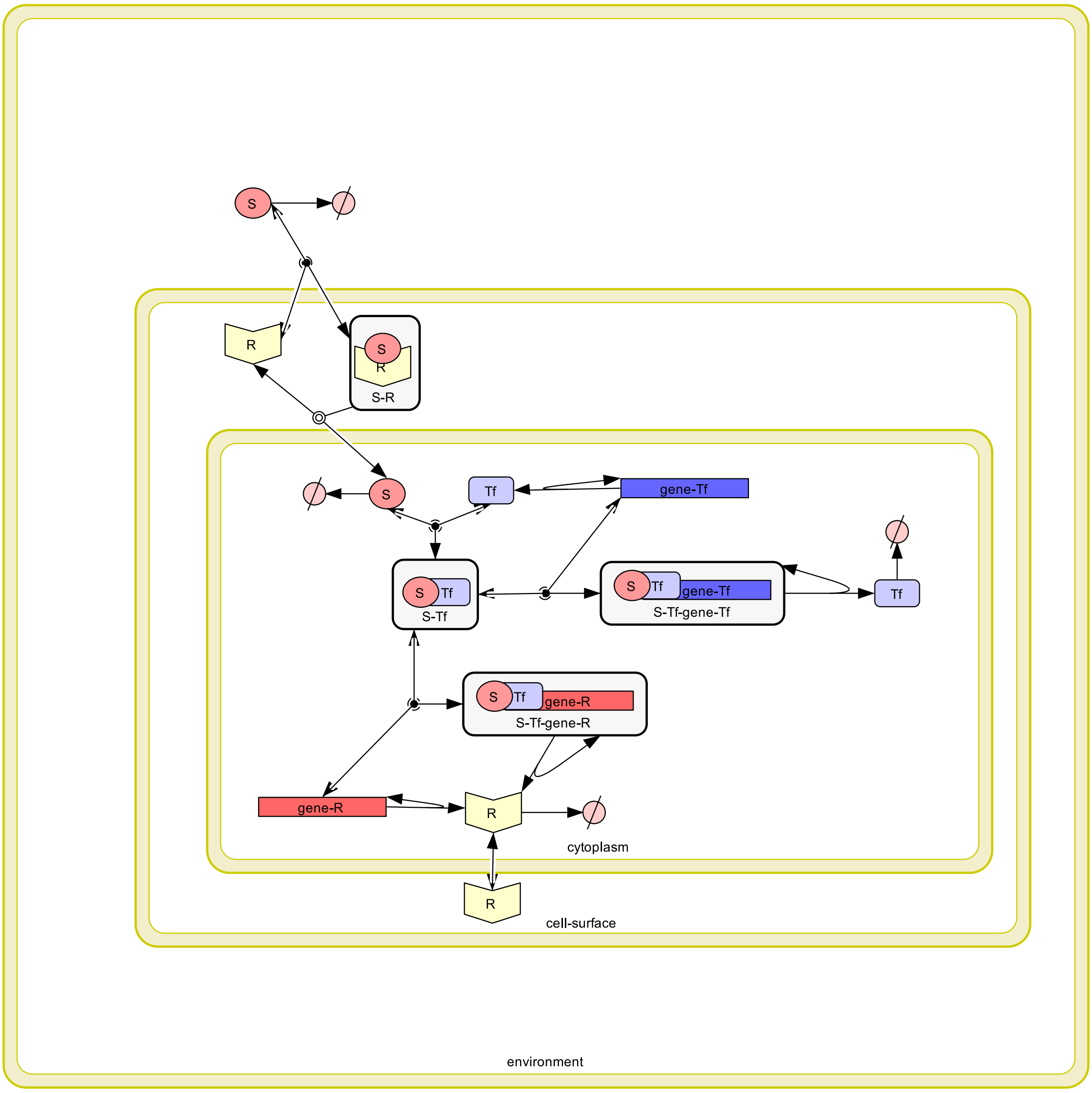

Figure 3.2.: An example of a cell system consisting of the activation of two genes, *gene–R* and *gene–$Tf$*, by a external signal $S$ which interacts with a protein $Tf$.

$r_2^{surf}$: $S\text{–}R\,[\;]_{cyto} \xrightarrow{c_2^{surf}} R\,[\,S\,]_{cyto}$

$r_3^{surf}$: $R\,[\;]_{cyto} \xrightarrow{c_2^{cyto}} [\,R\,]_{cyto}$

– The set of rules $R_{cyto}$ specify the molecular interactions taking place in the cytoplasm as follows:

$r_1^{cyto}$ : $[\,gene\text{–}Tf\,]_{cyto} \xrightarrow{c_1^{cyto}} [\,gene\text{–}Tf + Tf\,]_{cyto}$

$r_2^{cyto}$ : $[\,gene\text{–}R\,]_{cyto} \xrightarrow{c_2^{cyto}} [\,gene\text{–}R + R\,]_{cyto}$

These two rules represent the basal production of the protein $Tf$, $r_1^{cyto}$, and the receptor $R$, $r_2^{cyto}$.

$r_3^{cyto}$ : $[\,S + Tf\,]_{cyto} \xrightarrow{r_3^{cyto}} [\,S\text{–}Tf\,]_{cyto}$

$r_4^{cyto}$ : $[\,S\text{–}Tf\,]_l \xrightarrow{c_4^{cyto}} [\,S + Tf\,]_l$

The two previous rules describe the interactions between the signal $S$ and the protein $Tf$ leading to the formation and dissociation of the complex $S\text{–}Tf$.

$r_5^{cyto}$ : $[\,S\text{–}Tf + gene\text{–}Tf\,]_{cyto} \xrightarrow{c_5^{cyto}} [\,S\text{–}Tf\text{–}gene\text{–}Tf\,]_{cyto}$

$r_6^{cyto}$ : $[\,S\text{–}Tf\text{–}gene\text{–}Tf\,]_{cyto} \xrightarrow{c_6^{cyto}} [\,S\text{–}Tf + gene\text{–}Tf\,]_{cyto}$

$r_7^{cyto}$ : $[\,S\text{–}Tf\text{–}gene\text{–}Tf\,]_{cyto} \xrightarrow{c_7^{cyto}} [\,S\text{–}Tf\text{–}gene\text{–}Tf + Tf\,]_{cyto}$

$r_8^{cyto}$ : $[\,S\text{–}Tf + gene\text{–}R\,]_{cyto} \xrightarrow{c_8^{cyto}} [\,S\text{–}Tf\text{–}gene\text{–}R\,]_{cyto}$

$r_9^{cyto}$ : $[\,S\text{–}Tf\text{–}gene\text{–}R\,]_{cyto} \xrightarrow{c_9^{cyto}} [\,S\text{–}Tf + gene\text{–}R\,]_{cyto}$

$r_{10}^{cyto}$ : $[\,S\text{–}Tf\text{–}gene\text{–}R\,]_{cyto} \xrightarrow{c_{10}^{cyto}} [\,S\text{–}Tf\text{–}gene\text{–}R + R\,]_{cyto}$

The rules above specify the activation of the genes *gene–*$Tf$ and *gene–*$R$ by direct interaction with $S\text{–}Tf$; $r_5^{cyto}$, $r_6^{cyto}$, $r_8^{cyto}$ and $r_9^{cyto}$; and the active production of $Tf$ and $R$; $r_7^{cyto}$ and $r_{10}^{cyto}$.

$r_{11}^{cyto}$ : $[\,R\,]_{cyto} \xrightarrow{c_{11}^{cyto}} R\,[\;]_{cyto}$

$r_{12}^{cyto}$ : $[\,Tf\,]_{cyto} \xrightarrow{c_{12}^{cyto}} [\;]_{cyto}$

$r_{13}^{cyto}$ : $[\,R\,]_{cyto} \xrightarrow{c_{13}^{cyto}} [\;]_{cyto}$

$r_{14}^{cyto}$ : $[\,S\,]_{cyto} \xrightarrow{c_{14}^{cyto}} [\;]_{cyto}$

Rule $r_{11}^{cyto}$ represents the diffusion out from the cytoplasm or location of the receptor in the cell surface. Rules $r_{12}^{cyto}$, $r_{13}^{cyto}$ and $r_{14}^{cyto}$ specify the degradation in the cytoplasm of the protein $Tf$, receptor $R$ and signal $S$ respectively.

Once the corresponding P system specification, $\Pi_0$, is formulated one has to identify the parameters associated with it, $\mathcal{P}(\Pi_0)$. These parameters consists of:

- The initial multisets, $\mathcal{M}_0(\Pi_0) = (M_1, M_2, M_3)$, where $M_1 = (l_1, w_1, s_1)$ is associated with the environment, $M_2 = (l_2, w_2, s_2)$ is associated with the cell surface and $M_3 = (l_3, w_3, s_3)$ is associated with the cytoplasm.
- The rules constants $\mathcal{C}(\Pi_0) = (c_1^{env}, c_2^{env}, c_1^{surf}, c_2^{surf}, c_3^{surf}, c_1^{cyto}, \ldots, c_{14}^{cyto})$.

At this point we can define a family of P systems models by specifying finite ranges of values $\mathbb{M}_0$ and $\mathbb{C}$ for the parameters $\mathcal{M}_0(\Pi_0)$ and $\mathcal{C}(\Pi_0)$.

The possible values associated with the initial multisets, $\mathbb{M}_0 = (\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3)$, allow us the study of the behaviour of the system under different initial conditions. In our case we study the evolution of the system for different number of signals $S$ in the environment:

$$\begin{aligned}
&\mathbf{M}_1 = \{(env, \lambda, \lambda), (env, S^{50}, \lambda)\}\\
&\mathbf{M}_2 = \{(surf, \lambda, \lambda), (surf, R^3, \lambda)\}\\
&\mathbf{M}_3 = \{(cyto, gene\text{–}Tf + gene\text{–}R, \lambda), (cyto, R^3 + gene\text{–}Tf + gene\text{–}R, \lambda)\}
\end{aligned}$$

In most cases it is difficult or impossible to determine the specific values for the constants associated with the rules. Even in these cases, there exists some biological constraints which can be used to define ranges of possible values for the rule constants. We represent these possible values in $\mathbb{C}$. For example, as it will be discussed in chapter 3, the constant associated with the rules of the form $[\ a + b\ ]_l \xrightarrow{c_{on}} [\ c\ ]_l$ are diffusion limited by the value $\sim 0.16\ molec^{-1}sec^{-1}$ in a volume of $\sim 10^{-15}l$. Therefore we define $\mathbf{c}_1^{env} = \mathbf{c}_3^{cyto} = (0, 0.16)$. In a similar way using biological knowledge it is possible to determine ranges of values for the rest of rule constants in $\mathbb{C}$. The finite ranges of values defined in $\mathbb{C}$ allow us to search for estimates of the rule constants producing a evolution which best matches a given target behaviour, by using for example genetic algorithms. Another possibility is the study of the parameter sensitivity of the system.

Finally, we can investigate specific models from the family $\mathcal{F}(\Pi_0, (\mathbb{M}_0, \mathbb{C}))$.

## 3.3. Gillespie's Kinetics Theory for the Evolution of P Systems Models

In the original approach of membrane computing P systems evolve in a non deterministic and maximally parallel manner. All the objects in every membrane that can evolve according to any rule must evolve [89]. This produces a semi-quantitative framework that takes into account the discrete character of the molecular population and the role play by membranes in the structure and functioning of living cells. Although such coarse abstraction have been proved to achieve some success [6, 20], this approach fails to model quantitative aspects that are key to the functioning of many cell systems. Specifically the non deterministic and maximally parallel approach produces the following two inaccuracies:

- Reactions do not occur at a correct rate.
- All time steps are equal and do not represent the time evolution of the real cell system.

These two problems are interdependent and must be addressed when devising a relevant modelling framework for cell systems as it has been done in other computational approaches [46, 101].

In the field of membrane computing, the discrete aspect of the different components as well as the distributed and compartmentalised character of the structure, where the computation takes place, are fundamental. This is not the case with the non deterministic and maximal parallel semantics as have been studied in different variants [28, 36]. In this section the original approach will be replaced with various strategies that associate different rates to the rules depending on the current configuration of the system. Our strategies will be based on Gillespie's theory of stochastic kinetics [40, 41, 42, 43, 44].

To provide P systems with a stochastic extension a constant $c$ is associated to each rule. This constant depends only on the physical properties of the molecules involved in the reaction described by the rule and on other physical parameters of the system like temperature. It represents the probability per time unit that a particular reaction takes place and is used to compute the propensity of each rule which in turn determines the probability and time needed to apply the rule. This solves the two problems presented above.

There exists a different approach to modelling cell systems in membrane computing, based on the so called *Metabolic Algorithm* [23, 35], that keeps maximal parallelism as the strategy for the evolution of their models. Nonetheless they use rules of the form $a \rightarrow a$, called *transparent rules*, that have no effect on the state of the system, in order to bound the number of applied rules that actually change the system. Specific functions, called *reaction maps*, defined ad hoc, are also associated with rules to represent the reactions rates. By doing this the first of the two problems presented above is somehow solved; nevertheless the real time evolution of the system is not treated in this approach.

Finally, the *Metabolic Algorithm* is deterministic and so its applicability in certain cell systems suffers from the same drawbacks as other deterministic approaches like ODEs.

At the microscopic level of functioning of cellular processes the interactions between molecules follow the laws of physics. A fundamental result of theoretical statistical physics is the famous $\sqrt{n}$ law, which states that randomness or fluctuation level in a system are inversely proportional to the square root of the number of particles. As a result systems with a low number of molecules show high fluctuations and the application of the classical deterministic and continuous approach to modelling cell systems is questionable. In the following subsections different strategies for the evolution of P system models based on Gillespie algorithm are presented.

### 3.3.1. Multicompartmental Gillespie Algorithm

In section 2.6 an algorithm for simulating the time-course evolution of a stochastic kinetic model was introduced. This discrete-event simulation algorithm, usually referred to as the Gillespie algorithm or SSA (Stochastic Simulation Algorithm), has the nice properties that it simulates every reaction event and is exact in the sense that it generates exact independent realisations of the underlying stochastic kinetic model. Nevertheless, it should be emphasised that the Gillespie algorithm was developed for a single, well mixed and fixed volume or compartment. In what follows we present an adaptation of the Gillespie algorithm that can be applied in the different regions defined by the hierarchical and compartmentalised structure of a P system model. This will be referred to as *Multi-compartmental Gillespie algorithm.*

The starting point consists of treating each region, delimited by a membrane, as a well mixed and fixed volume where the classical Gillespie algorithm is applied. Thus in each compartment the next rule to be applied and the waiting time for this application is computed using a *local* Gillespie algorithm that only takes into account the number of molecules, rules and volume of the compartment. Given the state of a compartment $i$, $M_i = (l_i, w_i, s_i)$, the next rule to be applied and its waiting time is computed as follows:

1. Compute for each rule in $r_j \in R_{l_i}$ its propensity, $a_j(M_i)$, by multiplying the stochastic constant $c_j^{l_i}$ associated specifically with rule $r_j \in R_{l_i}$ by the number of distinct possible combinations of the objects and substrings present on the left-side of the rule with respect to the current contents of membranes involved in the rule.

2. Compute the sum of all propensities:

$$a_0(M_m) = \sum_{r_j \in R_{l_i}} a_j(M_i)$$

3. Draw two random numbers $r_1$ and $r_2$ from the uniform distribution in the unit-interval, and select $\tau_i$ and $j_i$ according to

$$\tau_i = \frac{1}{a_0(M_i)} \ln \left( \frac{1}{r_1} \right) \tag{3.3}$$

$$j_m = \text{the smallest integer satisfying} \sum_{j=1}^{j_i} a_j(\mathbf{x}) > r_2 a_0(M_i) \tag{3.4}$$

Then the compartments defined by membranes are ordered in a priority queue according to when the rules are scheduled to be applied. The first rule to be applied in the whole system occurs in the compartment on top of the priority queue. Depending on the type of rule that has been applied the state of a single compartment or of two compartments is changed. Therefore the waiting time and rule to be applied in these compartments must be recalculated. The algorithm stops when a prefixed simulation time is reached.

Next, the *Multi-compartmental Gillespie Algorithm* is described in detail:

- **Initialisation**
    - ○ set time of the simulation $t = 0$;
    - ○ for each membrane $i$ compute a triple $(\tau_i, j_i, i)$ by using the procedure described above; construct a list containing all such triples;
    - ○ sort this list of triples $(\tau_i, j_i, i)$ in increasing order according to $\tau_i$;
- **Iteration**
    - ○ extract the first triple, $(\tau_{i_0}, j_{i_0}, i_0)$ from the list;
    - ○ set time of the simulation $t = t + \tau_{i_0}$;
    - ○ update the waiting time for the rest of the triples in the list by subtracting $\tau_{i_0}$;
    - ○ apply the rule $r_{j_{i_0}}$ in membrane $i$ only once changing the number of objects and sites in the membranes affected by the application of the rule;
    - ○ for each membrane $i'$ affected by the application of the rule remove the corresponding triple $(\tau_{i'}, j_{i'}, i')$ from the list;

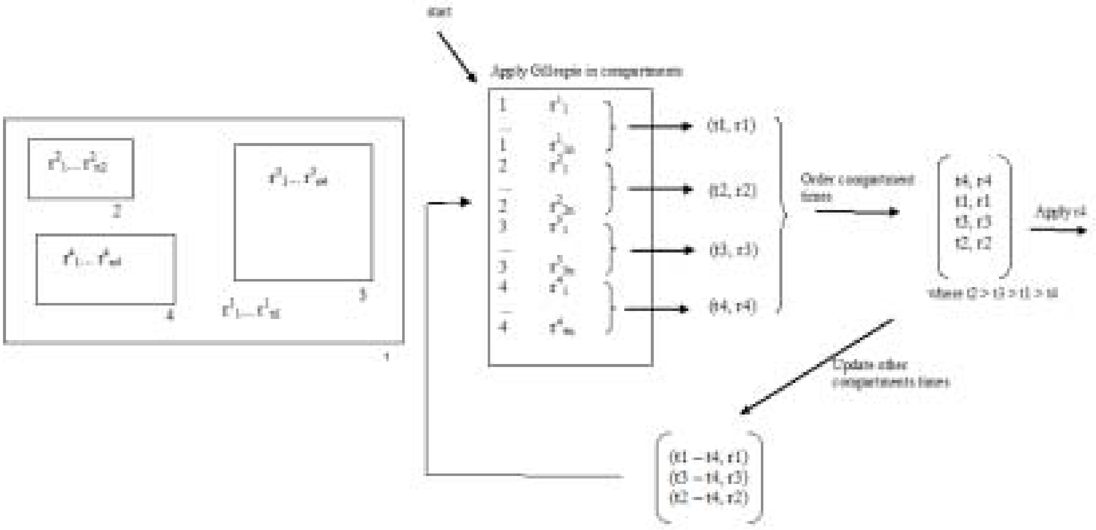

Figure 3.3.: Multicompartmental Gillespie algorithm

- ○ for each membrane $i'$ affected by the application of the rule $r_{j_{i_0}}$ re-run the Gillespie algorithm for the new context in $i'$ to obtain $(\tau'_{i'}, j'_{i'}, i')$, the next rule $r_{j'_{i'}}$, to be used inside membrane $i'$ and its waiting time $\tau'_{i'}$;
- ○ add the new triples $(\tau'_{i'}, j'_{i'}, i')$ in the list and sort this list according to each waiting time and iterate the process.

- **Termination**
  - ○ Terminate simulation when time of the simulation $t$ reaches or exceeds a preset maximal time of simulation.

It is worth noting that this is a local algorithm in the sense that all computations only consider the number of objects and rules of a single compartment. The only remaining global computation is the location of the index of the smallest waiting time, which could be improved by keeping all reaction times in an *indexed priority queue*. The advantage of having local computations is that the algorithm is easily implemented in an event-driven object-oriented programming style, such an implementation could be multithreaded on a hyper-threading machine and would also lend itself to full message-passing implementation on a parallel computing cluster.

In Figure 3.3 a graphical representation of the *Multicompartmental Gillespie Algorithm* is presented for a P system consisting of four compartments.

As an example of the use of this strategy we have taken two P systems models from the family $\mathcal{F}(\Pi_0, (\mathbb{M}_0, \mathbb{C}))$ defined in the previous section. These two P systems models $(\Pi_0; (\mathbf{M}_1(1), \mathbf{M}_2(1), \mathbf{M}_3(1)), C)$ and $(\Pi_0; (\mathbf{M}_1(2), \mathbf{M}_2(2), \mathbf{M}_3(2)), C)$ represent the different initial conditions, these being the absence and presence of signals in the environment respectively. Nevertheless, they share the same rule constants $C = (0.16,\ 10^{-3},\ 2,\ 4, 10^{-2},\ 10^{-2},\ 10^{-3},\ 0.1,\ 8,\ 5,\ 10,\ 0.2,\ 3,\ 10,\ 8 \times 10^{-2},\ 5 \times 10^{-3},\ 5 \times 10^{-3},\ 10^{-3},\ 3 \times 10^{-3})$ [3]. In figures, Figure 3.4 and Figure 3.5, we present two different realisations of the two models and an estimation of the probability of having different number of molecules at time 500 *sec* using 1000 different realisations of the models.

## 3.4. Probabilistic Model Checking on P Systems Models Using PRISM

Most research in systems biology focuses on the development of models of biological systems accurately enough such as to be able to reveal new properties that can be difficult or impossible to discover through direct lab experiments. One key question is what one can do with a model, other than simple simulation. Is it enough just to realise many simulations of a model, as has been done in Figure 3.4 and Figure 3.5, to really obtain novel knowledge on the system under study? This question has been considered in detail for deterministic models where a rich theory has been produced that can be used to analyse systems of differential equations such as stability and bifurcation analysis. However, this is not the case for stochastic models, as such systems defy conventional intuition and consequently are harder to conceive. The field is widely open for theoretical advances that help us to reason about systems in greater detail and with finer precision.

There are several attempts in this direction which consists of applying model checking tools to computational models of cell systems with the aim to analyse properties of such models automatically. The methods developed using model checking are not intended to replace the classical simulation and differential equation based approaches, instead they should be used in conjunction with them in order to gain greater insight into the complex interactions in cell systems.

In this respect, there are previous studies investigating the use of model checking for P system models [7, 29]. In this section we will propose the use of a probabilistic symbolic model checking approach based on PRISM (Probabilistic and Symbolic Model Checker) [133].

[3] The units of the constants associated with first order rules are $s^{-1}$ and the units of those constants associated with second order rules are $molec^{-1}s^{-1}$.

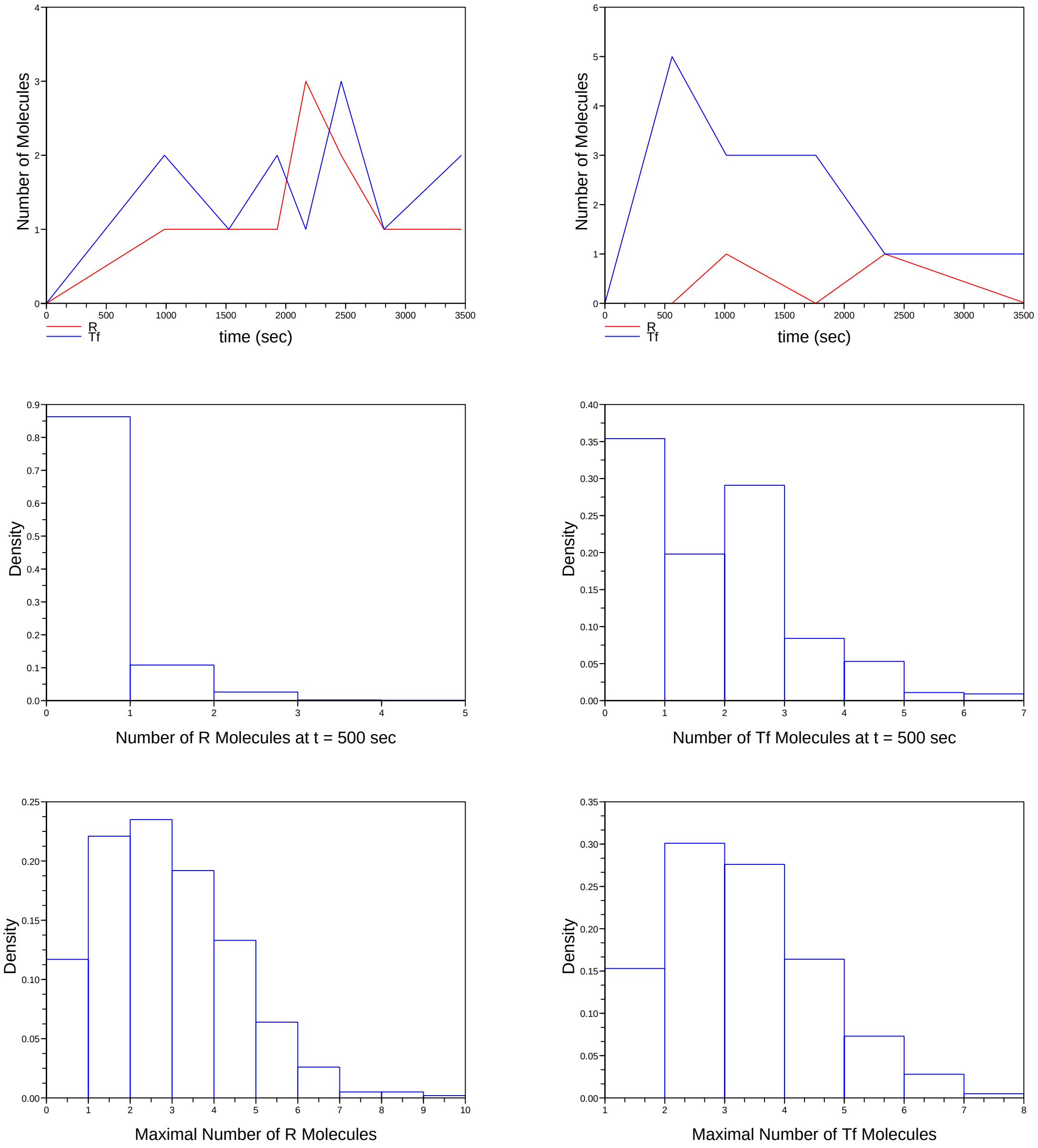

Figure 3.4.: The first two graphs constitute different evolutions of the number of receptor molecules $R$ and protein molecules $Tf$ in the P system model representing the absence of signals in the environment, $(\Pi_0; (\mathbf{M}_1(1), \mathbf{M}_2(1), \mathbf{M}_3(1)), C)$. The following two histograms present an estimation of the probability of having $n$ $R$ and $Tf$ molecules at time $t = 500$ *sec*. Finally, the last two histograms are an estimation of the probability of getting $n$ as the maximal number of $R$ and $Tf$ molecules within the first 500 seconds.

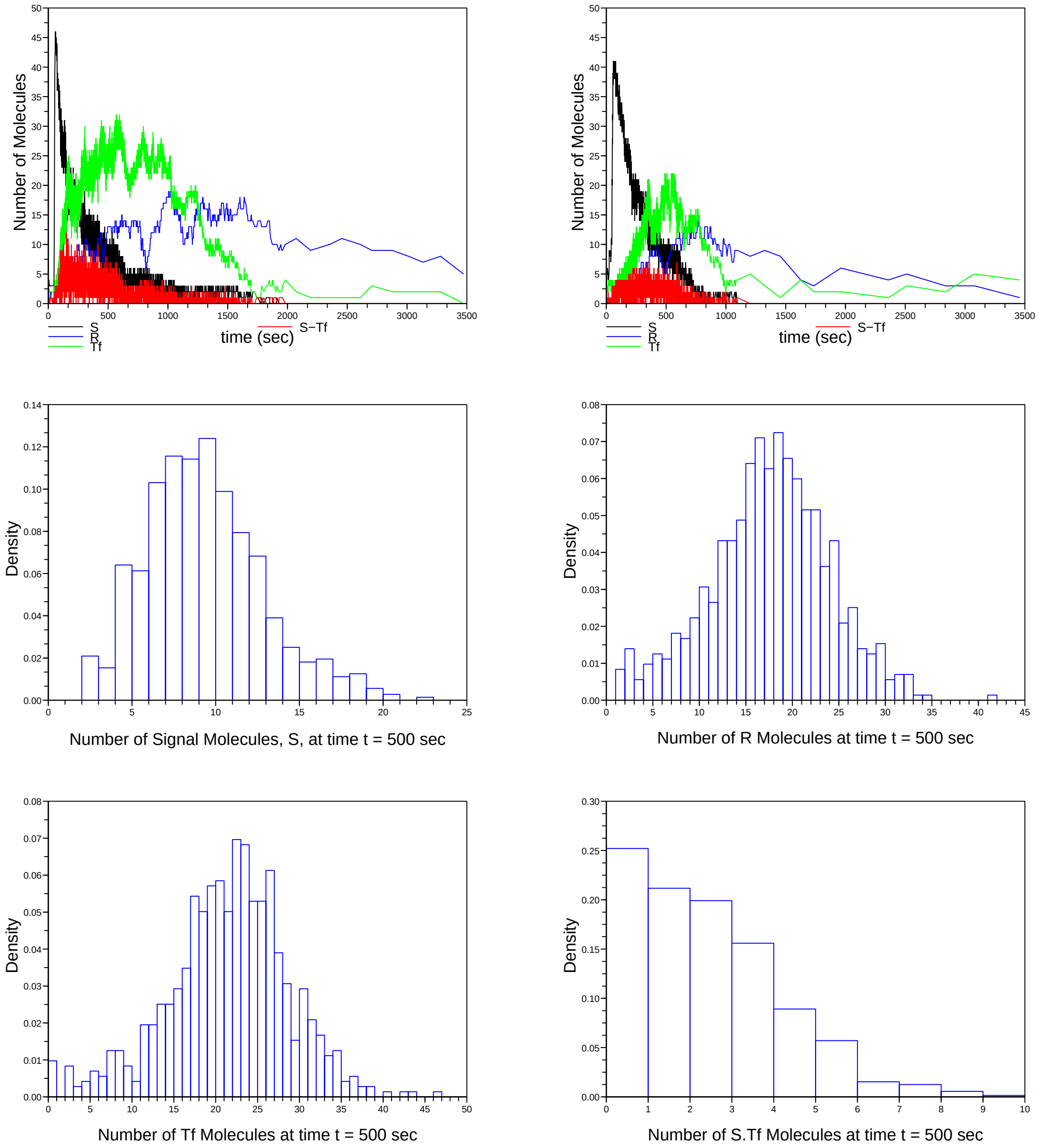

Figure 3.5.: In this figure we study the behaviour of the P system model, $(\Pi_0; (\mathbf{M}_1(2), \mathbf{M}_2(2), \mathbf{M}_3(2)), C)$, representing the presence of signals in the environment by analysing 1000 different simulations. As we saw in figure 3.4 three receptor and protein molecules are quite likely to be present in the system which explains the choice of the initial multisets $\mathbf{M}_2(2)$ and $\mathbf{M}_3(2)$.Observe the difference between the first two simulations of the system which suggest that the level of noise in the system is important. The following four histograms present an estimation of the probability of having $n$ molecules of the different molecular entities involved in the system at time 500 *sec*. These graphs also suggest a high level of noise.

### 3.4.1. Probabilistic Model Checking with PRISM

*Model checking* is a well established and widely used formal method for verifying the correctness of real life systems. *Probabilistic model checking* is a probabilistic variant of the classical model checking augmented with quantitative information regarding the likelihood that transitions occur and the times at which they do so [111]. Typically, probabilistic model checking works with *Markov chains* or *Markov decision process*. In this paper, it suffices to consider *continuous-time Markov chains* (CTMC). Formally, a CTMC is defined by a set of states $S$, a set of initial states $\overline{S} \subseteq S$ and a transition rate matrix $\mathbf{R} : S \times S \to \mathbb{R}$. This gives the rate $\mathbf{R}(s, s')$ at which transitions occur between each pair of states $s, s'$. The rate is taken as the parameter of an exponential distribution.

As a formal verification technique, probabilistic model checking has been successfully applied to the analysis of complex systems from a broad range of domains, including security and communication protocols, distributed algorithms and power management, [73].

Analytical methods based on probabilistic model checking consists of three different steps:

1. First, one must design a precise mathematical model of the system which is to be analysed. In this work, P systems specifications and models will be used as the formal description required in this step. Alternatively, other high-level formalisms such as Petri nets or process algebra could be used.

2. Once the formal model is built, one has to translate it into the specific language of the probabilistic model checking tool used to analyse the formal model, PRISM in this case. Typically in model checking and particularly in PRISM, this language will allow the construction of a labelled state-transition system in which each state represents a possible configuration and the transitions represent the evolution of the system from one configuration to another over time.

3. Finally, some properties of the model must be identified and expressed formally using temporal logic. This allows the probabilistic model checker to analyse these properties in an automatic way against the constructed model.

One of the major advantages of probabilistic model checking is that it is an exhaustive approach, that is, all possible behaviours of the system are analysed.

In this section we will use the software tool PRISM (Probabilistic and Simbolic Model Checker) to perform probabilistic model checking on P system models. In what follows a brief description of PRISM is presented, for details we refer to [133].

PRISM, the probabilistic and symbolic model checker in this study, supports three different types of probabilistic models, discrete time Markov chains (DTMC), Markov decision processes (MDP) and continuous time Markov chains (CTMC). PRISM supports systems specifications through two temporal logics, PCTL (probabilistic computation tree logic) for DTMC and MDP and CSL (continuous stochastic logic) for CTMC.

As mentioned before, in order to construct and analyse a model with PRISM, it must be specified in the PRISM language, a simple, high level, state-based language based on the Reactive Modules formalism of [5].

Here we describe some aspects of the PRISM language through the following illustrative example taken from [133].

```
// N-place queue + server

ctmc

const int N = 10;
const double mu = 1/10;
const double lambda = 1/2;
const double gamma = 1/3;

module queue
      q : [0 .. N] init 0;

      [] q < N -> mu : (q’ = q + 1);
      [] q = N -> mu : (q’ = q);
      [serve] q > 0 -> lambda : (q’ = q - 1);

endmodule

module server
      s : [0 .. 1] init 0;

      [serve] s = 0 -> 1 : (s’ = 1 );
      [] s = 1 -> gamma : (s’ = 0);

endmodule
```

The first step when writing a model in PRISM is to indicate which type of model is being described from those supported by PRISM. The keywords `dtmc`, `ctmc` or `mdp`, are typically placed at the very beginning of the file containing the model to announce that the model consist of a discrete time Markov chain, a continuous time Markov chain or a Markov decision process respectively. Our example constitutes a continuous Markov chain model as the key word `ctmc` attests.

The fundamental components of the PRISM language are *modules*, *variables* and *commands*. A model is composed of a number of modules which can interact with each other. A module contains a number of local variables and commands.

The previous example consists of two modules; the first one represents a `queue` and the second one represents a `server`.

A module is specified as:

```
module ⟨name⟩

endmodule
```

Note that, in the example above, there are only two local variables, `q` in the `queue` module representing the size of the queue, and `s` in the `server` module which represents whether or not the server is busy. In the declaration of a variable its initial value and range must be specified. A variable declaration looks like:

```
name : [ lower-bound .. upper-bound ] init value;
```

The values of these variables at any given time constitute the state of the module. The space of reachable states is computed using the range of each variable and its initial value. The global state of the whole model is determined by the local state of all modules.

The behaviour of each module is described by a set of commands. A command takes the form:

$$\texttt{[ action ] g} \rightarrow \lambda_1 : \mathbf{u_1} + \cdots + \lambda_n : \mathbf{u_n};$$

The guard `g` is a predicate over all the variables of the model. Each update $\mathbf{u_i}$ describes the new values of the variables in the module specifying a transition of the module. The expressions $\lambda_i$ are used to assign probabilistic information, rates, to transitions. For example, in the case of CTMC, $\lambda_i$ constitutes the parameter of a negative exponential distribution which determines the waiting time for the transition represented by $\mathbf{u_i}$ to take place.

The label `action` placed inside the square brackets are used to synchronise the application of different commands in different modules. This forces two or more modules to make transitions simultaneously. The rate of this transition is equal to the product of the individual rates, since the processes are assumed to be independent.

In our example, in the `queue` module there are three commands; the first one allows a new client to join the queue with probability `mu` if the maximal size, `N`, has not been reached yet; otherwise the second command maintains the size of the queue constant with probability `mu`. The third command is synchronised with the first command of the `server` module and describes the situation when there are clients in the queue and the server is free; in this case with rate `lambda` the server is set to busy and one client is removed from the queue. Note that the rate of this transition is equal to the product of the two individual rates (1 × `lambda` = `lambda`), this is a common technique, where one action is *passive* with rate 1 and the other action is *active*, defining the actual rate for the synchronised transition.

PRISM supports many other features like constants, expressions, process algebra operators, module renaming, etc. For a detailed description of the tool we refer to [133].

Once a probabilistic model has been specified and constructed in PRISM, one needs to identify one or more *properties* of the model which can be analysed using PRISM. This is done, as is common with formal verification techniques, using temporal logic. More specifically, PRISM supports properties expressed in a language based on the logics PCTL (probabilistic computation tree logic) for DTMCs and MDPs and CSL (continuous stochastic logic) for CTMC, probabilistic extensions of the classical temporal logic CTL. For details on the syntax and semantics of these two logics we refer to [50, 22, 10, 11]. This work will focus on CSL since we deal with Gillespie's kinetic theory which constitutes a Monte Carlo algorithm for the exact generation of trajectories of a CTMC represented by the chemical master equation, see section 2.6.

One key feature of probabilistic model checking and of PRISM in particular is the use of rewards associated with states and transitions. This allows to express reward-based properties in temporal logic which are quantitative in nature. Rewards associated with states, *cumulated rewards*, are incremented in proportion to the time spent in the state, while rewards associated with transitions *impulse rewards* are incremented each time the transition is taken. Rewards are defined using the construct **rewards ... endrewards**. State rewards are specified as *guard* : *reward*, which associates the reward *reward* with those states satisfying *guard*. For state rewards [*action*] *guard* : *reward* is used, the interpretation being that transitions from states which satisfy the guard *guard* and are labelled with the action *action* acquire the reward *reward*.

PRISM allows us to specify properties using three different operators:

- The **P** operator allows us to reason about the probability that a certain type of behaviour is observed. A behaviour is represented in PRISM by a set or path of states. Therefore, when specifying properties of a model one has to identify the set or path of states under study. This is achieved by writing a PRISM expression which evaluates to true for the corresponding states. PRISM supports the specification of path properties using the temporal operators *next* **X**, *until* **U** and *bounded until* **U** *time*. In the case of bounded until, *time* can take any of the three forms: $>= t$, $<= t$ or $[t_1, t_2]$ with $t, t_1$ and $t_2$ non negative doubles and $t_1 \leq t_2$.

  Given a path property, **pathprop**, the **P** operator can be used to check whether or not the probability of obtaining the behaviour described by **pathprop** satisfies the *bound*, where *bound* is of the form $>= p$, $> p$, $<= p$ or $< p$, with $p$ a non negative double in the range $[0, 1]$.

  $$\mathbf{P}\ \mathit{bound}\ [\ \mathbf{pathprop}\ ]$$

  PRISM can also compute the actual probability of a behaviour described by **pathprop** as follows:

  $$\mathbf{P}\ =?\ [\ \mathbf{pathprop}\ ]$$

- The **S** operator is used to reason about the *long run*, *equilibrium* or *steady state* behaviour of the model. In this respect this operator can only be used with the properties of a set of states, **prop**. This excludes path properties which use the temporal path operators.

  Similarly the probability of attaining a p can be checked or computed:

  $$\mathbf{S}\ \mathit{bound}\ [\ \mathbf{prop}\ ]$$

  $$\mathbf{S}\ =?\ [\ \mathbf{prop}\ ]$$

- The **R** operator analyses properties related to the expected values of rewards associated with the states of a PRISM model. This operator behaves in a similar way as the two previous ones, allowing us to check whether or not the expected reward satisfies a *bound* or to compute the expected reward itself using one of the following constructs:

**R** *bound* [ **rewardprop** ]

**R** =? [ **rewardprop** ]

where *bound* can be of the form $>= r$, $> r$, $<= r$ or $< r$, with $r$ a non negative double. There are four different types of reward properties, **rewardprop**, namely, *reachability reward*, **F** *prop*; *cumulative reward*, **C** $<= t$ ; *instantaneous reward*, **I** $= t$ and *steady state reward*, **S**.

### 3.4.2. Transforming P system Specifications into PRISM

Recall that the first step when analysing a formal model using probabilistic model checking is to translate it into the language of the particular tool that is being used. In what follows we will describe how P systems specifications, the high-level formal framework used to develop models of cell systems in this work, are specified in the PRISM language, the probabilistic model checker in this study. The example presented in Figure 3.2 is used to illustrate the method discussed in what follows.

First of all, we need to indicate which type of model is being described. In section 3.3 the strategy, according to which our P systems models evolve, was presented. This strategy is based on Gillespie's theory of stochastic kinetics which produces models that are equivalent to continuous time Markov chains. Therefore, a PRISM model specifying a P system model will start with the key word **cmtc**.

As discussed in section 3.1 the three essential components of a P system are a membrane structure consisting of a number of membranes that can interact with each other, multisets of objects and strings and rewriting rules associated with membranes. These components can easily be mapped into the components of the PRISM language using modules to represent membranes, variables to describe objects and commands to specify rules.

In what follows it is described in detail how to translate P system specifications into the PRISM language. Given a P system specification $\Pi = (O, L, \mu, M_1, M_2, \ldots, M_n, R_{l_1}, \ldots, R_{l_m})$ its components will be specified as follows:

- **Membrane structure:** Recall that each membrane in $\mu$ is uniquely identified with an identifier $i$, $1 \leq i \leq n$. Modules are used to describe the behaviour of membranes. In this respect for each membrane $i$ a module with name `compartment_i` will be introduced in the model.

For instance, the membrane structure of our example, Figure 3.2, consists of three membranes. Therefore the part of the PRISM model corresponding to the membrane structure will look like:

```
module compartment_1
          ⋮
endmodule

module compartment_2
          ⋮
endmodule

module compartment_3
          ⋮
endmodule
```

- **Alphabet and initial multisets:** For each object $obj \in O$ that can be present inside the compartment defined by membrane $i$ a local variable `obj_i` will be declared in module `compartment_i`.

  The initial value of this variable is determined by the initial multiset $M_i = (l_i, w_i, s_i)$ associated with membrane $i$. Since the initial multisets are part of the parameters of a P system specification we can define a constant, `ini_obj_i`, with no value assigned to it, which will make this variable a parameter of the PRISM model.

  The value range of the variables representing objects will be determined experimentally or it will be derived from the literature. In order to specify these ranges two constants will be declared `upb_obj_i` and `lob_obj_i`.

  In our example, the object $R$ appears in compartment 2, which identifies the cell surface. The variable `R_2` will be used to specify the number of molecules $R$. Its initial value corresponding to the number of molecules in the initial multiset $M_2$ will be described by the constant `ini_R_2`. The range of this variable will be defined using the constants `upb_R_2` and `lob_R_2`. This produces the following PRISM code fragment describing the object $R$. This code will be written in the module `compartment_2`.

```
const int ini_R_2;
const int upb_R_2;
const int lob_R_2;
R_2 : [ lob_R_2 .. upb_R_2 ] init ini_R_2 ;
```

- **Rewriting rules:** Commands are used in PRISM models to describe the rewriting rules of a P system specification. PRISM does not directly support the use of strings, because of this our work will be restricted to the type of rules in 3.1. These rules are typically referred to as protein-protein interactions in the literature.

  Given a compartment $i$, represented by module `compartment_i`, by checking the label associated with the compartment in the initial multiset $M_i = (l_i, w_i, s_i)$, one can determine the set of rules, $R_{l_i}$, that needs to be converted into the commands of the module.

  In general, protein-protein interaction rules need two membranes to interact in a synchronised way to exchange objects. Let's assume that membrane $i$ is contained in membrane $k$ described by module `compartment_k`. When a rule affects two different compartments, the two modules representing them will synchronise the application of two different commands, which describe the effect of the application of the rule in the corresponding compartments, by using the label `r_j_l_i`. In the case in which the application of a rule only involves one compartment the label `r_j_l_i` will still be used as it facilitates the formulation of rewards based properties associated with transitions.

  Therefore given a rule of the form:

  $$r_j^{l_i} \,:\, obj_1\,[\,obj_2\,]_l \xrightarrow{c_j^{l_i}} obj_1'\,[\,obj_2'\,]_l$$

  with $obj_1 = o_1^1 + \cdots + o_{n_1}^1$,$obj_2 = o_1^2 + \cdots + o_{n_2}^2$,$obj_1' = oo_1^1 + \cdots + oo_{m_1}^1$,$obj_2' = oo_1^2 + \cdots + oo_{m_2}^2$ some finite multisets and $c_j^{l_i}$ the stochastic constant associated with the rule. The variables `o_1_1_k`, ..., `o_n_1_1_k`, `oo_1_1_k`, ..., `oo_m_1_1_k` specify the objects from $obj_1$ and $obj_1'$ in module `compartment_k`. The objects `o_1_2_i`, ..., `o_n_2_2_i` and `oo_1_2_i`, `oo_m_2_2_i` represent the objects from $obj_2$ and $obj_2'$ in module `compartment_i`.

  The stochastic constants associated with the rules are part of the parameters of the P system specification and will be specified in the PRISM modules using constants with no value assigned to them, making them parameters of the PRISM model as well. More precisely, the constant $c_j^{l_i}$ associated with rule $r_j^{l_i}$ will be represented by the following constant declared at the beginning of the PRISM model before the specification of the modules.

```
const double c_j_l_i;
```

The command in module `compartment_i` describing the effect of an application of rule $r_j^{l_i}$ in compartment $i$ will be:

```
[ r_j_l_i ] o_1_2_i > 0 & ... & o_n_2_2_i > 0 &
            oo_1_2_i < upb_oo_1_2_i & ... &
            oo_m_2_2_i < upb_oo_m_2_2_i - >
                 c_j_l_i * o_1_2_i * ... * o_n_2_2_i :
                   (o_1_2_i’ = o_1_2_i - 1) & ... &
                   (o_n_2_2_i’ = o_n_2_2_i - 1) &
                   (oo_1_2_i’ = oo_1_2_i + 1) & ... &
                   (oo_m_2_2_i’ = oo_m_2_2_i + 1);
```

The command in module `compartment_k` describing the effect of an application of rule $r_j^{l_i}$ in compartment $k$ will be:

```
[ r_j_l_i ] o_1_1_k > 0 & ... & o_n_1_1_k > 0 &
            oo_1_1_k < upb_oo_1_1_k & ... &
            oo_m_1_1_k < upb_oo_m_1_1_k - >
                 o_1_1_k * ... * o_n_1_1_k :
                   (o_1_1_k’ = o_1_1_k - 1) & ... &
                   (o_n_1_1_k’ = o_n_1_1_k - 1) &
                   (oo_1_1_k’ = oo_1_1_k + 1) & ... &
                   (oo_m_1_1_k’ = oo_m_1_1_k + 1);
```

Observe that these two commands are applied when the guards hold, that is, if and only if there are some reactants in the corresponding membranes and the products have not reached the upper bounds determined experimentally. Also note that the rate of this transition is the product of the individual rates:

$$(\texttt{c\_j\_l\_i * o\_1\_2\_i * ... * o\_n\_2\_2\_i})\,(\texttt{o\_1\_1\_k * ... * o\_n\_1\_1\_k})$$

It has been assumed that all the objects on the left hand side of the rule are different. If there were objects with multiplicity greater than one present on the left hand side of the rule the rate associated with the command would be different and it will be computed as it is explained in section 2.6.

When this transition is performed the local variables representing the reactants are decreased by one and the variables representing the products are increased by one.

Finally, in order to illustrate this method of translating a P system specification into a PRISM model, we present how two of the rewriting rules from our example are converted into commands in the corresponding modules.

First, note that although the rules of the general form in (3.1) require synchronisation between two modules representing compartments, in many particular cases only one compartment is involved and no synchronisation is needed. For instance, the complex formation rule from our example:

$$r_3^{cyto} : \; [\, S + Tf \,]_{cyto} \xrightarrow{r_3^{cyto}} [\, S\text{–}Tf \,]_{cyto}$$

In this case there will be a single command representing this rule in module `compartment_3` which represents the cytoplasm in the PRISM. This command is presented in the following fragment of PRISM code:

```
const double c_3_cyto;

[r_3_cyto] S_3 > 0 & Tf_3 > 0 &

           S_Tf_3 < upb_S_Tf_3 ->

                c_3_cyto * S_3 * Tf_3 :

                   (S_3' = S_3 - 1) & (Tf_3' = Tf_3 - 1) &

                   (S_Tf_3' = S_Tf_3 + 1);
```

Although in many cases synchronisation is not used, the cases where it is necessary are not rare. For instance, in our example the binding of the signal molecule $S$ to the receptor $R$ implies the synchronisation of two modules, `compartment_1` and `compartment_2` representing the environment and cell surface respectively.

$$r_1^{env}\!: \; S\,[\, R \,]_{env} \xrightarrow{c_1^{env}} [\, S\text{–}R \,]_{env}$$

In this case it is necessary to split the description of this rule into two commands. The first command representing the effect of this rule in the compartment describing the environment, `compartment_1`, will be:

```
[ r_1_env ] S_1 > 0 ->

               c_1_env * S_1 :
                    S_1' = S_1 - 1;
```

The command describing the transition of module `compartment_2`, which specifies the cell surface, due to an application of the rule $r_1^{env}$ will be:

```
[ r_1_env ] R_2 > 0 &
             S_R_2 < upb_S_R_2 ->
                R_2 :
                  R_2' = R_2 - 1 &
                  S_R_2' = S_R_2 + 1;
```

Note that in the last example the general multiset of objects $obj_1'$ is empty and therefore there is no reference to it in the first command.

### 3.4.3. Analysis of P Systems Models Using PRISM

In this section we will illustrate how PRISM can be used to analyse P systems models by using our example in Figure 3.2. Here we will focus on the model $(\Pi_0; (\mathbf{M}_1(1), \mathbf{M}_2(1), \mathbf{M}_3(1)), C)$ presented in section 3.3 and which represents the initial condition with no signals.

The first step when analysing a model in PRISM is to associate the appropriate rewards with the corresponding states and transitions. We are interested in analysing the evolution over time of the number of molecules and the number of applications of rules. Therefore, two different list of rewards will be used:

- The first rewards list will associate with each state a reward representing the number of a particular molecule. A constant `molec` is used to identify which molecule is being tracked at the moment.

  ```
  const int molec;
  rewards "molecules"
  molec = 1 : s_1;
  molec = 2 : s_3;
  molec = 3 : r_2;
  ⋮
  molec = 11 : s_Tf_gene_r_3;
  endrewards
  ```

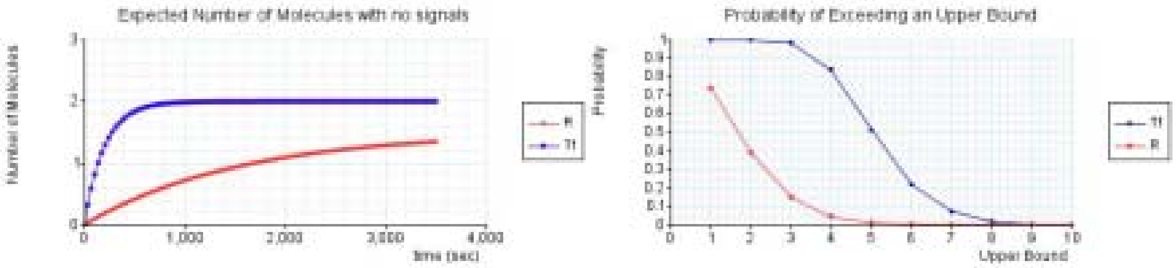

Figure 3.6.: Expected number of molecules and probability of exceeding a prefixed threshold

- In a similar manner a list of rewards will be used to associated with each transition a reward of 1 representing that the rule has been applied once. A constant `rule` is used to identify which rule is being analysed.

  ```
  const int rule;

  rewards "rules"

  [ r_1_env ] rule = 1 : 1;

  [ r_2_env ] rule = 2 : 1;

  [ r_1_surf ] rule = 3 : 1;
  ⋮
  [ r_14_cyto ] rule = 19 : 1;

  endrewards
  ```

Once the corresponding rewards have been associated with particular states and transitions one can use PRISM to model check some properties of the system.

For example, one common analysis, when dealing with stochastic models, is to compute the expected number of molecules over time. This can be studied in PRISM using instantaneous reward properties where a constant `time` indicates the time instant for which the expected number of molecules is computed.

```
R = ? [ I = time ]
```

Figure 3.6 left depicts the expected evolution of the number of $R$ and $Tf$ molecules over 3500 seconds. Note that both quantities of molecules seem to reach a steady state. The study of steady states is one of the most widely spread analytic methodologies. Steady-state reward properties of the form `R = ? [ S ]` allow us to compute the expected reward in the long run or steady state. This has been done for both, $R$ and $Tf$

molecules, which found that the expected number of $R$ and $Tf$ molecules in the long run is 2 and 1.5 molecules respectively.

Once the number of molecules in the steady state has been determined it is interesting to compute the expected time to reach the steady state. This can be done with the following reachability reward properties:

$$\texttt{R = ? [ F Tf\_3 = 2 ]} \Longrightarrow 250\ sec$$
$$\texttt{R = ? [ F R\_3 = 1 ]} \Longrightarrow 1000\ sec\ ^{4}$$

Although the expected number of molecules in the steady state for $Tf$ and $R$ is two and one respectively, the number of both types of molecules exceed these values in the simulations presented in 3.4. We can compute the probability of these molecules to exceed a preset threshold (`upb`) within the first 3500 seconds by asking PRISM to calculate the following probabilities:

```
P = ? [ true U <= 3500 Tf_3 > upb ]
P = ? [ true U <= 3500 R_2 > upb ]
```

The results of this computation are shown in Figure 3.6 right. Observe that the probability to exceed a threshold greater than the steady state is appreciable for both types of molecules. For instance, the probability of getting more than 5 $Tf$ molecules within 3500 *sec* is approximately 0.5. With regard to the expected time to reach the steady state, one can also check the probability of getting the expected number of molecules in the steady state before this expected time. These probabilities turn out to be considerable big as shown below:

$$\texttt{P = ? [ true U <= 200 Tf\_3 = 2 ]} \Longrightarrow 0.52$$
$$\texttt{P = ? [ true U <= 900 R\_2 = 1 ]} \Longrightarrow 0.59$$

As done before for the steady state values now we can compute the expected time when the number of $R$ and $Tf$ molecules exceeds an upper bound. This is shown in Figure 3.7.

PRISM also allows to reason about the evolution of P system models as a consequence of the applications of different rules. One can compute the probability and expected values of the number of applications of the rules. For instance, one can compute the expected number of applications of the different rules within $T$ units of time using cumulative reward properties of the following form:

[4] As we are working with a quantity of molecules represented as an integer number we assume the steady state of R is one molecule.

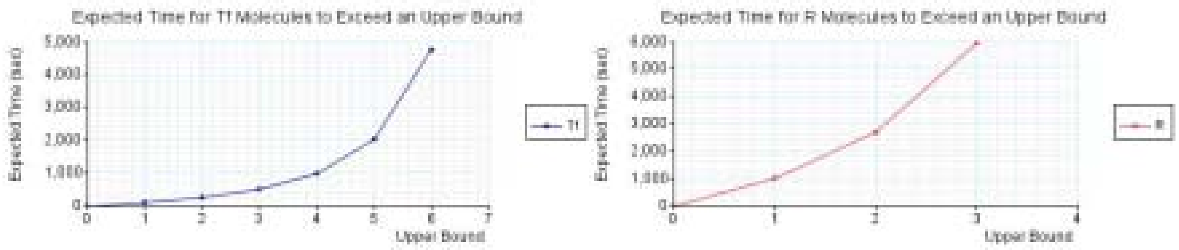

Figure 3.7.: Expected time for the number of $R$ and $Tf$ molecule to exceed a preset upper bound

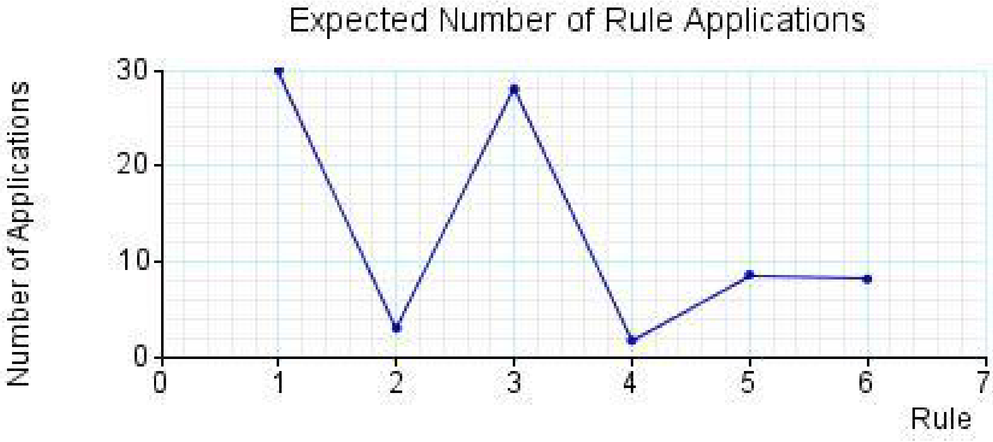

Figure 3.8.: Expected number of time each rule is applied during the evolution of our model

```
R = ? [ C <= T ]
```

In Figure 3.8 depicts the expected number of applications within 3500 *sec* of rules $r_1^{cyto}$, $r_{12}^{cyto}$, $r_2^{cyto}$, $r_{13}^{cyto}$, $r_{11}^{cyto}$ and $r_3^{surf}$.

The type of properties analysed in this section are only intended to illustrate how to use PRISM to study the behaviour of P system models. We do not intend to cover all possible properties, not even the most common ones, that can be checked in PRISM as the properties to study in a P system model depend very much on the model being analysed.

# 4. P System Specifications of Cellular Systems

*Life is an emergent, rather than an immanent and inherent, property of matter. Although it arises from the material world, it cannot be reduced to it.*

*Erwin Schrödinger*

*Life is a relationship among molecules and not a property of any molecule.*

*Linus Pauling*

In chapter 2 we discussed briefly different modelling approaches. All these modelling approaches were well established formalisms coming from different inspirations and sources before being applied to model cellular systems. For example, the $\pi$-calculus was introduced to specify mobile concurrent processes that interact through communication channels [80]. In contrast, P systems are inspired directly from the functioning and structure of the living cell. Therefore, the concepts in P systems are more similar to those used in molecular cell biology than the abstractions of other formalisms. In this chapter, P systems are presented as a formal framework for the specification and simulation of cellular systems which integrates structural and dynamic aspects in a comprehensive and relevant way while providing the required formalisation to perform mathematical and computational analysis.

Rather than being an alternative to more classical modelling frameworks, like ODEs, P systems constitute a complementary approach to be used when the classical modelling approaches fail to specify and simulate cellular systems correctly. Three situations where the applicability of conventional macroscopic, continuous and deterministic approaches, like ODEs, is questionable are:

- Cellular systems with a heterogeneous compartmentalised structure, low number of molecules and slow reactions.
- Processes in gene regulation like transcription and translation (inherently con-

current and discrete processes) and the binding of transcription factors to gene promoters (fundamentally a boolean behaviour).

- The order in which different genes are found in operons[1] on the genome is important as it determines the order in which genes are transcribed. These linear structures are not easily specified and simulated in classical approaches like ODEs.

In contrast to differential equations, P systems are an unconventional model of computation which explicitly represents the discrete character of the quantity of components of a cellular system by using rewriting rules on multisets of objects which represent molecules, and strings which describe the organisation of genes on the genome. The inherent stochasticity, external noise and uncertainty in cellular systems is captured by using stochastic strategies like the *Multicompartmental Gillespie's Algorithm* introduced in section 3.3.

The key differential feature of P systems is the so called *membrane structure* which represents the compartmentalisation in the structural organisation of cells. In this work we will show how by using membrane structures one can take into account the key role played by membranes and compartmentalisation in the functioning of the cellular systems. For instance, in chapter 6 selective uptake of molecules from the environment will be studied; in chapter 7 signalling at the cell surface will be specified and simulated; and finally in chapter 8 colonies of interacting bacteria which communicate by sending and receiving diffusing signals will be investigated. In all these case studies membranes in P system specifications will specify the relevant regions of the corresponding cellular system under study.

In what follows we present in detail how the main components of cellular systems are specified and simulated using P systems. We start by discussing how compartments are described using P system specifications. Section 4.2 presents an enumeration of the protein-protein interactions that can be specified using P system rewriting rules. The description of transcription networks is discussed in the following section. Here two different approaches are taken. Genes and mRNA can be represented using individual objects or strings. Moreover the processes involving proteins, genes and mRNA can be represented using either rewriting rules on multisets of objects, section 4.3.1, or using rewriting rules on multisets of objects and strings, section 4.3.2. Finally, section 4.4, introduces a method to translate into P systems cellular models specified using

[1] An operon is a group of genes physically linked on the chromosome and under the control of the same promoters. In an operon, the linked genes give rise to a single mRNA that is translated into the different gene products. This type of mRNA is called a polycistronic mRNA.

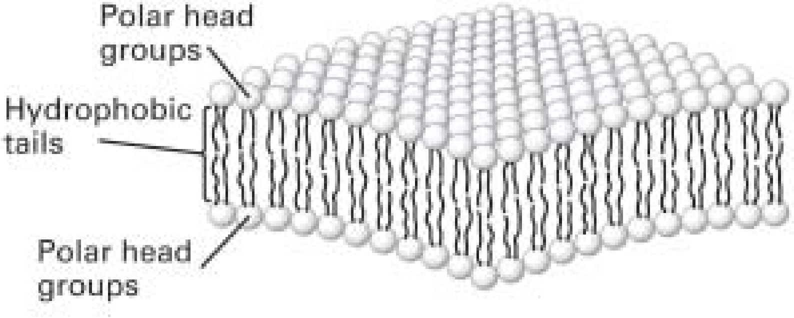

Figure 4.1.: Typical structure of biomembranes with a bilayer made of hydrophobic hydrocarbon chains and hydrophilic polar heads from [76]

SBML (Systems Biology Markup Language), a machine-readable language, derived from XML, for representing models of biochemical reaction networks.

## 4.1. P system Specifications of Compartments

Membranes play a key role in the functioning and structural organisation of both prokaryotic and eukaryotic cells. The basic structural unit of all biomembranes is a phopholipid bilayer, a two dimensional sheet with a hydrophobic core and hydrophilic faces, see Figure 4.1.

Prokaryotes, which are the smallest cells ($\sim 1 - 2\mu m$), present a simple structure consisting of a single compartment surrounded by a plasma membrane. In the larger eukaryotic cells, the rates of chemical reactions would be limited by the diffusion of small molecules if a cell were not structured into smaller subcompartments termed organelles. Each organelle is surrounded by one or more membranes enclosing a unique complement of proteins enabling it to carry out its characteristic cellular functions. For example, *endosomes* take up soluble macromolecules from the cell exterior, *lysosomes* are acidic organelles that contain a battery of degradative enzymes, the *smooth endoplasmic reticulum* synthesises fatty acids and phospholipids, the *rough endoplasmic reticulum* synthesises proteins, the *golgi complex* processes and sorts secreted and membrane proteins, *plant vacuoles* store small molecules, *mitocondria* are the principal sites of ATP production in aerobic cells, *chloropasts* contain internal compartments in which photosynthesis takes place, the *nucleus* contains the DNA genome and RNA synthesis aparatus, etc. The importance of internal membranes is evident from the fact that the total surface area of these membrane is roughly tenfold as great as that of the plasma membrane.

The function of membranes is not only limited to enclosing compartments where

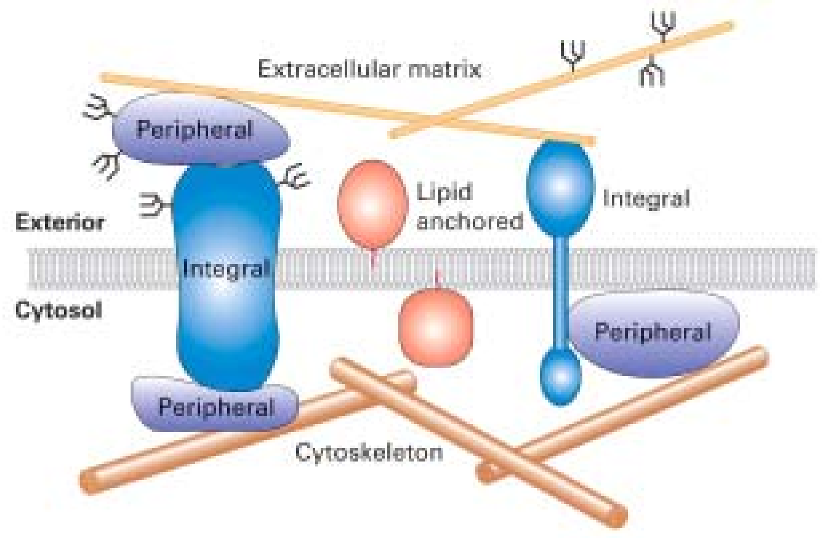

Figure 4.2.: Diagram showing how various classes of proteins are associated with the lipid bilayer making up a cell membrane from [76]

specific molecules and chemical reactions are located, they also control the movement of molecules between the inside and the outside of a cell and into and out of the organelles of eukaryotic cells. Another important mechanism where membranes play a key role is cell signalling. These processes are carried out by three different types of proteins that are associated with membranes, see Figure 4.2.

- *Integral or transmembrane proteins* which span the bilayer structure of membranes and are normally made of three different segments: one hydrophobic domain embedded in the membrane, one cytosolic domain and an exoplasmic one; these last two domains are hydrophilic.
- *Lipid anchored membrane proteins* which are tethered to one leaflet of the bilayer.
- *Peripheral membrane proteins* which are associated with membranes by specific noncovalent interactions with either integral or lipid anchored proteins.

The importance of these type of proteins is suggested from the finding that approximately a third of all yeast genes encode a membrane protein. The relative abundance of genes for membrane proteins is even greater in multicellular organisms in which membrane proteins have additional functions in cell adhesion.

Membrane proteins and lipids are not fixed. To the contrary, they can diffuse in the two-dimensional plane of a bilayer. Diffusion constants of $10^8 - 10^7$ $cm^2/s$ are characteristic of the proteins and lipids that constitute membranes. In this respect, the two dimensional region comprising a membrane can be considered as well mixed in certain situations.

Summing up, in cell compartments there are two distinct and relevant regions:

1. The *compartment surface* where a set of proteins, which control the movement of molecules and detect signals, are located.

2. The *lumen* or aqueous interior space where a characteristic complement of proteins carry out specific functions.

In the P system modelling framework, membranes are used to define relevant regions in cellular systems. In this work two different membranes will be used to specify the two relevant regions associated with a cellular compartment:

1. A first membrane will represent the compartment surface. In the region defined by this membrane the objects describing the integral, anchored and peripheral proteins associated with the compartment surface will be located. The processes involving molecular transport and cell signalling will be represented by rules which will also be associated with this region.

2. Another membrane will describe the aqueous interior of the compartment and thus it will be embedded inside the previous membrane. The set of objects and strings specifying the proteins and other molecules located in the lumen of the compartment will be placed in the region defined by this membrane. The rules describing the molecular interactions taking place inside the compartment are also associated with this membrane.

In some cases where no signalling or active transport of molecules is studied the membrane representing the compartment surface is omitted.

It is worth noting that up to now P systems have overlooked the importance of the cytoskeleton, a dense network of protein filaments that permeate the cytosol and mechanically support membranes. It is a key component of the structure of living cells serving as a scaffold to which particular sets of proteins and membranes are bound. It is also involved in a great variety of processes like molecular transport, cell division, cell mobility, etc. Its importance is suggested by the fact that while the surface of all membranes in an eukaryotic cell is around 8000 $\mu m^2$ the surface of the cytoskeleton is roughly 94000 $\mu m^2$.

## 4.2. P system Specifications of Protein-Protein Interactions

Large and complex networks of interacting proteins are responsible for most of the information processing within living cells. The present section aims to provide a comprehensive and relevant P system modelling schema for the most important protein-protein interactions that take place in living cells.

The theoretical and experimental description of protein-protein interactions is related to the field of chemical kinetics. A primary objective in this area is to determine the propensity or probability of a protein interaction, in order to describe the rate at which reactants are transformed into products. In this section the P system schema for each protein-protein interaction and its propensity, computed according to Gillespie's theory of stochastic kinetics [40, 41, 42, 43, 44], is presented.

A graphical representation will be provided for each P system schema using CellDesigner, a structured diagram editor for drawing gene-regulatory and biochemical networks [37].

- **Transformation and degradation:**

  A molecule $a$ can react to produce another molecule $b$ or it can be degraded by the cell machinery, see Figure 4.3. The dynamics of these reactions can be modelled using the *exponential decay law*. This law states that the rate of the reaction or its propensity is proportional to the number of molecules of the reactant $a$.

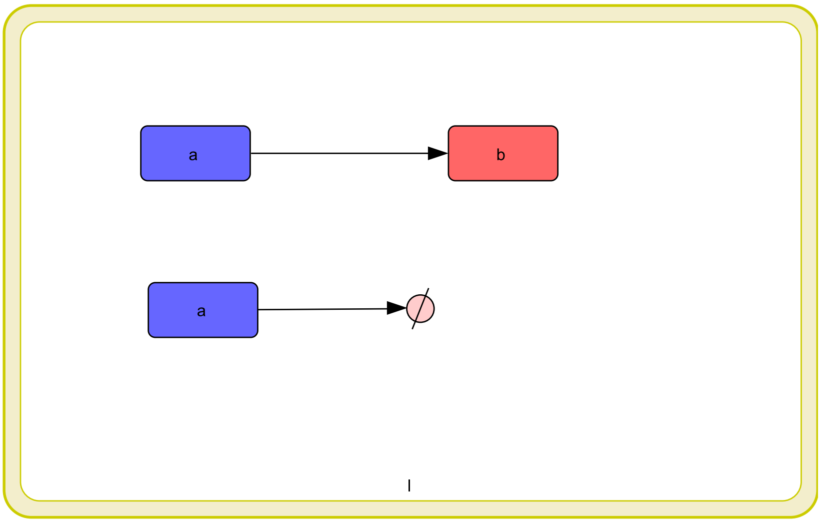

Figure 4.3.: Graphical representation of transformation and degradation.

In P system specifications, transformation and degradation are represented using the rewriting rules in the schema (4.1). In these rules the object $a$ is replaced with the object $b$ or is simply removed in the case of degradation. The compartment type where the molecules are transformed or degraded is also specified using square brackets with a label $l$. A constant $c$ is associated with the rule so that its propensity [2] can be computed.

$$\boxed{\begin{array}{ll} r_1 : \; [\, a \,]_l \xrightarrow{c} [\, b \,]_l & \\ & prop(r_i) = c \cdot |a| \quad i = 1,2 \\ r_2 : \; [\, a \,]_l \xrightarrow{c} [\ ]_l & \end{array}} \tag{4.1}$$

Note that these reactions are first order reactions or monomolecular reactions. As discussed in section 2.6, for this type of reaction the mesoscopic and stochastic constant, $c$, used in modelling approaches similar to ours is equal to the deterministic and macroscopic constant, $k$, used in ODEs. In this case this constant has units of time$^{-1}$.

- **Complex formation and dissociation:**

  Two molecules, $a$ and $b$, can collide and stick together through noncovalent interactions to produce a complex $c$. Once a complex has been formed it can dissociate back into its components, $d$ and $e$ which could have changed as a consequence of the interaction.

  In biochemistry, these reactions are referred to as complex formation, more specifically heterodimer formation when $a \neq b$ and homodimer formation when $a = b$; and complex dissociation, see Figure 4.2.

  Many important cellular processes depend on complex formation and dissociation, since the binding of a molecule to another one can alter (regulate) the activity of the complex which can be completely different from the activity of the single molecules.

  The dynamics of these reactions follow the *mass action law*, which states that the rate or propensity of the reaction is directly proportional to the product of the number of the reactant molecules. Thus, two constants $c_{cf}$ and $c_{cd}$ are associated with the complex formation and dissociation reactions respectively so their rates or propensities can be computed.

[2] In this work $|a|$ will be used to represent the number of molecules $a$

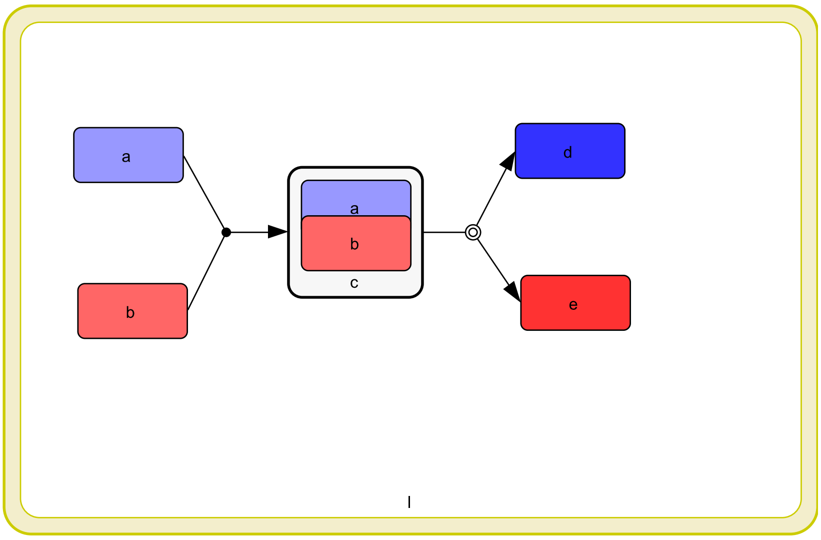

Figure 4.4.: Graphical representation of complex formation and dissociation.

In P system specifications, complex formation and dissociation reactions are specified using the rewriting rules in the schema (4.2) which take the name of the reactions they represent. In the complex formation rule, $r_{cf}$, the objects $a$ and $b$, representing the corresponding molecules, are replaced with the object $c$, representing the complex. In the same manner, in the complex dissociation rule, $r_{cd}$, the object $c$ is replaced with the objects $d$ and $e$. The compartment type in which the reactions take place is specified using square brackets and a label $l$.

$$\boxed{\begin{array}{llll} r_{cf}: & [\, a + b \,]_l \xrightarrow{c_{cf}} [\, c \,]_l & prop(r_{cf}) = \begin{cases} c_{cf} \cdot |a||b| & \text{if } a \neq b \\ c_{cf} \cdot \dfrac{|a|(|a|-1)}{2} & \text{if } a = b \end{cases} \\ r_{cd}: & [\, c \,]_l \xrightarrow{c_{cd}} [\, d + e \,]_l & prop(r_{cd}) = c_{cd} \cdot |c| \end{array}} \quad (4.2)$$

In this case, the stochastic constants $c_{cf}$ and $c_{cd}$, associated with the complex formation and complex dissociation rules, have units of time$^{-1}$ molecules$^{-1}$ and time$^{-1}$, respectively. As discussed in section 2.6, the constant $c_{cf}$ associated with the second order rule representing complex formation can be computed using the deterministic constant $k_{on}$, which have units of concentration$^{-1}$ time$^{-1}$, as follows:

$$c_{on} = \begin{cases} \dfrac{k_{on}}{V \times N_A} & \text{if } a \neq b \\ \\ \dfrac{2\ k_{on}}{V \times N_A} & \text{if } a = b \end{cases}$$

where $N_A$ represents Avogrado's number, the number of molecules in a mole, $N_A \sim 6.023 \times 10^{23}$ and $V$ represents the volume of the system. Here we have assumed that the units of concentration used are $M$ (moles/liter).

At this point it is worth noting that $c_{cf}$ represents the average number of collision events per time unit, and therefore it is limited by the rate of collisions of a diffusing molecule hitting a protein size target. Thus $c_{cf}$ has a diffusion-limited value of about $10^8 - 10^9\ M^{-1}sec^{-1}$[3]. This upper bound is independent of the details of the complex formation and therefore it can be used to determine a possible range of values for the constants associated with the complex formation rules when developing a P system model from a P system specification.

In contrast, the stochastic constant, $c_{cd}$, associated with the dissociation rule, $r_{cd}$, is equal to the deterministic constant $k_{off}$, since this rule represents a first order reaction. There is no general upper bound for $c_{cd}$, which can vary over many orders of magnitude for different reactions. This is due to the fact that $c_{cd}$ is determined by the strength of the chemical bounds between the molecules forming the complex.

Cellular systems are not only composed of networks of interacting proteins. Membranes and compartmentalisation play a key role in the organisation and the functioning of such systems. For example, in signalling at the cell surface, selective uptake of substances from the environment and diffusion across membranes in a colony of interacting bacteria.

Most high-level formalisms proposed so far to model cellular systems like ODEs, process algebra, Petri nets, agent-based approaches, etc. capture some of the information regarding pathways and their molecular components and interactions; nonetheless, none fully integrates quantitative dynamics, interactions among molecular entities and the structural organisation of cells in different compartments. Nevertheless, there have been recent attempts to specify compartmentalisation in process algebra like the $\pi$-calculus [101] and BioAmbients [100]. In this respect P systems present a clear advantage since membranes, compartments and interactions involving them can be explicitly

[3] In order to obtain the upper bound that can be compared directly to $c_{cf}$ it is necessary to express this value in molecules by multiplying it by $N_A$ and V

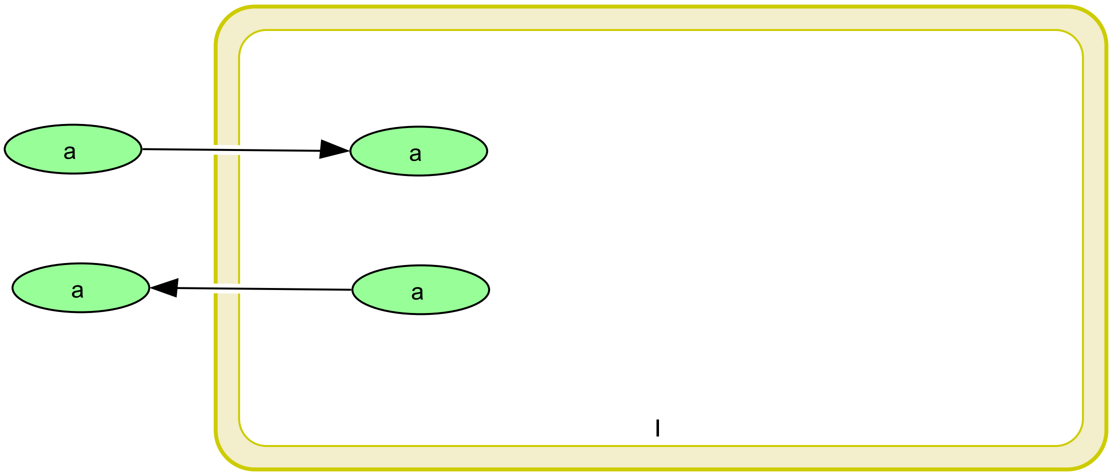

Figure 4.5.: Diffusion across a cell membrane

and directly specified in a comprehensive and relevant manner using the so called *membrane structure* and the P system specification schema presented in this chapter.

In what follows we will describe how the fundamental processes of communication and transport between different compartments in cellular systems, are specified within the P system modelling framework. Basically, we will deal with communication through passive diffusion across membranes, signalling at the cell surface and selective uptake of substances from the environment.

- **Diffusion in and out:**

  Small molecules can readily move by simple passive diffusion across membranes without the aid of transport proteins and without the consumption of any metabolic energy. The movement takes place down to the chemical concentration gradient of the diffusing molecule; that is, from regions with high concentrations to regions with low concentrations. The relative diffusion rate of any substance is proportional to its concentration gradient across the layer and to its hydrophobicity and size; charged molecules are also affected by electric potentials across the membrane, see Figure 4.5.

  The rewriting rules in (4.3) constitute a P system specification for diffusion in and out of a compartment. This compartment is represented by squared brackets with a label $l$, that identifies the type of the compartment. For diffusion in the object $a$ is moved from the compartment surrounding compartment $l$ inside the

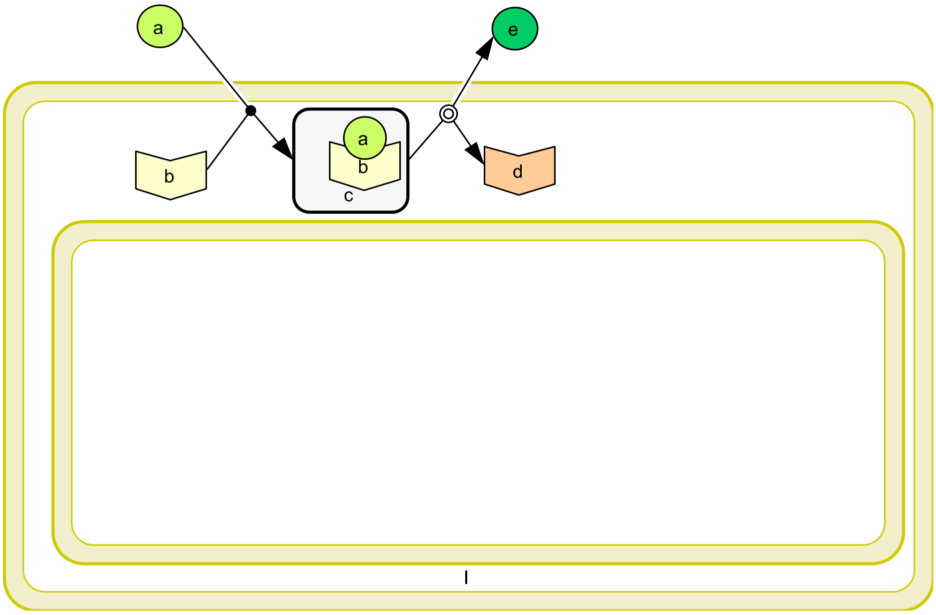

Figure 4.6.: Binding of a ligand to a transmembrane receptor

region defined by it. Viceversa for the case of diffusion out from compartment $l$. As in previous cases the constants $c_{in}$ and $c_{out}$ associated with the rules are used to compute their rates or propensities.

$$\boxed{\begin{array}{lll} r_1: & a\ [\ ]_l \xrightarrow{c_{in}} [\ a\ ]_l & prop(r_1) = c_{in}|a| \\ r_2: & [\ a\ ]_l \xrightarrow{c_{out}} a\ [\ ]_l & prop(r_2) = c_{out}|a| \end{array}} \tag{4.3}$$

- **Binding and debinding:**

  One of the key steps in the process of converting signals into cellular responses, *signal transduction*, is the binding of signalling molecules to structurally complementary sites on the extracellular or membrane-spanning domains of receptors leading to their activation, Figure 4.6.

  Note that, in signal transduction, the cell membrane plays a key role being the region where receptors are located and where they carry out their activities. In this respect in P system specifications, the plasma membrane is represented as the region defined by two membranes; one of them will represent the cell surface and the other one the cytoplasm; for case studies where this membrane structure is used see Chapters 6 and 7. In this region all the molecules associated with the cell membrane will be placed, as is the case with molecules that are attached either to the inside or the outside of the cell membrane, transmembrane receptors, receptor complexes, etc.

In P system specifications, the binding and debinding of a ligand to its receptor, located on the cell surface, is specified using the rewriting rules in (4.4). For the binding rule, the object $a$ representing the ligand is placed outside the compartment representing the cell surface, square brackets with label $l$. The receptor is specified using the object $b$ placed inside the square brackets. These two objects are replaced with the object $c$, the complex receptor-ligand, inside the square brackets which represent the cell membrane.

The debinding reaction, Figure 4.6, is specified by replacing the object $c$, inside the square brackets, with the object $d$, representing the ligand, outside the square brackets and the object $e$, representing the free receptor, inside them.

$$\boxed{\begin{array}{llll} r_1 : & a\;[\;b\;]_l \xrightarrow{c_{lb}} [\;c\;]_l & & prop(r_1) = c_{lb}|a||b| \\ r_2 : & [\;c\;]_l \xrightarrow{c_{ld}} d\;[\;e\;]_l & & prop(r_2) = c_{ld}|c| \end{array}} \tag{4.4}$$

Similar to the case of complex formation and dissociation the constants $c_{cf}$ and $c_{cd}$ are used to compute the propensity of the corresponding rules according to the mass action law. Also in this case $c_{lb}$ is diffusion limited and has units of $time^{-1}\,molec^{-1}$ whereas there is no general upper bound for $c_{ld}$ which has units of $time^{-1}$.

The P system schema representing binding and debinding reactions will be used to model signalling at the cell surface in chapter 7. Nevertheless, this schema is not limited to representing receptor activation. It can also be used to specify selective uptake (binding) of certain substances from the environment and delivering of substances to the environment (debinding) by specific transport proteins located on the cell surface. An example of this latter application of this schema will be presented in chapter 6.

- **Recruitment and releasing:**

  Binding of a ligand to its receptor produces a conformational change in the cytosolic domain or domains of the receptor that triggers the recruitment of some cytoplasmic proteins. These proteins are subsequently transformed and released back into the cytoplasm which ultimately induces specific cellular responses, see Figure 4.7.

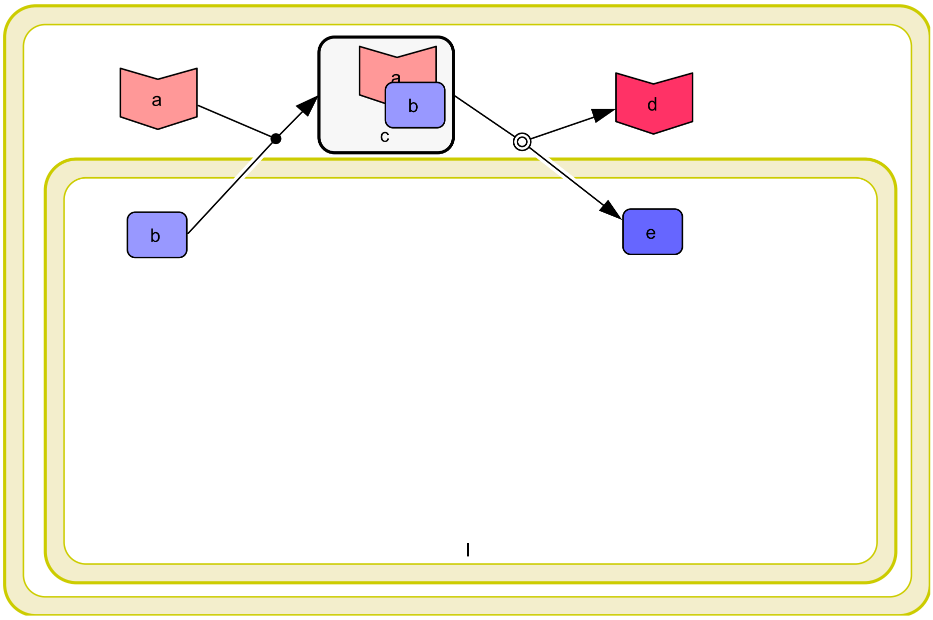

Figure 4.7.: Recruitment of a cytoplasmic protein by an active transmembrane receptor

The rules in (4.5) model recruitment and releasing reactions in P system specifications. The compartment from where or to where the proteins are recruited or released is specified using square brackets with a label $l$. In the recruitment rule, $r_{rt}$ the active receptor is represented by the object $a$ placed outside the compartment $l$ where the object $b$ represents the protein that is recruited. These objects are replaced with the object $c$ outside compartment $l$ specifying the formation of the complex formed by the active receptor and the recruited protein.

Conversely, in the releasing rule, $r_{rl}$, the object $c$ outside compartment $l$ is replaced with the objects $d$ outside and the object $e$ inside the compartment.

The constant $c_{rt}$ associated with the recruitment rule, $r_{rt}$, which can be considered to represent a complex formation reaction, is analogous to the constant $c_{cf}$ associated with the complex formation rule in the sense that it is used to compute the propensity of the corresponding rule using the mass action law. This constant is diffusion limited and has units of $time^{-1}$ $molec^{-1}$. The constant $c_{rl}$ is not bounded in general and has units of $time^{-1}$; in this sense it is similar to the complex dissociation rule $c_{cd}$.

$$\boxed{\begin{array}{llll} r_{rt}: & a\ [\ b\ ]_l \xrightarrow{c_{rt}} c\ [\ \ ]_l & & prop(r_{rt}) = c_{rt}|a||b| \\ r_{rl}: & c\ [\ \ ]_l \xrightarrow{c_{rl}} d\ [\ e\ ]_l & & prop(r_{rl}) = c_{rl}|c| \end{array}} \tag{4.5}$$

This P system specification schema will be used in chapter 7 to describe processes involving recruitment and releasing reactions in signalling at the cell surface. In chapter 6 this schema is used to specify processes consisting in the uptake (recruitment) of certain substances from the cytoplasm and the delivering of some substances to the cytoplasm (releasing) by specific transport proteins located on the cell surface.

## 4.3. P system Specifications of Transcription Networks

Living cells can sense very complex environmental signals through some of the molecular interactions described in the previous section; commonly through the binding of signals to receptors described in (4.4). They can also sense information about the internal state of the cell; commonly through complex formation described in (4.2). Cells respond to these signals by producing appropriate proteins codified in specific genes [4]. The rate of production of these proteins are regulated by special proteins called *transcription factors* which bind genes increasing or decreasing the rate at which they are transcribed into mRNA. In this sense, cells use transcription factors as an internal representation of the environmental and internal state of the cell. For instance, the *Escherichia Coli* has an internal representation of about 300 transcription factors (degrees of freedom) which regulate the rate of production of roughly 4000 proteins.

The interaction between transcription factors and genes leading to a change in the rate of production of certain proteins are described by *transcription networks*. In this section, P system specification schemas for transcription networks in prokraryotes are presented. For simplicity only prokaryotes will be considered. In contrast to prokaryotic cells where there is a single compartment, in eukaryotes there are two compartments involved in gene regulation, namely, the nucleus, where transcription and more complex processes like RNA splicing take place, and the cytoplasm where, for example, post transcriptional control and translation occur. In spite of the differences between gene

[4] The simplest definition of a gene is a stretch of DNA that contains information to specify synthesis of a single polypeptide chain (protein) or functional RNA

regulation control in prokaryotes and eukaryotes the same fundamental principles and mechanisms still apply in both cases.

The *central dogma* of molecular cell biology states that the necessary information for the production of proteins is contained in stretches of DNA called genes. Transcription of a gene is the process by which a protein called *RNA polymerase* produces the mRNA that corresponds to a gene's coding sequence. This mRNA is then translated into a protein or gene product, by *ribosomes*, complexes made of specific proteins and ribosomal RNA. This picture is much more complex than it first appears since *transcription factors*, which are also proteins encoded in certain genes, acts as regulators in the transcription rate of genes by binding to specific regions or sites of the DNA. These genes can codify in turn other transcription factors or other proteins produced to carry out specific tasks. Transcription factors can regulate genes in a positive or negative manner; an increase in the quantity of transcription factor leads to more or less gene expression. This provides a feedback pathway by which genes can regulate the expression of other genes and, in this manner, the production of the proteins encoded by them.

In this work two different approaches to the specification of transcription networks and gene regulation processes will be discussed.

In the first approach only objects will be used to specify the proteins, transcription factors and genes involved in the system. Rewriting rules on multisets of objects will describe the interactions between the different components of transcription networks in this approach.

In the second approach a much more detailed description of the interactions will be developed using objects to represent proteins and transcription factors and strings to represent genes, operons and mRNA. Rewriting rules on multisets of objects and strings will provide a more mechanistic description of the processes that take place in transcription networks.

### 4.3.1. Specification of Transcription Networks using Objects

In a simplistic approach processes like transcription and translation can be abstracted as individual reactions. In this case genes and operons will be specified as individual objects which produce in a single step their complementary mRNA also represented by a single object. The production of a protein from the mRNA is also described in a single step. Finally, the processes involved in gene expression control, like binding and debinding of transcription factors from genes, are also specified using rewriting rules on multisets of objects.

In what follows, we present P system specification schemas using objects and rewriting rules on multisets of objects to describe the processes that take place in transcription networks.

- **Transcription and translation:**

  As mentioned above the central dogma of molecular cell biology states that genetic information is stored in the DNA. This information is transcribed into mRNA which in turn is translated into proteins, see Figure 4.8.

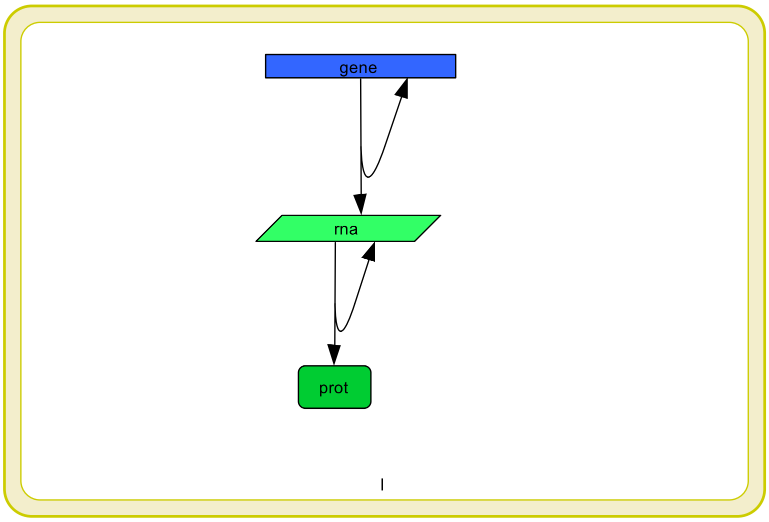

Figure 4.8.: Transcription and translation of a gene codifying the protein *prot*.

Here the linear information contained in genes and mRNAs is represented by individual objects. Specifically, in the P system specification schema in (4.6) the objects *gene*, *rna* and *prot* specify the stretch of DNA consisting of the gene, its complementary mRNA, and its gene product or protein, respectively.

The transcription of the gene into its complementary mRNA is described by the rewriting rule, $r_{tc}$. According to this rule in the compartments of the type represented by the label $l$, the object *gene* is replaced with the objects *gene* and *rna*. In this manner, when the rule is applied the object *gene* remains in the compartment and an object *rna* representing the mRNA is produced.

$$\boxed{\begin{array}{lll} r_{tc}: & [\, gene \,]_l \xrightarrow{c_{tc}} [\, gene + rna \,]_l & prop(r_{tc}) = c_{tc}|gene| \\ r_{tl}: & [\, rna \,]_l \xrightarrow{c_{tl}} [\, rna + prot \,]_l & prop(r_{tl}) = c_{tl}|rna| \end{array}} \tag{4.6}$$

In a similar way translation is described by the single rewriting rule $r_{tl}$, according to which the object *rna* is replaced with the objects *rna* and *prot*. The application of this rule does not consume the object *rna* but it produces an object *prot* representing the translated protein.

The propensities of the rewriting rules in the P system specification schema in 4.6 are computed in the same way as the other first order rules presented previously. The units of the contants $c_{tc}$ and $c_{tl}$ are time$^{-1}$ and they represent the average number of mRNA transcripts produced per time unit and the average number of protein products translated from a single mRNA per time unit, respectively.

In a more simplistic specification of the production of proteins one can ignore the intermediary step consisting of the production of the mRNA and assume that the protein is produced directly from the gene in a single process described in (4.7). Although this is an extremely simplistic representation of the complex processes involved in the production of proteins it will be used in this work when other processes are the focus of research.

$$\boxed{r_{prod} : \quad [\ gene\ ]_l \xrightarrow{c_{prod}} [\ gene + prot\ ]_l \qquad prop(r_{prod}) = c_{prod}|gene|} \tag{4.7}$$

- **Binding and debinding of transcription factors to genes:**

  The rewriting rules in (4.6) on their own only allow us to specify genes whose expression does not depend on transcription factors and consequently on any external or internal signal. These genes are called *constitutive genes* and they are continuously transcribed at approximately the same rate since the cell needs the proteins encoded in them to perform basic tasks. For this reason they are also termed *housekeeping genes.*

  Unlike *constitutive genes*, *putative genes* are only expressed or transcribed when needed according to some signals sensed by the cells using transcription factors that are only active when these signals are present either in the environment or inside the cell. There are, basically, two different types of transcription factors, *activators* and *repressors.* Although both types bind to genes, see Figure 4.9, they have opposite effects. Activators increase the rate of transcription of genes whereas repressors produce a decrease in the rate of transcription of the genes to which they bind.

  The binding of a transcription factor to a gene is reversible. The processes of binding and debinding of a transcription factor to a gene can be described by

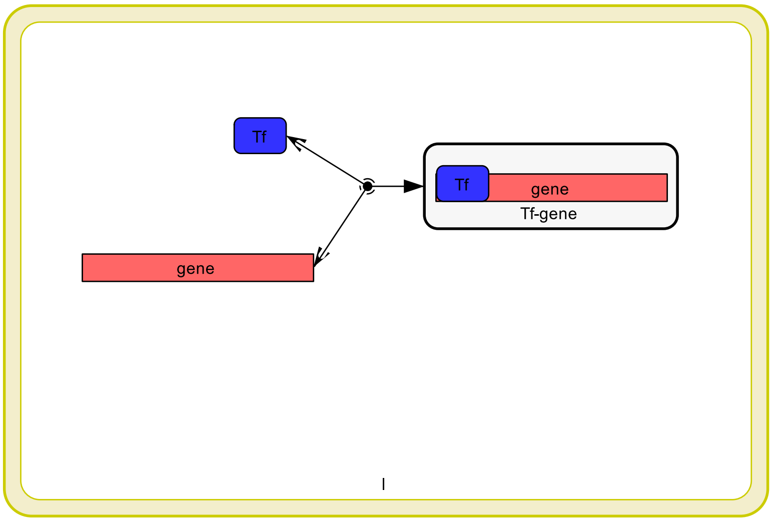

Figure 4.9.: Binding and debinding of a transcription factor Tf to a gene.

similar rules to the ones used to specify complex formation and dissociation. The P system specification schema in (4.8) constitute the specification of these processes through rewriting rules on multisets of objects.

Rule $r_{gon}$ represents the binding of a transcription factor, represented by the object $Tf$, to a gene, specified with the object $gene$. According to this rule in compartments of the type represented by the label $l$ a object $Tf$ and an object $gene$ can be replaced with an object $Tf\text{–}gene$, which represents the situation when the transcription factor, $Tf$, is bound to the gene, $gene$. The reverse process, the debinding of a transcription factor from a gene, is described through the rule $r_{goff}$. When this rule is applied in compartments of the type $l$ the object $Tf\text{–}gene$ is replaced with the objects $gene$ and $Tf$.

$$\boxed{\begin{array}{lll} r_{gon}: & [\, Tf + gene \,]_l \xrightarrow{c_{gon}} [\, Tf\text{–}gene \,]_l & prop(r_{gon}) = c_{gon}|Tf||gene| \\ r_{goff}: & [\, Tf\text{–}gene \,]_l \xrightarrow{c_{goff}} [\, Tf + gene \,]_l & prop(r_{goff}) = c_{goff}|Tf\text{–}gene| \end{array}} \tag{4.8}$$

The constant $c_{gon}$ associated with the rules describing the binding of transcriptions factors to genes is similar to the constant $c_{cf}$ associated with the complex formation rule in (4.2) in the sense that both have units of $molec^{-1}\ time^{-1}$ and can be computed from the macroscopic and deterministic constant used in ODEs. A very important similarity between these two constants is that both are diffusion limited. Nevertheless, the upper bound in the case of $c_{gon}$ is higher because of the one-dimensional diffusion effects due to the sliding of the transcription factor along the DNA. The upper bound to $c_{gon}$ is approximately $10^{10}$–$10^{11}$ $M^{-1}\, sec^{-1}$. This value is actually the upper bound for the corresponding macroscopic and deterministic constant $k_{gon}$ used in ODEs. In order to use this value for our constant $c_{gon}$ it is necessary to convert it into $molec^{-1}\ sec^{-1}$ using the volume of the compartment where the reaction takes place and Avogrado's number as described in section 2.6. The constant $c_{goff}$ associated with the rule describing the debinding of the transcription factor from the gene is similar to the constant, $c_{cd}$, associated with the dissociation rule in (4.2) in the sense that it has units of $time^{-1}$, coincides with the constant used in ODEs and has got no general upper bound as it is determined by the strength of the chemical bonds between the transcription factor and the gene.

### 4.3.2. Specification of Transcription Networks using Strings

The use of individual objects to represent the complex structure of genes in the DNA and RNA and the use of single rules to describe the complex processes of transcription and translation have been very successful in modelling transcription networks. Nevertheless, in order to model accurately the processes involved in transcriptions networks using this approach, it was necessary to incorporate in an artificial manner some features, like delays, that emerge from the linear organisation of genes in the DNA and the time scale difference between transcription and translation. For instance, in prokaryotes genes codifying proteins involved in similar tasks are arranged together in a piece of DNA called operon so that they are transcribed in a single strand of mRNA. The order in which these genes are placed in operons is relevant, as it determines the order in which they are transcribed, and thus the order in which their protein products become available. Therefore, it is necessary to specify genes using linear structures like strings if one wants to produce relevant models of transcription networks.

Another important fact that is overlooked in approaches describing transcription and translation as individual processes is that in prokaryotes shortly after transcription has started and before it is over ribosomes can bind to the growing mRNA and start translation. Furthermore, there can be many processes of transcription and translation going on at the same time. Summing up, transcription and translation are concurrent and parallel processes that are difficult to specify using individual objects and single step rules. The specification of transcription and translation as concurrent and parallel processes has been addressed already in process algebra based approaches; more precisely in the $\pi$-calculus [74].

Finally, another problem that arises from the use of single step rules for the description of transcription, translation and protein-protein interaction is the difference in the time scales of these processes as can be seen in the table of Figure 2.3 in chapter 2. While protein-protein interactions take seconds, transcription and translation may need half an hour to complete. This difference in the time scales produces a difference of many orders of magnitude in the stochastic constants associated with the corresponding rules. When this is the case the applicability of Gillespie's theory of stochastic kinetics is questionable as the difference among the stochastic constants distorts appreciably the evolution of the system. In this section this problem is solved by decomposing the processes of transcription and translation into simpler interactions whose time scales are similar to those of protein-protein interactions.

In what follows we propose the use of strings to represent the linear structure of strands of DNA and RNA and the use of rewriting rules on multisets of objects and strings to describe the binding and debinding of transcription factors to genes and the processes of transcription and translation as concurrent and parallel processes.

- **Binding and debinding of transcription factors to specific sites on the DNA:**

  As discussed in the previous section the rate of transcription of most genes is regulated by specific proteins called transcription factors. These proteins bind to specific regions of the genes called operators. These sites are normally located around the region where the RNA polymerase, the protein that directs transcription, binds to start transcription. The binding of a transcription factor to an operator produces a change in the conformation of that region of the gene that can yield two different and opposite effects. On the one hand, the binding of a transcription factor to an operator can help the RNA polymerase to bind to the gene activating transcription. On the other hand, the binding of a transcription factor can block the region where the RNA polymerase binds and therefore it represses transcription. The binding of transcription factors to operators is reversible.

  The P system specification schema in 4.9 describes the binding and debinding of a transcription factor represented by the object $Tf$ to an operator specified by the substring $\langle site \rangle$. The rule $r_{tfb}$ describes the binding of the transcription factor $Tf$ to an operator, $\langle site \rangle$. The effect of this rule consists of the consumption of an object $Tf$ and the rewriting of the substring $\langle site \rangle$ representing the free operator with the substring $\langle site' \rangle$ representing the operator occupied by the transcription factor.

The reverse process of debinding a transcription factor from an operator is described in rule $r_{tfd}$. An application of this rule produces an object $Tf$ and the replacement of the substring $\langle site' \rangle$ with the substring $\langle site \rangle$.

$$\boxed{\begin{array}{lll} r_{tfb}: & [\ Tf + \langle site \rangle\ ]_l \xrightarrow{c_{tfb}} [\ \langle site' \rangle\ ]_l & prop(r_{tfb}) = c_{on}|Tf||\langle site \rangle| \\ r_{tfd}: & [\ \langle site' \rangle\ ]_l \xrightarrow{c_{tfd}} [\ Tf + \langle site \rangle\ ]_l & prop(r_{tfd}) = c_{off}|\langle site' \rangle| \end{array}} \quad (4.9)$$

The constant $c_{tfb}$ associated with rule $r_{tfb}$ represents the affinity between the transcription factor and the operator. Similar to the previous constant associated with rules describing the formation of molecular complexes, this constant has units of $molec^{-1}time^{-1}$ and it is diffusion limited. The constant $c_{tfd}$ associated with rule $c_{tfd}$ has unit of $time^{-1}$ and is determined by the strength of the bonds formed between the operator and the transcription factor. Therefore, there is no general upper bound for it.

- **Transcription:**

  Transcription is the process by which a specific protein called **RNA polymerase** syntheses from a template sequence of DNA the corresponding complementary strand of RNA. During synthesis of RNA, the four-base language of DNA, consisting of sequences of A (adenine), G (guanine), C (cytosine) and T (thymine), is copied, or transcribed, into the four-base language of RNA. The four bases in RNA are identical to those of DNA except for the case of T (thymine) which is replaced with U (uracil) in the RNA.

  To carry out transcription, RNA polymerase performs several distinct functions, as depicted in Figure 4.10, namely, transcription initiation, mRNA elongation and transcription termination. In what follows a detailed description of the P system schemas used to specify these interactions are presented. As mentioned before these schemas will consists of rewriting rules on multisets of objects and strings. In this respect, the RNA polymerase will be represented by the object RNAP. The linear structure of genes in the DNA will be described by strings whose substrings represent relevant regions or sites in the gene. For instance, the string in (4.10) will be used to illustrate the effect of the rules presented in this section.

$$\overbrace{prom.site_{ini}.site_{mid}.\cdots.site_{mid}.site_{ter}}^{gene} \tag{4.10}$$

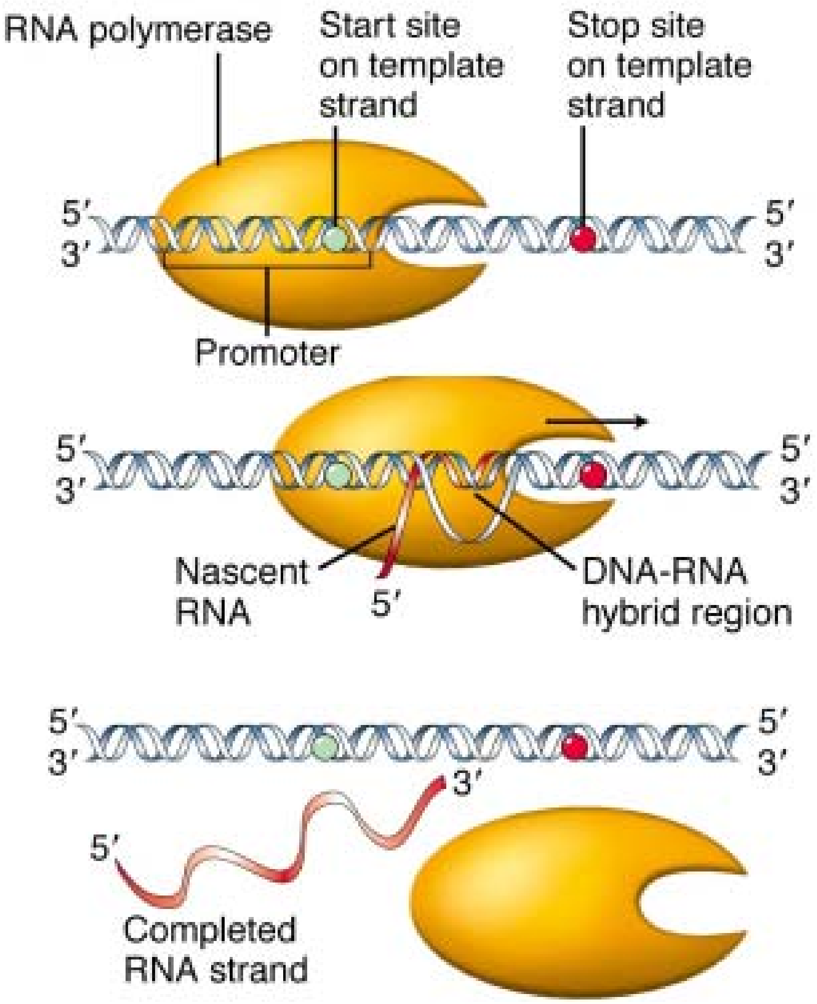

Figure 4.10.: Transcription stages taken from [76]

– First the RNA polymerase recognises and binds reversibly to a specific site at the beginning of the gene, called the *promoter*, represented by the substring $\langle prom \rangle$.

  The rewriting rules in (4.11) describe the processes of recognition of the promoter by the RNAP, the binding of the RNAP to the promoter and the debinding of the RNAP from the promoter. These rewriting rules act on multisets of objects and strings.

  The recognition of the promoter and binding of the RNAP to the promoter is described in rule $r_{rb}$. An application of this rule in a compartment of the type specified by the label $l$ consumes an object RNAP and replaces $\langle prom \rangle$ with $\langle prom.\text{RNAP} \rangle$ in a string which contains $\langle prom \rangle$ as a substring. This produces the insertion of the object RNAP after the substring $\langle prom \rangle$ in the corresponding string which describes the binding of the RNAP to the promoter of the gene. The propensity of this rule is computed using the mass action law, as with previous rules describing the formation of a molecular complex of any kind. The constant $c_{rb}$ measures the affinity between the RNAP and the promoter. Similarly to the case of the binding of transcription factors to DNA sites this constant is diffusion limited.

The debinding of the RNAP from the promoter is specified in rule $r_{rd}$. According to this rule in a compartment of type $l$ the substring $\langle prom.\text{RNAP} \rangle$ is rewritten with the substring $\langle prom \rangle$ and an object RNAP is produced. An application of this rule produces the removal of the object RNAP from the string where $\langle prom \rangle$ is located, representing the dropping of the RNAP from the promoter. The constant $c_{rd}$ associated with this rule is similar to previous constants associated with rules describing the dissociation of complexes of any type. In this respect, there is no general upper bound for it since its value represents the strength of the bonds between the RNAP and the promoter. The strength of the bond between the RNAP and the promoter does not only depend on the nucleotide[5] sequence of the promoters; transcription factors can bind to the promoter enhancing the bounds.

$$\boxed{\begin{array}{rl} r_{rb}: & [\ \text{RNAP} + \langle prom \rangle\ ]_l \xrightarrow{c_{rb}} [\ \langle prom.\text{RNAP} \rangle\ ]_l \\ & \quad prop(r_{rb}) = c_{on}|\text{RNAP}||\langle prom \rangle| \\ r_{rd}: & [\ \langle prom.\text{RNAP} \rangle\ ]_l \xrightarrow{c_{rd}} r[\ \text{RNAP} + \langle prom \rangle\ ]_l \\ & \quad prop(r_{rd}) = c_{rd}|\langle prom.\text{RNAP} \rangle| \end{array}} \tag{4.11}$$

Below we present an example of the effect of an application of the rule $r_{rb}$

[5] Nucleotides are the monomoners or elementary molecules which form the nucleid acids, DNA and RNA.

on the string introduced in (4.10).

$$\begin{array}{l}\text{RNAP} \\ prom.site_{ini}.site_{mid}.\cdots.site_{mid}.site_{ter} \\ \qquad\qquad\qquad\qquad \Downarrow \\ prom.\text{RNAP}.site_{ini}.site_{mid}.\cdots.site_{mid}.site_{ter}\end{array}$$

– During **transcription initiation** the RNA polymerase melts the DNA strands in order to make the bases in the template strand available for base pairing. After several ribonucleotides [6] have been polymerised[7] the RNA polymerase dissociates from the promoter which becomes available for other polymerases to bind and start transcription.

This process is described by the rewriting rule on multisets of strings in (4.12). This rule specifies the melting of the double strand of the DNA and the transcription of the first nucleotides. These nucleotides are represented by the substring $\langle site_{ini}\rangle$. The complementary ribonucleotides are represented by the substring $\langle\overline{site}_{ini}\rangle$ which mark the beginning of the nascent (growing) mRNA. The effect of an application of the rule $r_{ti}$ in a compartment of type $l$ consists of the replacement of the substring $\langle \text{RNAP}.site_{ini}\rangle$ with the substring $\langle site_{ini}.\overline{site}_{ini}.\text{RNAP}\rangle$ in the string representing the gene.

$$\boxed{\begin{array}{ll} r_{ti}: & [\ \langle \text{RNAP}.site_{ini}\rangle\ ]_l \xrightarrow{c_{ti}} [\ \langle site_{ini}.\overline{site}_{ini}.\text{RNAP}\rangle\ ]_l \\ & prop(r_{ti}) = c_{ini}|\langle \text{RNAP}.site_{ini}\rangle| \end{array}} \tag{4.12}$$

An example of the application of rule $r_{ti}$ is given below. Note that after an application of rule $r_{ti}$ the substring $\langle prom\rangle$ is free so another object $RNAP$ representing an RNA polymerase can bind to it. In this manner the binding of RNA polymerases to the promoter of a gene which is already

[6] The ribonucleotides are the specific nucleotides forming the RNA.

[7] Polimerisation is the process of bonding monomers to form longer chains called polymers.

being transcribed can be described using our approach.

$$\begin{gathered}
\mathit{prom}.\mathrm{RNAP}.\mathit{site}_{\mathit{ini}}.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{ter}}\\
\Downarrow\\
\mathit{prom}.\mathit{site}_{\mathit{ini}}.\overline{\mathit{site}}_{\mathit{ini}}.\mathrm{RNAP}.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{ter}}
\end{gathered}$$

– During the stage of **strand elongation**, RNA polymerase moves along the template DNA adding nucleotides to the nascent (growing) RNA chain. Although, the growing mRNA hangs from the RNA polymerase and is not part of the DNA; in our specification using strings, the substring representing the growing mRNA is part of the string which represents the DNA. Nevertheless, different symbols will be used to specify DNA sites and RNA sites so the growing mRNA can be easily identified. (4.13) depicts how the growing mRNA hanging from the RNAP is specified as part of the string representing the gene.

$$\begin{array}{c}
\mathit{mRNA}\quad\left\{\begin{array}{l}\overline{\mathit{site}}_{\mathit{ini}}\\ \overline{\mathit{site}}_{\mathit{mid}}\\ \quad\vdots\\ \overline{\mathit{site}}_{\mathit{mid}}\end{array}\right.\\
\mathit{prom}.\mathit{site}_{\mathit{ini}}.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{mid}}.\quad\mathrm{RNAP}\quad.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{ter}}\\
|||\\
\mathit{prom}.\mathit{site}_{\mathit{ini}}.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{mid}}.\overbrace{\overline{\mathit{site}}_{\mathit{ini}}.\overline{\mathit{site}}_{\mathit{mid}}.\cdots\overline{\mathit{site}}_{\mathit{mid}}}^{\mathit{mRNA}}.\mathrm{RNAP}.\mathit{site}_{\mathit{mid}}.\cdots.\mathit{site}_{\mathit{ter}}
\end{array}\tag{4.13}$$

The rewriting rule $r_{el}$ in (4.14) describes the process of mRNA elongation. The substring $\langle\overline{\mathit{site}}_{\mathit{ini}}.w.\mathrm{RNAP}.\mathit{site}_{\mathit{mid}}\rangle$ represents the situation when the RNA polymerase (RNAP) with a partially formed chain of mRNA, $\langle\overline{\mathit{site}}_{\mathit{ini}}.w\rangle$, is ready to transcribe the next site in the DNA, $\langle\mathit{site}_{\mathit{mid}}\rangle$. Note that $w$ is a string representing the growing mRNA consisting only of symbols for RNA sites and ribosomes. The substring $\langle\overline{\mathit{site}}_{\mathit{ini}}\rangle$ marks the end of the growing mRNA.

The addition of newly transcribed nucleotides is achieved by adding the substring $\langle\overline{\mathit{site}}_{\mathit{mid}}\rangle$ to the substring representing the growing mRNA, $\langle\overline{\mathit{site}}_{\mathit{ini}}.w\rangle$.

The movement of the RNA polymerase along the DNA leaving behind the already transcribed sites is described by moving the substring $\langle site_{mid} \rangle$ from immediately ahead of the symbol RNAP to the end of the growing mRNA represented by the substring $\langle \overline{site}_{ini} \rangle$.

All this is achieved by rewriting the substring $\langle \overline{site}_{ini}.w.\text{RNAP}.site_{mid} \rangle$ with the substring $\langle site_{mid}.\overline{site}_{ini}.w.\overline{site}_{mid}\text{RNAP} \rangle$ in the string representing the gene being transcribed as it is described in (4.14).

$$\boxed{\begin{array}{ll} r_{el}: & [\ \langle \overline{site}_{ini}.w.\text{RNAP}.site_{mid} \rangle\ ]_l \xrightarrow{c_{el}} [\ \langle site_{mid}.\overline{site}_{ini}.w.\overline{site}_{mid}\text{RNAP} \rangle\ ]_l \\ & prop(r_{el}) = c_{el} |\langle \overline{site}_{ini}.w.\text{RNAP}.site_{mid} \rangle| \end{array}} \tag{4.14}$$

The constant $c_{el}$ is used to compute the propensity of the rule $r_{el}$. This is done in the same manner as for the previous first order rules. This constant describes the average number of times the RNA polymerase transcribes a site of the length of $site_{mid}$. This value can be estimated since it is know that RNA polymerase moves along the DNA synthesising RNA at a rate of approximately 1000 nucleotides per minute.

Below an example showing the effect of an application of rule $r_{el}$ is presented. Note that after one application of rule $r_{el}$ the symbol $site_{mid}$ is left behind and becomes available for another step representing the transcription of this site by another RNA polymerase. In this way the parallel transcription of the same gene by many RNA polymerases can be modelled using our approach.

$$prom.site_{ini}.site_{mid}.\cdots .site_{mid}.\overline{site}_{ini}.\ \overbrace{\overline{site}_{mid}.\cdots \overline{site}_{mid}}^{w}\ .\text{RNAP}.site_{mid}.\cdots .site_{ter}$$

$$\Downarrow$$

$$prom.site_{ini}.site_{mid}.\cdots .site_{mid}.site_{mid}.\overline{site}_{ini}.\ \overbrace{\overline{site}_{mid}.\cdots \overline{site}_{mid}}^{w}\ .\overline{site}_{mid}.\text{RNAP}.\cdots .site_{ter}$$

– The last stage in RNA synthesis is **transcription termination** when a completed RNA molecule is released from the RNA polymerase and the polymerase dissociates from the template. There are specific sequences in the template DNA that mark where RNA polymerase must terminate transcription.

The rule $r_{ter}$ in (4.15) describes the last step in transcription. The termination site on the DNA marking the place where RNA polymerase must dissociate is represented by the symbol $\langle\ site_{ter}\ \rangle$. The situation when the RNA polymerase, RNAP, with a growing mRNA reaches a termination site is represented by the substring $\langle \overline{site}_{ini}.w.\text{RNAP}.site_{ter}\rangle$. The dissociation of the RNA polymerase from the DNA is described by rewriting the substring $\langle \overline{site}_{ini}.w.\text{RNAP}.site_{ter}\rangle$ with $\langle site_{ter}\rangle$. The release of the RNA polymerase is specified by the production of an object RNAP. Finally, the release of a completed mRNA is represented by the production of a new string $\langle \overline{site}_{ini}.w.\overline{site}_{ter}\rangle$ where the symbols $\overline{site}_{ini}$ and $\overline{site}_{ter}$ mark the beginning and end of the newly formed mRNA.

$$\boxed{\begin{array}{ll} r_{ter}: & [\ \langle \overline{site}_{ini}.w.\text{RNAP}.site_{ter}\rangle\ ]_l \xrightarrow{c_{ter}} [\ \text{RNAP} + \langle site_{ter}\rangle;\ \langle \overline{site}_{ini}.w.\overline{site}_{ter}\rangle]_l \\ & prop(r_{ter}) = c_{ter}|\langle \overline{site}_{ini}.w.\text{RNAP}.site_{ter}\rangle| \end{array}} \tag{4.15}$$

The constant $c_{ter}$ is used to compute the propensity of the rule $r_{ter}$. This constant is similar to the constant $c_{el}$ as it describes the average number of times the RNA polymerase transcribes a site of the length of $\langle site_{ter}\rangle$.

Below an example showing the effect of an application of rule $r_{ter}$ is presented. Note that after an application of the rule $r_{ter}$ the object RNAP becomes available to bind again to the substring representing the promoter and start the process of transcription.

$$prom.site_{ini}.site_{mid}.\cdots\ .site_{mid}.\overline{site}_{ini}.\overbrace{\overline{site}_{mid}.\cdots\overline{site}_{mid}}^{w}.\text{RNAP}.site_{ter}$$

$\Downarrow$

RNAP

$$\overline{site}_{ini}.\overbrace{\overline{site}_{mid}.\cdots\overline{site}_{mid}}^{w}.\overline{site}_{ter}$$

$$prom.site_{ini}.site_{mid}.\cdots\ .site_{mid}.site_{ter}$$

- **Translation:**

Translation is the whole process by which the nucleotide sequence of an mRNA is used to order and join the amino acids[8] in a polypeptide chain. This sequence of amino acid determines the three-dimensional structure of the polypeptide chain and its activity. Therefore, translation is critical to the production of functional proteins and hence to the proper functioning of cells.

During protein synthesis the four-base language of RNA is translated into the 20 amino acid language of proteins. The genetic code used by cells is a triplet code, where every three nucleotides in the mRNA or codon corresponds either with an amino acid or a termination point.

Ribosomes, complexes of an special RNA called rRNA (ribosomal RNA) and proteins, direct the formation of polypeptides. Similarly to transcription, the complex process of translation can be divided into three stages, initiation, elongation and termination, see Figure 4.11.

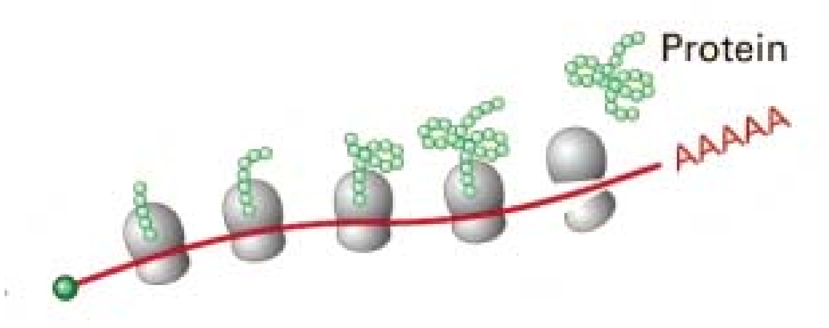

Figure 4.11.: Translation stages taken from [76]

- In prokaryotes, shortly after RNA polymerase starts transcription and before it is over, ribosomes bind to the growing mRNA to start translation. Synthesis of all polypeptide chains in prokaryotic and eukaryotic cells begins with the amino acid methionine. In most mRNAs, the codon codifying this amino acid and thus marking the start point for translation is AUG[9]. In our P system specification schema for translation initiation, the initiation codon is specified using the substring $\langle \overline{site_{ini}} \rangle$ in the string describing the mRNA. Ribosomes are represented using the object Rib. The rule $r_{tli}$ describes the binding of a ribosome, Rib, to the initiation codon, $\langle \overline{site_{ini}} \rangle$. An application

[8] Amino acids are the monomers or building blocks of proteins. Proteins are synthesised via polymerisation of amino acids through peptide bonds yielding a chain which constitutes the primary structure of proteins.

[9] Although AUG is the initiator codon in most cases, in a few bacterial mRNAs GUG and CUG are used as initiator codons.

of this rule in a compartment of type $l$ consumes an object Rib and rewrites the substring $\langle \overline{site}_{ini} \rangle$ with $\langle\ \overline{site}_{ini}.\mathrm{Rib}\ \rangle$ in the string describing the mRNA. This produces the insertion of the object Rib after the substring $\langle \overline{site}_{ini} \rangle$ which describes the initiation of a translation process by a ribosome.

$$\boxed{\begin{array}{ll} r_{tli}: & [\ \mathrm{Rib} + \langle \overline{site}_{ini} \rangle\ ]_l \xrightarrow{c_{tli}} [\ \langle \overline{site}_{ini}.\mathrm{Rib} \rangle\ ]_l \\ & prop(r_{tli}) = c_{tli} |\mathrm{Rib}| |\langle \overline{site}_{ini} \rangle| \end{array}} \tag{4.16}$$

The constant $c_{tli}$ associated with the rule in (4.16) represents the affinity between a ribosome and the initiation codon specified by the substring $\langle \overline{site}_{ini} \rangle$.

Next we present an example of the effect of an application of rule $r_{tli}$. Note that in the string used in this example there is a symbol RNAP which indicates that transcription is in process and has not yet finished. This does not prevent translation from initiation which is the case in prokaryotes. Therefore, in our approach it is possible to specify transcription and translation as concurrent and parallel processes.

Rib

$prom.site_{ini}.site_{mid}.\cdots.site_{mid}.\overline{site}_{ini}.\overline{site}_{mid}.\cdots\overline{site}_{mid}.\mathrm{RNAP}.site_{mid}.\cdots.site_{ter}$

$\Downarrow$

$prom.site_{ini}.site_{mid}.\cdots.site_{mid}.\overline{site}_{ini}.Rib.\overline{site}_{mid}.\cdots\overline{site}_{mid}.\mathrm{RNAP}.site_{mid}.\cdots.site_{ter}$

– Ribosomes direct elongation of a polypeptide by moving along an mRNA chain interacting with various proteins factors and tRNAs[10]. The key steps in elongation are the entry of each amino acid, the formation of a peptide bond[11] with the preceding amino acids and the movement or translocation of the ribosome on the mRNA being translated.

In our approach the growing chain of amino acids is not specified as our focus is on the release of the protein once translation is finished. Therefore, in elongation we only specify the movement of ribosomes along the mRNA.

[10] tRNAs are small molecules of special RNA that transfer specific amino acids to a growing polypeptide chain through interactions with ribosomes

[11] A peptide bond is formed between two amino acids when the carboxyl group of one of them reacts with the amino group of the other group releasing a molecule of water.

The rule in (4.17) describes a step of elongation. The substring $\langle \mathrm{Rib}.\overline{site}_{mid} \rangle$ describes the situation when a ribosome, Rib, is posed to read the next piece of mRNA, $\langle \overline{site}_{mid} \rangle$ and attach the corresponding amino acids to the growing polypeptide. The translocation of a ribosome along the mRNA is achieved by rewriting the substring $\langle \mathrm{Rib}.\overline{site}_{mid} \rangle$ with the substring $\langle \overline{site}_{mid}.\mathrm{Rib} \rangle$.

$$\boxed{\begin{array}{ll} r_{tle}: & [\ \langle \mathrm{Rib}.\overline{site}_{mid} \rangle\ ]_l \xrightarrow{c_{tle}} [\ \langle \overline{site}_{mid}.\mathrm{Rib} \rangle\ ]_l \\ & prop(r_{tle}) = c_{tle} |\langle \mathrm{Rib}.\overline{site}_{mid} \rangle| \end{array}} \tag{4.17}$$

The constant $c_{tle}$ associated with the rule $r_{rle}$ represents the average number of sites represented by $\overline{site}_{mid}$ are translated by time unit. This constant can be deduced using the fact that ribosomes add new amino acids to a growing polypeptide at a rate of approximately three to five amino acids per second.

Below we present an example of an application of rule $r_{tle}$ in a string representing a gene that is being transcribed and translated at the same time:

$$prom.site_{ini}.site_{mid}.\cdots .site_{mid}.\overline{site}_{ini}.Rib.\overline{site}_{mid}.\cdots \overline{site}_{mid}.\mathrm{RNAP}.site_{mid}.\cdots .site_{ter}$$

$$\Downarrow$$

$$prom.site_{ini}.site_{mid}.\cdots .site_{mid}.\overline{site}_{ini}.\overline{site}_{mid}.Rib.\cdots \overline{site}_{mid}.\mathrm{RNAP}.site_{mid}.\cdots .site_{ter}$$

– The final stage in the production of a polypeptide chain consists in translation termination. Ribosomes dissociate from a mRNA and release a completed polypeptide chain when they reach specific codons marking termination points. There are three termination codons UAA, UGA and UAG.

  This last process in the translation of a mRNA into a polypeptide is described by the rule $r_{tlt}$ in (4.18). The situation when a ribosome reaches a termination codon is represented by the substring $\langle \mathrm{Rib}.\overline{site}_{ter} \rangle$. The dissociation of the ribosome from the mRNA and the release of the polypeptide chain are described by rewriting the substring $\langle \mathrm{Rib}.\overline{site}_{ter} \rangle$ with the substring $\langle \overline{site}_{ter} \rangle$ in the string representing the mRNA and the production of an object Rib and Prot specifying a free ribosome and a newly produced protein, respectively.

$$\boxed{\begin{aligned} r_{tlt}:\quad & [\ \langle \text{Rib}.\overline{site}_{ter}\rangle\ ]_l \xrightarrow{c_{tlt}} [\ \text{Rib} + \langle \overline{site}_{ter}\rangle;\ \text{Prot}\ ]_l \\ & prop(r_{tlt}) = c_{tlt}|\langle \text{Rib}.\overline{site}_{ter}\rangle| \end{aligned}} \tag{4.18}$$

The constant $c_{tlt}$ associated with rule $r_{tlt}$ and used to compute its propensity is similar to the previous constant $c_{tle}$ as it represents the average number of sites of the length of the site represented by $\langle \overline{site}_{ter}\rangle$ are translated by time unit.

Below we present an example of an application of rule $r_{tlt}$. Note that for the application of a rule of this type, the production of a complete mRNA between an initiation and a termination codon is necessary, which it is normally referred to as a **reading frame**.

$\overline{site}_{ini}.Rib.\overline{site}_{mid}.\cdots\overline{site}_{mid}.Rib.site_{ter}$

$\Downarrow$

Rib

Prot

$\overline{site}_{ini}.Rib.\overline{site}_{mid}.\cdots\overline{site}_{mid}.site_{ter}$

Note that our approach allows us to describe the translation of the same mRNA by many ribosomes which is the case in living cells. For instance, in the example above there are two symbols which representes two ribosomes translating the same strand of mRNA.

## 4.4. Systems Biology Markup Language and P System Models

This section addresses the translation of Systems Biology Mark-Up Language (SBML) Level 2 version 1 models into P systems models.

Computational modelling in cellular research is becoming increasingly important as a means of helping to better understand cellular functions by using software tools. Due to the large community of researchers working in the field they often face several problems. Normally, users need to work with complementary software tools like simulators, model checkers, parameter estimators, graphical tools, etc. and it becomes difficult to transfer a model from one tool to another. It is also usual to work with models published in peer-reviewed journals. These models, in spite of being sometimes accompanied by instructions for obtaining their definitions in electronic format, are not clearly identified or easily retrieved because each author may use a different representation language. Another problem appears when simulation software packages are no longer supported and models developed in these systems become unusable.

The diversity of approaches and methods in cellular modelling implemented by different software tool developers demands a common intermediate format, a *lingua franca*, enabling communication of the most essential aspects of a model. The Systems Biology Markup Language (SBML) was developed in an effort to address this demand. SBML is a format for representing biochemical networks and is intended to be convenient for computer software tools to generate and parse, thereby enabling communication of biochemical networks and interoperability between disparate modelling and simulation tools. It is intended to be independent of particular modelling approaches and should be appropriate for any modelling approach. It is essentially an eXtensible Markup Language (XML) encoding the main components of biochemical networks.

Here we will concentrate on the current specification, SBML Level 2 (version 1).An SBML model consists of the optional lists of functions, units, compartments, species, parameters, rules, reactions and events. This is the basic structure of an SBML Level 2 (version 1) specification:

```
<?xml version="3.0" encoding="UTF-8"?>
 <sbml xmlns="http://www.sbml.org/sbml/level2" level ="2" version ="1">
   <model id = " ">
      <listOfFunctionDefinitions>
                 ...
      </listOfFunctionDefinitions>
      <listOfUnitDefinitions>
                 ...
      </listOfUnitDefinitions>
      <listOfCompartments>
                 ...
      </listOfCompartments>
      <listOfSpecies>
                 ...
      </listOfSpecies>
      <listOfParameters>
                 ...
      </listOfParameters>
      <listOfRules>
```

```
                ...
      <listOfRules>
      <listOfReactions>
                ...
      </listOfReactions>
      <listOfEvents>
                ...
      </listOfEvents>
   </model>
</sbml>
```

Here we only describe the list of units, compartments, species, parameters and reactions, as these are sufficient for adequately describing models coming from most modelling approaches notably ours. For a complete description of SBML Level 2 version 1 see [135].

- The *list of units* allows definition and redefinition of the units used in the model. The units are a delicate issue in a model as it is the main point where mesoscopic-stochastic and macroscopic-deterministic models differ. Mesoscopic models normally use number of molecules as units while macroscopic models work with concentration units. By default SBML assumes the use of concentration units and this is changed by the following code:

```
<listOfUnitsDefinitions>
    <unitDefinition id="species">
        <listOfUnits>
            <uniy kind = "item" />
        </listOfUnits>
    </unitDefinition>
</listOfUnitsDefinitions>
```

- The *list of compartments* enumerates the compartments in the model. A model must have at least one compartment and each compartment should be given an `id`. One may also specify a name for the compartment, a size (or volume), units for the volume, etc. A hierarchical structure consisting of compartments embedded inside other compartments can be specified using the optional field `outside`. For example a model consisting of two nested compartments would be specified as follows:

```
<listOfCompartments>
    <compartment id = "id1" name = "cytoplasm" />
    <compartment id = "id2" name = "nucleus" outside = "id1" />
<\listOfCompartments>
```

- The *list of species* states all the molecular species of the model. Each species must have an `id` and can also be given a compartment where it is located, the initial amount or concentration and other attributes.

```
<listOfSpecies>
    <species id=" " compartment=" " initialAmount= " " />
    ...
</listOfSpecies>
```

- The *list of parameters* contains the definition of constants that are used in the model. It is also possible to declare local parameters in the kinetic law of reaction as discussed below. An example of a parameter list is given below:

```
<listOfParameters>
    <parameter id = "c1" value="0.01" />
    <parameter id = "c2" value="0.2" />
</listOfParameters>
```

- The *list of reactions* consists of an enumeration of the reactions of the system. A reaction in turn consists of a list of reactants, a list of products and a kinetic law. The list of reactants and the list of products contains the identifiers of the reactant and product species of the reaction. The kinetic law is a MathML encoding of the formula describing the rate of the reaction. Within the kinetic law local parameters can be specified. For example the code given below specifies a reaction with reactants `reactant1` and `reactant2`, product `product1` and kinetic law `K*reactant1*reactant2` where `K` is a local parameter.

```
<reaction id="r1" reversible ="false">
    <listOfReactants>
        <speciesReferences species="reactant1">
        <speciesReferences species="reactant2">
    </listOfReactants>
    <listOfProducts>
```

```
        <speciesReferences species="product1">
    </listOfProducts>
    <kineticLaw>
        <math xmlns="http://www-w3.org/1998/Math/MathML">
        <apply>
            <times/>
            <cn> K </cn>
            <ci> reactant1 </ci>
            <ci> reactant2 </ci>
        </apply>
        </math>
        <listOfParameters>
            <parameter id="K" value="0.002">
        </listOfParameters>
    </kineticLaw>
</reaction>
```

By using SBML different software tools can all operate on an identical representation of a model, reducing opportunities for errors in translation and assuring a common starting point for simulation and analysis.

SBML should not be regarded as an alternative to other representations, but simply as an electronic format which could in principle be used in conjunction with the specifications of any modelling framework. Also note that SBML representations should be generated and manipulated not by hand but rather by software tools which present the user a friendlier interface. In this work we will use CellDesigner, a structured diagram editor for drawing gene-regulatory and biochemical networks [37] which exports graphical models of cellular systems to SBML format.

### 4.4.1. Translating SBML Format into P Systems Models

Not all the components of an SBML Level 2 version 1 file can be translated into a P system model. Here we will only discuss how to translate the sections corresponding to *units*, *species*, *compartments* and *reactions*.

- **Units** are the first section to be analysed when translating an SBML file into a P system model. It is necessary to check the units of the initial amounts of the species and parameters. By default, SBML assumes concentration units of moles per liter. In this case it is necessary to expresse the initial amount of the species in molecules by multiplying the quantities given in the SBML file by the volume of the compartment where the species are located and by Avogradro's number. The conversion of deterministic-macroscopic constants into the corresponding mesoscopic-stochastic ones is done as discussed in section 2.6.

- **Compartments** are considered in SBML format as bounded regions in which species are located. In P system models, membranes do not necessarily correspond to real cellular membranes delimiting compartments. Instead they are used to define homogeneous regions where localised reactions take place. Here we will only consider the cases when the regions delimited by membranes in a P system model correspond to actual compartments or compartments surfaces. In this scenario given a compartment specification in SBML:

  ```
  <compartment id = "id1" name = "comp" outside="id0"  />
  ```

  Two membranes are generated: one with label `comp-surf` representing the compartment surface, $[\ ]_{comp-surf}$; and another one with label `comp`, representing the inner part of the compartment, $[\ ]_{comp}$. This second membrane will be embedded inside the first one, $[\ [\ ]_{comp}\ ]_{comp-surf}$ . The rest of the hierarchy in the membrane structure is inherited directly from the SBML specification using the field `outside` of each compartment.

- **Species** are treated in SBML format as simple, indivisible biochemical entities. This permits the specification of species as objects in the alphabet of a P system model. Species are also given an initial amount and a compartment where they are located. Since each compartment from the SBML format produces two membranes it is not obvious in which of them the objects representing species will be placed. Therefore given the specification of a species in SBML:

  ```
  <species id="s1" name = "species1"

           compartment="id-comp" initialAmount= "Init" />
  ```

  an object `species1` will represent this species in the alphabet of the P system model. In order to decide in which of the two membranes used to describe the compartment `id-comp` this object is placed it is necessary to determine the species that interact with `species1`. This is done by analysing the list of reactions. If `species1` reacts with species located in the compartment outside the compartment `id-comp` then we consider that it is located on the compartment surface and therefore we place the object in the membrane with label `comp-surf`. Otherwise it is located in the membrane with label `comp`. The initial amount `Init` is used to compute the multiplicity of the object `species1` in the initial multiset associated with the corresponding membrane.

- **Reactions** in SBML are described in terms of reactants, products and an optional kinetic law. SBML does not impose any restriction on the number of reactants and products. In principle P systems rules support any number of objects on the left and right hand side of the rules. Nevertheless, since Gillespie's theory of stochastic kinetics is used in this work the types of rules supported are limited to those discussed in sections 4.2 and 4.3.1. All these rules describe first or second order reactions. Although reactions with more than two reactants can be factorized in binary interactions SBML does not provide any information about the dynamics of these intermediary reactions preventing the use of this procedure. Therefore our study is limited to the translation of SBML models containing at most second order reactions.

  Another issue arises from the fact that, in SBML Level 2 version 1, compartments are attributes of species but not of reactions. Consequently, reactions are not associated with regions as it is the case in P systems models where the sets of rules are localised in the different regions defined by membranes. This makes the translation of the list of reactions of an SBML file into the rules of a P system model a delicate process. In order to localise a given reaction in a region defined by a membrane the first step is to determine which type of rule from the ones introduced in section 4.2 can be used to represent this reaction.

  Finally, the kinetic law associated with a reaction can be an arbitrary complex mathematical expression specified in MathML with no information as to whether the given rates are deterministic or stochastic. Thus the given kinetic law will not be taken into account and the correct propensities associated with the reactions will be computed using Gillespie's theory of stochastic kinetics with the parameter declared in the reaction as a local parameter.

  The following steps demonstrate how to translate the reactions from an SBML file into P system rules distinguishing between first and second order reactions.

  1. **First order reactions:** Reactions with a single reactant, $r$, can be described using different types of rules depending on the number and location of the products. The first step in translating a first order reaction consists of determining the membrane in which $r$ is placed. This was done before when translating the section corresponding to species. Let's assume that $r$ is associated with the membrane with label `l`. Since the only reactant is located in membrane $l$ it is natural to associate the rules that represent this

type of reaction with membrane $l$. The rule that can be used to represent a first order reaction depends on the number of products and their location.

- Reactions with a single product $p$ located in membrane `l`: In this case the following *transformation rule* is used to describe the reaction,

  $$[\ r\ ]_l \xrightarrow{c} [\ p\ ]_l$$

  The constant $c$ represents the local parameter defined in the section kinetic law of the analysed reaction.

- Reactions with a single product $p$ located in the membrane $l'$ outside membrane $l$: Here a *diffusion out rule* of the following form describes this type of reaction,

  $$[\ r\ ]_l \xrightarrow{c} p\ [\ \ ]_l$$

- Reactions with a single product $p$ located in a membrane embedded into membrane $l$ identified with the label $l'$: In this case the following *diffusion in rule* specifies this type of reaction,

  $$r\ [\ \ ]_{l'} \xrightarrow{c} [\ p\ ]_{l'}$$

- Reactions with two products, $p_1$ and $p_2$ both located in membrane $l$: The following *complex dissociation rule* represents this type of reaction,

  $$[\ r\ ]_l \xrightarrow{c} [\ p_1 + p_2\ ]_l$$

- Reactions with two products, $p_1$ located in the membrane $l'$ outside membrane $l$ and $p_2$ located in membrane $l$: In this situation the following *debinding rule* is used to specify this type of reaction,

  $$[\ r\ ]_l \xrightarrow{c} p_1\ [\ p_2\ ]_l$$

- Reactions with two products, $p_1$ located in membrane $l$ and $p_2$ located in a membrane embedded into membrane $l$ identified with the label $l'$: Here the following *releasing rule* describes this type of reaction,

  $$r\ [\ \ ]_{l'} \xrightarrow{c} p_1\ [\ p_2\ ]_{l'}$$

2. **Second order reactions:** Reactions with two reactants, $r_1$ and $r_2$, and a single product $p$ can be described using different types of rules depending on the location of $r_1, r_2$ and $p$. Thus, the first step of the translation consists

in determining the membrane or membranes where $r_1, r_2$ and $p$ are placed. This was done before when translating the section corresponding to species. Depending on the location of the reactants, $r_1$ and $r_2$, the rule representing the corresponding reaction will be associated with a different membrane. The rules used to represent a second order reaction depend on the location of $r_1, r_2$ and $p$ and are selected as follows.

- Reactions with both reactants $r_1$ and $r_2$ located in the same membrane $l$ and with a single product $p$ also located in the same membrane $l$: In this situation a complex formation rule of the following form is used to describe this type of reaction,

  $$[\ r_1 + r_2\ ]_l \xrightarrow{c} [\ p\ ]_l$$

  It is natural to associate this rule with membrane $l$ since all the objects involved in the rule are located in this membrane.
- Reactions with $r_1$ located in membrane $l$, which is outside membrane $l'$ where $r_2$ and the product $p$ are located: In this case a *binding rule* of the following type represents this type of reaction,

  $$r_1\ [\ r_2\ ]_{l'} \xrightarrow{c} [\ p\ ]_{l'}$$

  As discussed in section 4.2 this type of rule can be used to describe the interaction of a ligand $r_1$ and a receptor $r_2$, placed on the surface of a compartment, to produce a complex receptor-ligand. This interaction takes place in the compartment where the ligand is present, thus this type of rule is associated with membrane $l$ where $r_1$, which represents the ligand, is located.
- Reactions with $r_1$ and the product $p$ located in membrane $l$, which is outside membrane $l'$ where $r_2$ is located: The following *recruitment rule* is used to specify this type of reaction,

  $$r_1\ [\ r_2\ ]_{l'} \xrightarrow{c}\ p\ [\ \ ]_{l'}$$

  In section 4.2 recruitment rules were introduced to describe the process whereby a protein $r_1$ placed on the inner part of a compartment surface interacts with another protein $r_2$ floating freely in the compartment to form a complex $p$ that remains attached to the compartment surface. This interaction takes place in the inner part of the compartment which is represented in this rule by the membrane with label $l'$. Therefore this rule will be associated with the membrane with label $l'$.

# 5. Modularisation in P systems

*Although cell biology has evolved to function and not to be comprehensible, they show an inherent simplicity, by employing and combining a rather small set of basic building-block circuits, each for specific tasks.*

*Uri Alon*

## 5.1. Modules in Cellular Biology and P Systems

Cellular functions are rarely performed by an individual molecule; most biological functions arise as emergent behaviour from the interactions among modules made up of many molecular species. In this work a **module** is defined as a discrete entity which performs a specific function separable from those of other modules. The function of a module is a result of the interactions among their components, these functions are not easily predicted by studying the properties of their isolated components. This separation depends on chemical isolation, which can originate from spatial localisation in different compartments or from chemical specificity [53]. These both features can be easily represented in P systems using membranes for spatial localisation and rules for chemical specificity.

Modules can be isolated or connected to each other. Isolation allows the cell to carry out many diverse processes without interferences among them that would harm the cell, whereas connectivity allows higher level functions to be built by assembling different modules. P systems have proved to be a useful tool in studying self-assembling processes [16, 18], therefore they are a suitable framework to study how the modularisation and pattern of connections among the functional cellular modules achieve emergent behaviours such as amplification, adaptation, robustness, insulation, error correction, coincidence detection, etc.

In this chapter we will show how P systems can specify in a comprehensive and relevant manner functional modules in cellular systems. The modular specification of cellular systems allows us to identify the organisation and function of the living cell and to explain general principles that govern different self-assembly processes occurring in synthetic sciences like engineering or artificial life.

As mentioned before modularisation in cellular systems is due to chemical specificity and spatial location in different compartments.

- On the one hand, modularisation due to chemical specificity is represented in P systems using modules of rewriting rules on multisets of objects and strings of one of the following two forms:

$$obj_1 \, [\, obj_2 \,]_l \xrightarrow{c} obj_1' \, [\, obj_2' \,]_l \tag{5.1}$$

  where $obj_1, obj_2, obj_1', obj_2'$ are finite multisets of objects over an alphabet $O$ and $l$ a label specifying a compartment type. According to rules of this type, a multiset of objects $obj_1$ located outside a compartment of the type specified by label $l$ and a multiset $obj_2$ inside this compartment can be simultaneously replaced with a multiset $obj_1'$ and a multiset $obj_2'$ respectively.

$$[\, obj_1 + str_1; \ldots ; obj_p + str_p \,]_l \xrightarrow{c} [\, obj_1' + str_{1,1}' + \ldots str_{1,i_1}^{'}; \ldots ; obj_p' + str_{p,1}' + \ldots str_{p,i_p}' \,]_l \tag{5.2}$$

  where $obj_1, \ldots, obj_p$ and $obj_1', \ldots, obj_p'$ are finite multisets of objects over an alphabet $O$, $str_1, \ldots, str_p$ and $str_1', \ldots, str_p'$ are strings over $O$ and $l$ a label specifying a compartment type. According to rules of this type, the multisets $obj_1, \ldots, obj_p$ are replaced with the objects $obj_1', \ldots, obj_p'$ inside a compartment of the type specified by label $l$. And at the same time, the strings $str_1, \ldots, str_p$ are rewritten with the strings $str_1', \ldots, str_p'$ on the strings where $str_1, \ldots, str_p$ are substrings.

  The stochastic constant $c$ associated with each rule is used to compute its propensity according to the discussions in Chapter 4.

  **Definition 5.1 (P system Module).** A P system module is a set of rules of one of the form in (5.1) or (5.2) representing molecular interactions which occur repetitively in many cellular systems. A module is identified with a name and three sets of variables, $V$, $C$ and *Lab*. $V$ represents variables that can be instantiated using objects or strings which describe molecular entities. $C$ represents the stochastic constants associated with each rule. *Lab* specifies the labels of the compartments involved in the rules. Formally, a module, *mod*, with variables $V$, constants $C$ and labels $L$ will be written as $mod(V, C, L)$. More complex modules can be constructed from simple modules by applying set union.

- On the other hand, modularisation due to spatial location in different compartments is specified in P systems using a membrane structure; a hierarchical arrangement of membranes defining compartments where all the membranes but one must be included in a unique main membrane, the *skin*, which defines the *boundary* of the system with respect to the external environment. The membrane structure can be represented formally, as a rooted tree, where the nodes are called membranes, the root is called skin, and the inclusion of a membrane inside another one is represented by its node being the descendent of another one.

Recall that a P system specification is a construct

$$\Pi = (O, L, \mu, M_1, M_2, \ldots, M_n, R_1, \ldots, R_n)$$

where $O$ is a finite alphabet of symbols representing molecular entities; $L$ is a finite alphabet of labels representing types of compartments; $\mu$ is a membrane structure containing $n \geq 1$ membranes labelled with labels from $L$; $M_i = (l_i, w_i, s_i)$, for each $1 \leq i \leq n$, is the initial configuration of membrane $i$ with $l_i \in L$; $w_i$ a finite multiset of objects over $O$; and $s_i$ a multset of strings over $O$. Finally, $R_i$ the set of rules associated with compartment $i$ , for each $1 \leq i \leq n$, can be obtained as a finite union of modules representing molecular interactions involving this compartment.

Regarding modularisation due to spatial compartmentalisation, a P system specification can be seen as representing an individual entity such as a bacterium or eukaryotic cell and so the behaviour of a colony or tissue of such entities can be studied using colonies of P systems as it will be discussed in Chapter 8.

## 5.2. A Library of Basic P System Modules

In this section we introduce a library of basic P system modules representing protein-protein and gene control interactions that will be used to construct more complex modules.

### 5.2.1. Protein-protein Interaction Modules

Many important cellular processes depend on interactions between proteins. For example, the binding of a molecule to another one can alter (regulate) the activity of the complex which can be completely different from the activity of the single molecules.

- **Dimerisation Module:** Two molecules, $X$ and $Y$, can collide and stick together through noncovalent interactions to produce a complex $Z$. This complex in turn can dissociate back into $X$ and $Y$. In biochemistry, these reactions are referred as dimerisation.

$$dim(\{X,Y,Z\},\{c_1,c_2\},\{l\}) = \left\{ \begin{array}{c} [\,X+Y\,]_l \xrightarrow{c_1} [\,Z\,]_l, \\ [\,Z\,]_l \xrightarrow{c_2} [\,X+Y\,]_l \end{array} \right\} \tag{5.3}$$

- **Enzymatic reaction**: Some chemical reactions are assisted by some proteins called **enzymes**. The enzyme $X$ binds reversibly to the substrate $Y$. Once they are together forming a complex $Z$ the enzyme produces a change in the substrate and then they dissociate releasing the unchanged enzyme $X$ and the product $W$.

$$enz(\{X,Y,Z,W\},\{c_1,c_2,c_3\},\{l\}) = \left\{ \begin{array}{c} [\,X+Y\,]_l \xrightarrow{c_1} [\,Z\,]_l, \\ [\,Z\,]_l \xrightarrow{c_2} [\,X+Y\,]_l, \\ [\,Z\,]_l \xrightarrow{c_2} [\,X+W\,]_l \end{array} \right\} \tag{5.4}$$

When the substrate is available in such quantities that it does not affect the dynamics of the reaction the above module can be simplified by considering that the enzyme $X$ produces a product $Z$ not being consumed in the reaction.

$$prod(\{X,Z\},\{c_1\},\{l\}) = \{\ [\,X\,]_l \xrightarrow{c_1} [\,X+Z\,]_l\} \tag{5.5}$$

- **Degradation**: The cell machinery degrades molecules in order to control their activity.

$$deg(\{X\},\{c_1\},\{l\}) = \{\,[\,X\,]_l \xrightarrow{c_1} [\ \ ]_l\}$$

- **Diffusion**: Small molecules can readily move by passive simple diffusion across membranes without the aid of transport proteins. The rewriting rules in these modules constitute the specification in P systems of diffusion in and out of a compartment. This compartment is represented by square brackets with a label $l$, that identifies the type of the compartment. For diffusion in the object $X$ is moved from the compartment surrounding compartment $l$ inside the region defined by it, and viceversa for diffusion out from compartment $l$.

$$\begin{array}{l} diff_{in}(\{X\},\{c\},\{l\}) = \{X\ [\ \ ]_l \xrightarrow{c} [\,X\,]_l\} \\ diff_{out}(\{X\},\{c\},\{l\}) = \{[\,X\,]_l \xrightarrow{c} X\ [\ \ ]_l\} \end{array}$$

### 5.2.2. Gene Regulation Modules

We now present some modules for the specification of gene regulation control in prokraryotes constructed using previous modules. For simplicity, transcription and translation are represented using individual rules. Nevertheless, in living cells transcription and translation involve many interactions between RNA polymerase, DNA, mRNA and ribosomes that take place in a concurrent manner.

- **Constitutive expression**: We start by modelling *constitutive genes*; genes whose level of expression does not depend on transcription factors. This kind of genes are transcribed continually at a relatively constant level.

  In this case from the gene encoded in the DNA, $G$, the mRNA, $R$, is transcribed or produced using the module $\mathit{prod}(\{G, R\}, \{c_1\}, \{l\})$. The mRNA translation and production of the protein product associated with the gene, $P$, is specified in the module $\mathit{prod}(\{R, P\}, \{c_2\}, \{l\})$. The mRNA and protein degradation are represented by the modules $\mathit{deg}(\{R\}, \{c_3\}, \{l\})$ and $\mathit{deg}(\{P\}, \{c_4\}, \{l\})$.

$$\mathit{const}(\{G, R, P\}, \{c_1, \ldots, c_4\}, \{l\}) = \left\{ \begin{array}{ll} \mathit{prod}(\{G, R\}, \{c_1\}, \{l\}) & \cup \\ \mathit{prod}(\{R, P\}, \{c_2\}, \{l\}) & \cup \\ \mathit{deg}(\{R\}, \{c_3\}, \{l\}) & \cup \\ \mathit{deg}(\{P\}, \{c_4\}, \{l\}) & \end{array} \right\} \quad (5.6)$$

  This type of genes is also referred to as *unregulated genes*, and therefore the previous module can be also denoted using *UnReg*.

- **Positive regulation**: Unlike *constitutive genes*, *facultative genes* are only expressed when needed according to some signals received by the cell from its surroundings. *Activators* are transcription factors which bind to the promoter of genes and activate their expression by recruiting RNA polymerase.

  The activation of the gene $G$ after the binding with the activator $A$ is specified using the module $\mathit{dim}(\{A, G, G_{on}\}, \{c_1, c_2\}, \{l\})$. The processes of transcription and translation are represented using the same modules as the case of constitutive expression. Note that here the active form of the gene is producing the mRNA.

$$pos(\{A, G, G_{on}, R, P\}, \{c_1, \ldots, c_6\}, \{l\}) = \left\{ \begin{array}{ll} dim(\{A, G, G_{on}\}, \{c_1, c_2\}, \{l\}) & \cup \\ prod(\{G_{on}, R\}, \{c_3\}, \{l\}) & \cup \\ prod(\{R, P\}, \{c_4\}, \{l\}) & \cup \\ deg(\{R\}, \{c_5\}, \{l\}) & \cup \\ deg(\{P\}, \{c_6\}, \{l\}) & \end{array} \right\} \tag{5.7}$$

- **Negative regulation**: As opposed to positive regulation in some conditions cells do not need the protein product encoded by a gene; in this case this gene is turned off or repressed by transcription factors called *repressors*. *Repressors* bind to the promoter site of genes blocking it so that polymerase cannot bind to it and thus preventing genes from being transcribed. The blocking of gene $G$ after the binding of repressor $Rep$ to its promoter is specified using the module $dim(\{Rep, G, G_{off}\}, \{c_1, c_2\}, \{l\})$.

$$\begin{array}{l} neg(\{Rep, G, G_{off}, R, P\}, \{c_1, \ldots, c_6\}, \{l\}) = \\ = \left\{ \begin{array}{ll} dim(\{Rep, G, G_{off}\}, \{c_1, c_2\}, \{l\}) & \cup \\ prod(\{G, R\}, \{c_3\}, \{l\}) & \cup \\ prod(\{R, P\}, \{c_4\}, \{l\}) & \cup \\ deg(\{R\}, \{c_5\}, \{l\}) & \cup \\ deg(\{P\}, \{c_6\}, \{l\}) & \end{array} \right\} \end{array} \tag{5.8}$$

In this section we have presented a simple approach on P systems specification modularisation. The approach emphasises the benefits of a flexible and uniform modelling paradigm based on modules obtained either from simple P systems rules or other already defined modules. The method allows to specify a large class of problems and to reuse a model for other similar contexts.

## 5.3. Modelling Cellular Modules Combining P System Modules

Now we can start to specify and model more complex patterns that can be found in cellular systems by combining the basic modules introduced in the previous section. The first step in this task consist of the search of cellular modules. Recall that intuitively a cellular module is a recurring pattern in networks of interacting molecular entities that occurs far more often than at random [4].

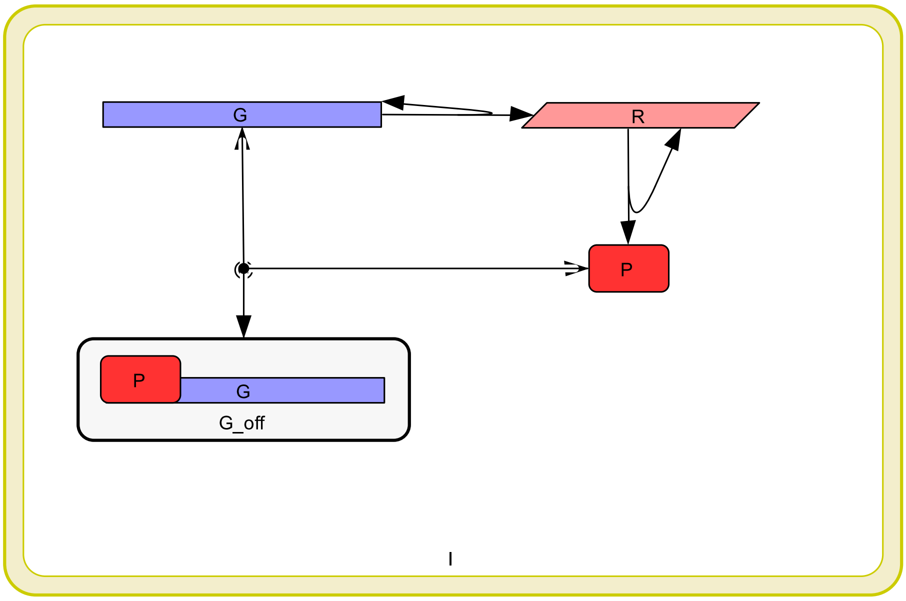

Figure 5.1.: Negative autoregulation

In this section we will use only autoregulation in transcription networks to illustrate our approach.

### 5.3.1. Negative Autoregulation

Negative autoregulation is a pattern in transcription networks occurring in about half of the repressors in Escherichia coli [122]. It consists of a gene whose protein product acts as a transcription factor repressing its own transcription, Figure 5.1.

Negative autoregulation (NAR) can be obtained from the *negative regulation* module in 5.8 by constraining the variables representing the repressor and the protein product to be instantiated with the same objects.

$$NAR(\{G, G_{off}, R, P\}, \{c_1, \dots, c_6\}, \{l\}) = neg(\{P, G, G_{off}, R, P\}, \{c_1, \dots, c_6\}, \{l\}) \tag{5.9}$$

In what follows we will try to find properties of the negative autoregulation module which have been the reason why evolution has selected it as a recurrent pattern in cellular systems. In order to elucidate these properties the dynamics of the NAR module will be compare with the dynamics of the module representing an unregulated gene or constitutive gene (UnReg). To make a meaningful comparison between two different designs, one must carried the experiments with equivanlence of as many parameters as possible between the alternative designs, this comparison methodology is referred to as **mathematically controlled comparison** [113, 112]. In this work PRISM will be used to compare the dynamics of different P system modules following the methodology introduced in section 3.4.

The research of the dynamics of modules focuses in the study of how fast or slow the number of the molecular specifies involved in the module change over time. An important measure for the characterisation of these dynamics is the response time. The response time is generally defined as the time to reach halfway between the initial and the final levels in a dynamic process.

Following a mathematically controlled comparison we will study the dynamics of the NAR and UnReg modules using the same stochastic constants for the rules describing translation and degradation of the mRNA and protein. More specifically, the stochastic constants for the UnReg module will be $c_1^{UnReg} = 2.89 \times 10^{-3}\ molec\ sec^{-1}$, $c_2^{UnReg} = 0.04\ sec^{-1}$, $c_3^{UnReg} = 2 \times 10^{-3}\ sec^{-1}$, and $c_4^{UnReg} = 5.78 \times 10^{-4}\ sec^{-1}$, and the stochastic constants for the NAR module will be $c_1^{NAR} = 0.03\ molec^{-1} sec^{-1}$, $c_2^{NAR} = 5.78 \times 10^{-2}\ sec^{-1}$, $c_3^{NAR} = 0.13\ molec\ sec^{-1}$ , $c_4^{NAR} = c_2^{UnReg}$, $c_5^{NAR} = c_3^{UnReg}$, and $c_6^{NAR} = c_4^{UnReg}$. In Figure 5.2 the expected evolution of the number of proteins over time for the NAR module and UnReg module with stochastic constants enumerated before is depicted. Note that the dynamics of both modules reach a steady state of about 100 protein molecules. The constant $c_3^{NAR}$ was chosen so both modules share the same steady state as we need to have modules having as many parameters equal as possible.

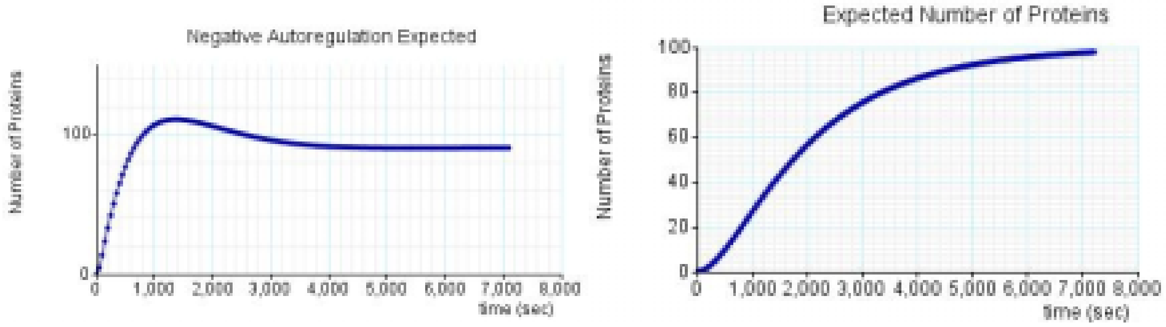

Figure 5.2.: Expected evolution of the number of proteins over time for the NAR and UnReg modules.

The property we will compare is the response time in both modules. It has been found that negative autoregulation speeds the response time in transcription networks [109]. We will check this property for our stochastic and discrete P system modules using PRISM. The dynamics of both modules, with respect to the number of proteins, go from zero proteins to a steady state of 100 proteins. Therefore, the response time in this case is the expected time to reach 50 proteins. In Figure 5.3, the expected responses time for both the NAR and UnReg modules are compared using the expected evolution of the number of proteins over time. Note that indeed the response time in the NAR module is much shorter.

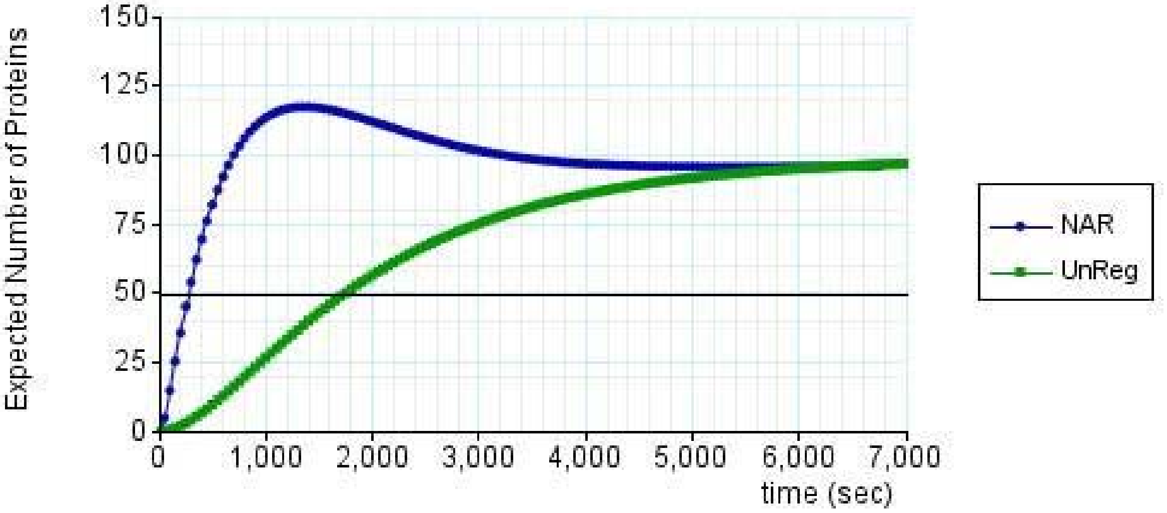

Figure 5.3.: Negative autoregulation accelerates response times

PRISM is not only limited to the computation of the expected behaviour of systems. For instance, here we can compute the probability of exceeding 50 protein molecules over time which is associated with the response time. This is shown in Figure 5.4. Note that the probability of reaching 50 molecules in the case of negative autoregulation is appreciable within the first hundreds of seconds; whereas for the case of unregulated expression this probability is not appreciable until the first thousands of seconds.

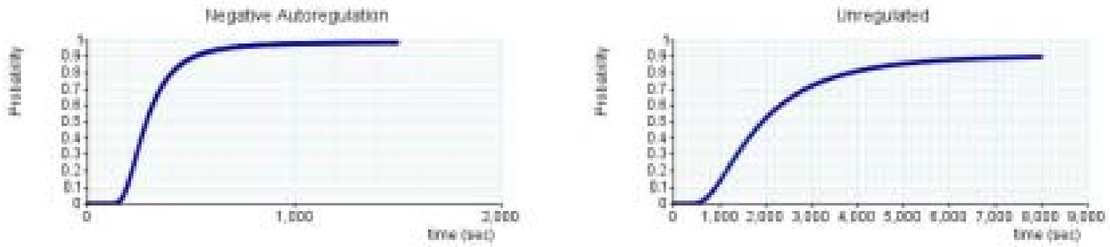

Figure 5.4.: Probability of the response time in negative autoregulation (left) and unregulated expression (right).

### 5.3.2. Positive Autoregulation

Positive autoregulation is a pattern in transcription networks which consists in a gene whose protein product acts as a transcription factor enhancing its own transcription,

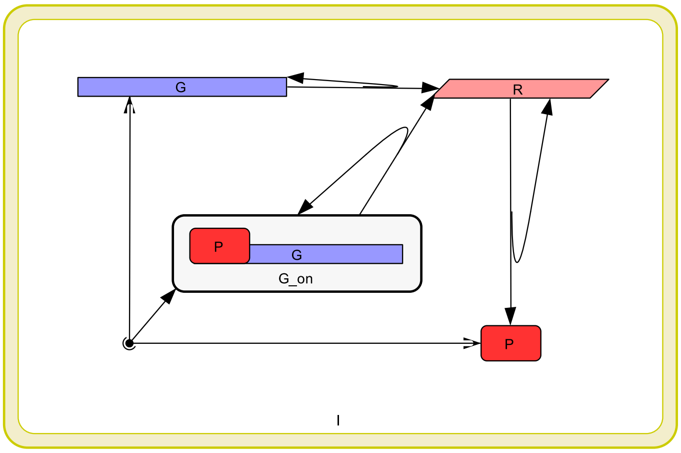

Figure 5.5.: Positive autoregulation

Figure 5.5.

Positive autoregulation (PAR) can be obtained from the *positive regulation* module in 5.7 by constraining the variables representing the activator and the protein product to be instantiated with the same objects.

$$PAR(\{G, G_{on}, R, P\}, \{c_1, \ldots, c_6\}, \{l\}) = pos(\{P, G, G_{on}, R, P\}, \{c_1, \ldots, c_6\}, \{l\}) \tag{5.10}$$

For this case following the previous analysis for the NAR and Unreg modules we compare the response time in the PAR module. Figure 5.6 shows the expected evolution of the number of proteins in the PAR module (left) and in the UnReg module (right). It can be observed that in this case opposite to negative autoregulation positive autoregulation slows down the response time.

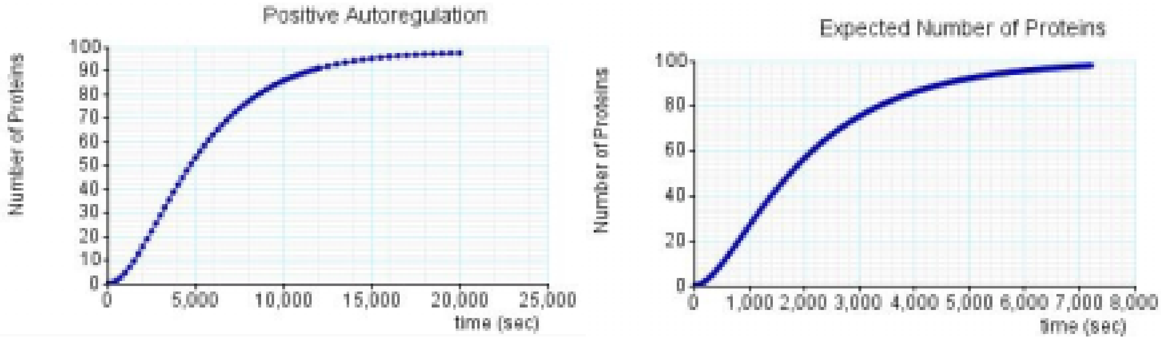

Figure 5.6.: Positive autoregulation slows down the response time

The study of more complex P systems modules representing modular patterns in cellular systems remains a promising and challenging future work open in this thesis.

# Part III.

# Case Studies

5.

# 6. Modelling Prokaryotic Gene Regulation

The structure and function of living cells is determined by the specific set of proteins they contain. These specific and characteristic proteins are codified in the cell genome. The control of gene expression determines which genes are expressed and therefore the specific proteins present in a cell. In this respect, the control of gene expression is a fundamental aspect in molecular cell biology subject to intensive research in computational systems biology.

In most bacteria or prokaryotic cells gene expression is highly regulated in order to produce the necessary proteinic machinery to respond to environmental changes. Therefore, at a given time, a bacterial cell synthesises only those proteins necessary for its survival under the particular conditions of that time. Gene expression in prokaryotes, and also in eukaryotes, is regulated primarily by mechanisms that control transcription initiation.

In this chapter we describe the basic modelling principles for prokaryotic gene regulation within the P system modelling framework. The lac operon regulation system will be used as a case study to illustrate the general principles presented in chapter 3. A brief description of the gene expression control in the lac operon is presented in section 6.1. According to this description a P system specification and a family of P system models are developed in section 6.2. Finally, in section 6.3, an analysis of the behaviour of the lac operon system under different environmental conditions is discussed.

## 6.1. Gene Expression Control in the Lac Operon

Many of the genes in Escherichia coli (E. coli) are expressed constitutively; that is, they are always turned on. Others, however, are active only when their products are needed by the cell, so their expression must be regulated. The most direct way to control the expression of a gene is to regulate its rate of transcription; that is, the rate at which RNA polymerase transcribes the gene into molecules of messenger RNA (mRNA).

Adding a new substrate to the culture medium may induce the formation of new enzymes capable of metabolising that substrate. An example of this phenomenon happens when we take a culture of E. coli that is feeding on glucose and transfer some of the cells to a medium containing lactose instead, a revealing sequence of events takes place.

At first the cells are quiescent: they do not metabolise lactose, their other metabolic activities decline, and cell division ceases. Soon, however, the culture begins growing rapidly again with the lactose being rapidly consumed. During the quiescent interval, the cells began to produce three enzymes that they had not been producing before: a permease, LacY, that transports lactose across the plasma membrane from the culture medium into the interior of the cell; $\beta$-galactosidase which hydrolyses lactose into glucose and galactose, and a transacetylase, LacA, whose function is still uncertain.

The genes encoding these proteins which are involved in the uptake and consumption of lactose are located on a region of the E. coli genome called **lac operon.**

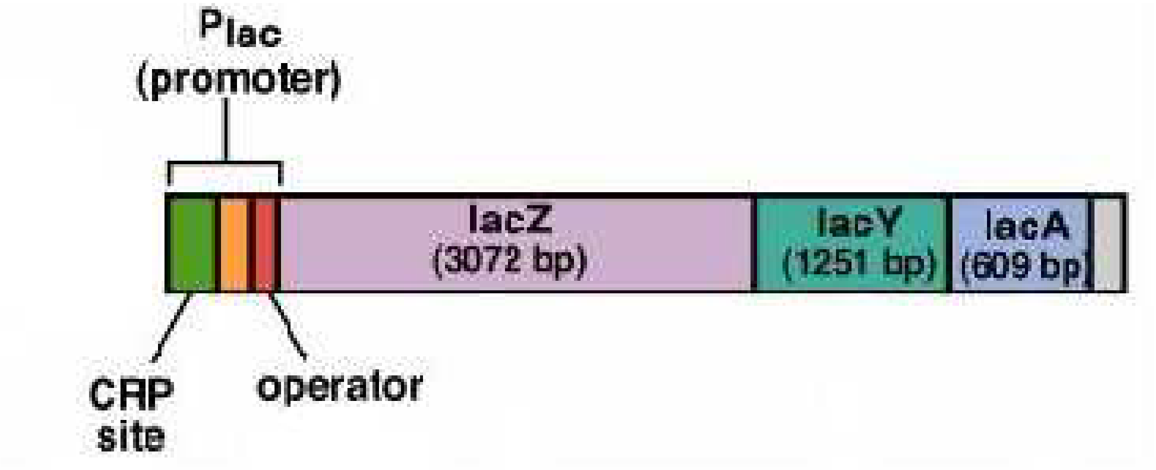

Figure 6.1.: Lac Operon

The gene lacI encodes a protein called LacI that acts as a repressor. The lac repressor is made up of four identical polypeptides (the protein product of the gene lacI). Part of this molecule has a site that enables it to recognise and bind to 24 base pairs of the lac promoter called the lac operator, preventing the RNA polymerase from transcribing the structural genes lacZ, lacY and lacA that encode $\beta$-galactosidase, the permease and the transacetylase respectively.

Nonetheless, sometimes the repressor drops from the promoter allowing transcription at a basal rate. Besides, the repressor contains another site where allolactose, a product of the reaction of lactose with $\beta$-gallactoside, can bind producing a conformational change. As a result of this change, the repressor can no longer bind to the operator region and falls off. RNA polymerase can then bind to the promoter and transcribe the lac genes.

Thus, when lactose is added to the culture medium, it causes the repressor to be released from the operator so RNA polymerase can transcribe the three structural genes of the operon into a single molecule of messenger RNA. Hardly does transcription begin before ribosomes attach to the growing mRNA molecule and move down it to translate it into the three proteins. In this respect, transcription and translation are concurrent processes in bacteria which play an important role in gene expression control.

Absence of active lac repressor is essential but not sufficient for effective transcription of the lac operon. The presence of glucose in the culture medium, even in the presence of lactose, seems to repress or inhibit the synthesis of $\beta$-galactosidase. The molecular mechanism of this effect is called **catabolite repression**. The phenomenon of catabolite repression has been observed in operons other than the lac operon; for example, the arabinose and the maltose operons.

Catabolite repression is mediated through the effects that the system of glucose transport into the cell has on the internal concentration of cyclic AMP (cAMP). If glucose is abundant in the growth medium it will be transported in to the cell by the action of the glucose transport system. During its transport to the inside of the bacterium cell glucose is phosphorylated with the phosphate group being donated by a component of the transport system called EIIA$\sim$P. The same component also activates the enzyme, adenylate cyclase (AC) which helps in the synthesis of cAMP. As long as EIIA$\sim$P is participating in glucose transport, it is not able to activate adenylate cyclase. The result is that as glucose is transported into the cell, the concentration of cAMP falls because adenylate cyclase is not being activated to synthesise cAMP any more. If there is little or no glucose in the growth medium, the glucose transport system is not operational. The phosphate donor component is now free to activate adenylate cyclase. The result is that in the absence of glucose, the concentration of cAMP rises. Thus there is an inverse relationship between the external concentration of glucose and the cytoplasmic concentration of cAMP. As one rises, the other falls. Furthermore, EIIA, the non phosphorylated state of EIIA$\sim$P, inhibits the permease involved in the uptake of lactose inside the bacterial cell preventing lactose from entering the cytoplasm.

Therefore when glucose is scarce in the medium, cAMP is abundant in the cytoplasm and it can be bound by the cAMP receptor protein (CRP), which is also known as catabolite activator protein (CAP). As its name suggests, this protein is responsible for mediating the phenomenon of catabolite repression through its ability to activate transcription. The complex CRP-cAMP$_2$ binds to the lac operon just upstream of the promoter. In this position it can assist RNA polymerase to bind by direct protein-protein contacts significantly increasing the rate of transcription by acting as an activator.

CRP, and its mechanism of action, have been the subject of intensive research over the past years since it plays a role in many other operons. However, the molecular details of how it functions are different in every case. Thus, it has been difficult to establish any unifying model for the mechanism of action of this protein.

Summing up, the lactose operon is subject to both negative and positive control. The lac repressor, LacI, negatively regulates expression and, the activator, cAMP-CRP$_2$, positively activates expression.

There are, as a result, four basic states of expression of the lac operon:

- **NO** Glucose and **NO** Lactose

  Under these conditions, there will be a large number of cAMP molecules in the cell and CRP-cAMP$_2$ will be bound at its binding site upstream of the lac promoter. It will assist RNA polymerase to bind to the promoter but it will not activate transcription because the lactose repressor will remain bound to the operator site since there is no inducer, allolactose, present.

  There will essentially be no transcription of the lac operon.

  This makes physiological sense. Without sugar substrates the cell cannot carry out much metabolism; however, it remains poised to use whatever it can whenever it can. In this case, if lactose does become available, the cell can and will immediately respond because lactose permease will transport the lactose into the cell and RNA polymerase is positioned to start the expression of $\beta$-galactosidase so that the lactose can be utilised immediately.

- Glucose present but **NO** Lactose

  Under these conditions, there will be a low number of cAMP molecules in the cell so CRP-cAMP$_2$ will not be bound at the lac promoter. In addition, the activity of lactose permease will be inhibited by EIIA.

  There will be no transcription of the lac operon.

  This also makes physiological sense. As long as glucose is present in the growth medium there is little need to metabolise lactose and since lactose is not present there is no need to transport lactose into the cell or to express the genes of the lac operon.

- Glucose and lactose present

  Under these conditions, there will be a low number of cAMP molecules in the cell so CRP-cAMP$_2$ will not be bound at the lac promoter. Lactose permease will be inhibited by EIIA nevertheless some lactose will still enter the cell producing some inducer molecules of allolactose. These molecules in turn will deactivate some LacI molecules decreasing the level of repression of the lac operon.

  There will be a low level of transcription of the lac operon.

  Again, this makes physiological sense. As long as glucose is present in the growth medium there is little need to metabolise lactose. However, since lactose is now present, it would be inefficient to ignore a sugar supply completely. The lac operon will be induced but, since CRP is not bound, the amount of transcription is relatively low.

- **NO** Glucose but abundant lactose

  Under these conditions, there will be a large number of cAMP molecules in the cell so CRP-cAMP$_2$ will be bound at the lac promoter. Lactose permease is not inhibited, so it will transport the lactose into the cell. Once in the cytoplasm lactose will interact with $\beta$-galactosidase to produce allolactose which in turn binds to the repressor LacI preventing it from inhibiting the expression of lac operon.

  There will be maximal transcription of the lac operon.

  This also makes physiological sense. With lactose as the sole sugar source, the cell must use every available molecule for its own benefit. Thus the lactose permease transport system will bring lactose into the cell and the lac operon will be both induced and activated.

The presence of two separate control systems allows the cell to respond more sensitively to the needs imposed by changing growth conditions. Many bacterial operons have dual control systems, nevertheless the details are different in the different cases [96].

## 6.2. Modelling the Lac Operon Regulation Using P systems

In this section we develop a P system specification and a family of P system models for the transcription regulation system in the lac operon. Our models will illustrate the specification principles discussed in Chapter 3. More specifically we will use objects to represent molecular entities like proteins and sugars; strings will specify molecular entities with a linear structure like the lac operon and mRNAs; and rewriting rules on multisets of objects and strings will describe the interactions between the different molecular components of the system. Finally, the relevant regions in the transcription regulation system are described using membranes delimiting compartments.

In the models proposed in this section we will study the behaviour of system for different initial conditions in the culture medium. Namely, a culture medium with/without glucose and with/without lactose.

### 6.2.1. A P system Specification of the Lac Operon System

In what follows we present a detailed description of the P system specification of the lac operon regulation system in this work.

- **Specification of the relevant regions in the lac operon system:**

  Membranes play a key role in the functioning and structural organisation of the gene regulation system in the lac operon. E. coli has been the subject of intensive research which has produced vast knowledge of the functionality of its cellular systems. E. coli presents a relatively simple structure consisting of a single compartment surrounded by a plasma membrane. The function of the plasma membrane is to define the bacterium itself by isolating and differentiating it from the surrounding medium. Nevertheless, its function is not only limited to enclose specific molecules interacting through particular cellular processes. The plasma membrane also controls the movement and transport of molecules between the inside and the outside of the bacterium. Another important mechanism where the plasma membrane plays a key role is cell signalling. These two processes, namely, selective transport of molecules and cell signalling at the cell surface, are crucial within the lac operon system. In this respect, in an E. coli bacterium, there are two distinct and relevant regions:

  - The *bacterium cell surface* which contains a set of proteins that control the movement of molecules and detect signals.
  - The *bacterium lumen* or aqueous interior of the bacterium where a characteristic complement of proteins are involved in specific cellular processes.

  According to the specification principles presented in Chapter 3, regarding the use of membranes to define relevant regions in cellular system, two different membranes will be used in this work to specify an E. coli bacterium.

1. The first membrane, identified with the label $s$, represents the bacterium cell surface. This membrane will be used to define the bacterium itself. The objects describing the molecular entities associated in some manner with the plasma membrane will be located in the region defined by this membrane. The rewriting rules specifying the processes of selective uptake of substances from the environment and cell signalling at the bacterium cell surface will also be associated with this region.
2. The second membrane, identified with the label $c$, will describe the aqueous interior of the bacterium and therefore is embedded inside the previous membrane. The set of objects and strings specifying the proteins and other molecular entities, like the lac operon itself, which are located in the lumen of the bacterium, will be placed in the region defined by this membrane. The rewriting rules describing the molecular interactions taking place inside the bacterium are also associated with the region or compartment defined by this membrane.
3. There exists another relevant region in the lac operon system, the culture medium, which must be taken into account if one intends to produce a complete model of the lac operon regulation system. Another membrane, identified with the label $e$, will be used to specify this region. Depending on the sugar sources, glucose and/or lactose, present in the culture medium an E. coli bacterium will exhibit different behaviour. In this work the different responses of an E. coli bacterium to different sugar sources will be obtained by placing the objects corresponding to these sugar sources in the region defined by this last membrane. Finally, note that this membrane will determine the boundary of our system and it contains the membrane representing the bacterium cell surface which in turn contains the membrane describing the bacterium interior. In Figure 6.2, a Venn diagram representation of this membrane structure is depicted.

- **Specification of the molecular entities of the lac operon system:**

  Among the molecular entities involved in the lac operon system there are two significantly different sets of molecules that differ in their structure:

  - On the one hand, we have the proteins and enzymes involved in the selective uptake of sugars from the culture medium and in their metabolism, for instance EIICB and the enzyme $\beta$–galactosidase. Although the structure of these molecules is complex and determines their functionality, an explicit

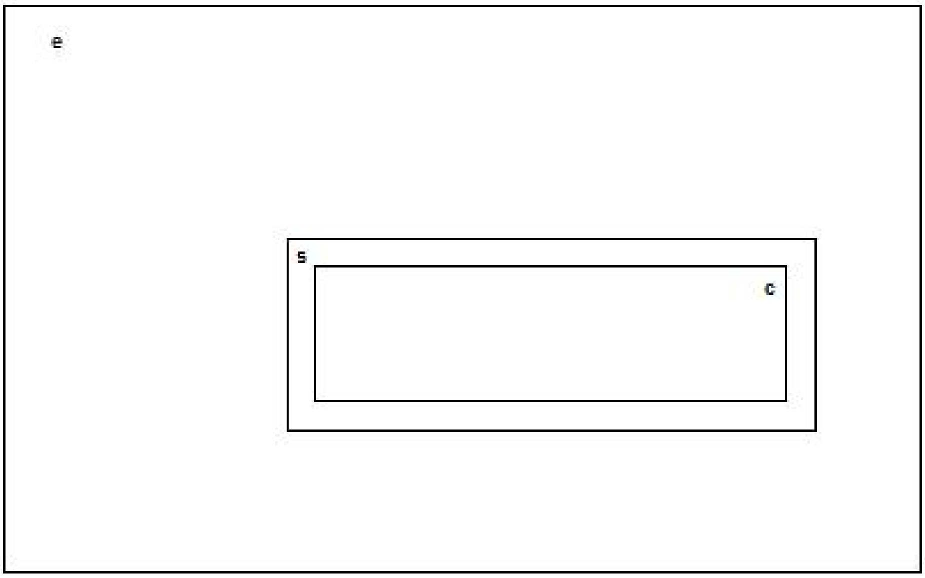

Figure 6.2.: The membrane structure depicting the relevant regions in the lac operon system.

specification of their structure is not crucial. Therefore, these molecules can be thought of as individual entities without an internal structure. In this respect, they will be represented using objects, and the molecular population they constitute will be described using multisets of objects.

The objects used in this work to represent the proteins and enzymes involved in the lac operon are collected in the alphabet $O_{prot}$. Figure 6.3 shows the correspondence between the objects from $O_{prot}$ and the proteins, enzymes and molecular complexes involved in the lac operon system.

$O_{prot} = \{$ EIIA, EIIA∼P, EIICB, EIICB–EIIA∼P, EIICB∼P, Gluc, Gluc∼P, EIICB∼P–Gluc, Lact, LacY, Lact-LacY, AC, AC–EIIA, AC–EIIA∼P, ATP, AC–EIIA∼P–ATP, LacY-EIIA, $\beta$–Galac, Lact, $\beta$–Galac–Lact, Allolact, LacI, LacI–Alloct, CRP, CRP–cAMP, CRP–cAMP$_2$, RNAP $\}$

The functions carried out by these proteins will be described later when discussing the specification of the cellular processes in the lac operon regulation system using rewriting rules.

– On the other hand, we have the genes which codify the necessary information for the synthesis of some of the proteins and enzymes described in the

| **Symbol** | **Molecule** |
|---|---|
| EIICB<br>EIICB$\sim$P | Glucose Transporter Enzyme IICB<br>and its phosphorylated state |
| EIIA<br>EIIA$\sim$P | Glucose Transporter Enzyme IIA<br>and its phosphorylated state |
| EIICB–EIIA$\sim$P | Complex transporter enzymes IICB and phophorilated IIA |
| Gluc Gluc$\sim$P | Glucose and its phosphorylated state |
| EIICB$\sim$P–Gluc | Complex transporter IICB phosphorylated and glucose |
| Lact | Lactose |
| LacY | Lactose permease |
| Lact-LacY | Complex lactose permease and lactose |
| LacY-EIIA | Complex lactose permease and transporter enzyme IIA |
| ATP<br>cAMP | Adenosine triphosphate and<br>cyclic adenosine monophosphate |
| AC | Adelynate Cyclase |
| AC–EIIA<br>AC–EIIA$\sim$P<br>AC–EIIA$\sim$P–ATP | ATP and AC complexes with and ATP |
| $\beta$–Galac<br>$\beta$–Galac–Lact | $\beta$–Galactosidase and its complex with lactose |
| Allolact | Allolactose, the inducer |
| LacI LacI–Alloct | The lac repressor and its complex with allolactose |
| CRP<br>CRP–cAMP<br>CRP–cAMP$_2$ | cAMP Receptor Protein, the activator,<br>its complex with cAMP and its dimer |
| RNAP | RNA polymerase |

Figure 6.3.: Objects representing the proteins, enzymes and other molecules involved in the lac operon regulation system.

alphabet $O_{prot}$. The genes studied in this work are arranged together in a linear structure called the lac operon. The order in which these genes appear in the lac operon is relevant as it determines the order at which the genes are expressed making their corresponding protein products available to perform specific tasks. For instance, the first gene in the lac operon is the LacZ gene which codifies the enzyme $\beta$-galactosidase. This enzyme cleaves lactose into glucose and galactose producing as a byproduct allolactose, the inducer of the lac operon system. Since this constitutes the first step in the consumption of lactose and in the production of the signal marking the presence of lactose in the environment evolution has located the gene LacZ in first place. The other linear structures in the lac operon system are the mRNAs.

The linear structure of these molecular entities, genes and mRNAs, will be specified in this work using strings. The relevant sites in the lac operon and in the corresponding mRNA will be represented using symbols from the alphabets $O_{dna}$ and $O_{rna}$ respectively. A detailed description of the sites represented by the symbols from $O_{dna}$ and $O_{rna}$ is presented in Figure 6.4.

$$O_{dna} = \{\ \text{cap},\ \text{cap}^{CRP-cAMP_2},\ \text{op},\ \text{op}^{LacI}, \text{lacZ}_s, \text{lacZ}_m, \text{lacZ}_e, \text{lacY}_s, \text{lacY}_m, \text{lacY}_e, \text{lacA}_s, \text{lacA}_m, \text{lacA}_e\}$$

$$O_{rna} = \{\text{Rib}, \overline{\text{op}}, \overline{\text{lacZ}}_s, \overline{\text{lacZ}}_m, \overline{\text{lacZ}}_e, \overline{\text{lacY}}_s, \overline{\text{lacY}}_m, \overline{\text{lacY}}_e, \overline{\text{lacA}}_s, \overline{\text{lacA}}_m, \overline{\text{lacA}}_e\}$$

| **Symbol** | **Site** |
|---|---|
| cap | Free CAP site where the activator CRP-cAMP$_2$ binds |
| $\text{cap}^{CRP-cAMP_2}$ | CAP site occupied by the activator CRP-cAMP$_2$ binds |
| op | Free operator site where the repressor LacI binds |
| $\text{op}^{LacI}$ | Operator site occupied by the repressor LacI binds |
| $\text{lacZ}_s$<br>$\text{lacY}_s$<br>$\text{lacA}_s$ | Sites marking the start point of the lacZ, lacY and lacA gene respectively |
| $\text{lacZ}_m$<br>$\text{lacY}_m$<br>$\text{lacA}_m$ | Sites located in the middle of the lacZ, lacY and lacA gene respectively |
| $\text{lacZ}_e$<br>$\text{lacY}_e$ | Sites located in the final point of the lacZ and lacY gene respectively. |
| $\text{lacA}_e$ | Site marking the end of gene lacA which coincides with the transcription termination site of the lac operon |
| $\overline{\text{op}}$ | Site marking the starting point of the mRNA transcript |
| $\overline{\text{lacZ}}_s$ $\overline{\text{lacZ}}_e$<br>$\overline{\text{lacY}}_s$ $\overline{\text{lacY}}_e$<br>$\overline{\text{lacA}}_s$ $\overline{\text{lacA}}_e$ | Sites marking the beginning and end of the reading frames in the mRNA for the genes lacZ, lacY and lacA respectively |
| $\overline{\text{lacZ}}_m$<br>$\overline{\text{lacY}}_m$<br>$\overline{\text{lacA}}_m$ | Sites located in the middle of reading frames of lacZ, lacY and lacA gene respectively |

Figure 6.4.: Symbols specifying the relevant sites in the lac operon and in its corresponding mRNA.

More specifically, the following string, $s_3$, represents the lac operon in E. coli:

$$\langle\ \text{cap.}\ \text{op}\ .\ \overbrace{\text{lacZ}_s.\text{lacZ}_m.\cdots\ .\text{lacZ}_m.\text{lacZ}_e}^{30}\ \overbrace{.\text{lacY}_s.\text{lacY}_m.\cdots\ .\text{lacY}_m.\text{lacY}_e}^{12}\ .$$

$$\overbrace{\text{lacA}_s.\text{lacA}_m.\cdots\ .\text{lacA}_m.\text{lacA}_e}^{6}\ \rangle$$

Due to its size RNA polymerase occupies around 100 nucleotides, and so each symbol, $\langle$ lacZ$_i$ $\rangle$, $\langle$ lacY$_i$ $\rangle$ and $\langle$ lacA$_i$ $\rangle$ with $i = s, m, e$, represents a sequence of 100 nucleotides of the corresponding gene, instead of a single one. Therefore we have only 30 sites lacZ, 12 lacY and 6 lacA representing the 3000, 1200 and 600 nucleotides of the corresponding genes.

The CAP binding site and the operator are represented by $\langle$ cap $\rangle$ and $\langle$ op $\rangle$.

- **Specification of the molecular interactions in the lac system:**

As discussed in Chapter 3 rewriting rules on multisets of objects and strings are used within the P systems modelling framework to describe molecular interactions in living cells. More specifically, rewriting rules on multisets of objects are generally used for the specification of protein-protein interactions whereas rewriting rules on multisets of objects and strings are applied when describing processes involving linear structures as in the case of gene expression control.

In what follows, a detailed description of the rewriting rules describing the molecular processes in the lac operon system is given. The stochastic constants associated with each rule will also be presented.

– Rules describing the glucose transport system.

The uptake of glucose consists of a transfer pathway where a phosphoryl group is transferred sequentially along a series of proteins to glucose. This process will be described using the P system specification schema for recruitment and releasing (4.5) and for binding and debinding (4.4).

The uptake glucose from the environment starts with the recruitment of the cytoplasmic phosphorylated protein EIIA∼P by the transmembrane protein EIICB. This is describe by the recruitment rule $r_1$. This process is reversible, EIIA∼P can be released back unchanged to the cytoplasm, releasing rule $r_2$.

$$r_1\text{: EIICB [ EIIA}\sim\text{P }]_c \xrightarrow{c_1}\text{EIICB–EIIA}\sim\text{P [ ]}_c,\ c_1 = 0.235\, molec^{-1} sec^{-1}$$

$$r_2\text{: EIICB–EIIA}\sim\text{P [ ]}_c \xrightarrow{c_2} \text{EIICB [ EIIA}\sim\text{P }]_c,\ c_2 = 2 \times 10^{-4}\, sec^{-1}$$

Once EIIA∼P is recruited by EIICB they can interact producing the phosphorylation of EIICB in the cell surface and the release of EIIA to the cytoplasm. The releasing rule $r_3$ describes this interaction.

$r_3$: EIICB–EIIA∼P [ $]_c \overset{c_3}{\rightarrow}$ EIICB∼P [ EIIA $]_c$, $c_3 = 0.0706\,sec^{-1}$

Once the phosphorylated protein EIICB∼P is present on the cell surface glucose can start to be transported inside the cell by binding to EIICB∼P, binding rule $r_4$. On the one hand, this interaction is reversible and it can result in the debinding of the glucose from EIICB∼P, debinding rule $r_5$. On the other hand, glucose can be transported inside the bacterial cell and released into the cytoplasm, releasing rule $r_6$. In this process glucose is phosphorylated, Gluc∼P.

$r_4$: Gluc [ EIICB∼P $]_s \overset{c_4}{\rightarrow}$ [ EIICB∼P–Gluc $]_s$, $c_4 = 6.96 \times 10^{-3}\,molec^{-1}sec^{-1}$

$r_5$: [ EIICB∼P–Gluc $]_s \overset{c_{18}}{\rightarrow}$ Gluc [ EIICB∼P $]_s$, $c_5 = 1.04 \times 10^{-2}\,sec^{-1}$

$r_6$: EIICB∼P–Gluc [ $]_c \overset{c_6}{\rightarrow}$ EIICB [ Gluc∼P $]_c$, $c_6 = 0.128\,sec^{-1}$

− Rules describing the lactose transport system.

The transport of lactose inside the bacterial cell is specified with the same type of rules as the glucose transport system, namely binding, debinding, recruitment and releasing rules. However, the lactose transport system is simpler as it does not imply a sequential transfer of a phosphoryl group, as it is the case in the glucose transport system.

For lactose to enter the bacterial cell the presence of the permease LacY on the cell surface is necessary. LacY is the protein product of the second gene in the lac operon, lacY. Although, LacY is synthesised in the cytoplasm it moves to the cell surface. The transport of LacY to the cell surface is specified using the diffusion out rule, $r_7$.

$r_7$: [ LacY $]_c \overset{c_7}{\rightarrow}$ LacY [ $]_c$, $c_7 = 0.02 sec^{-1}$

The uptake of lactose, Lact, present in the environment is carried out in two steps. First, lactose binds to the permease, binding rule $r_8$. Once bound to the permease it is transported and released into the cytoplasm, releasing rule $r_9$.

$r_8$: Lact [ LacY $]_s \overset{c_8}{\rightarrow}$ [ Lact-LacY $]_s$, $c_8 = 5.12 \times 10^{-3} molec^{-1}sec^{-1}$

$r_9$: Lact-LacY [ $]_c \overset{c_9}{\rightarrow}$ LacY [ Lact $]_c$, $c_9 = 5.12 \times 10^{-3} sec^{-1}$

The uptake of lactose is inhibited by EIIA through direct interaction. EIIA binds to the permease LacY producing the complex LacY-EIIA, recruitment rule $r_{10}$. Lactose can not bind to this complex on the cell surface blocking the transport of lactose into the bacterial cell. The complex LacY-EIIA can dissociate back into its components, releasing rule $r_{11}$.

$r_{10}$: LacY [ EIIA $]_c \overset{c_{10}}{\rightarrow}$ LacY-EIIA [ $]_c$, $c_{10} = 10^{-4}\, molec^{-1}sec^{-1}$

$r_{11}$: LacY-EIIA [ $]_c \overset{c_{11}}{\rightarrow}$ LacY [ EIIA $]_c$ $c_{11} = 10^{-3}\, sec^{-1}$

These molecular interactions are important in catabolite repression; a high number of non-phosphorylated EIIA in the cytoplasm is a consequence of the presence of glucose in the environment. During its transport glucose is phosphorylated with a phosphoryl group that is transferred from EIIA∼P to EIIBC and finally to the glucose. Therefore, in the presence of glucose, EIIA∼P is depleted from the cytoplasm and EIIA is produces in high numbers. This situation inhibits the permease LacY and the uptake of lactose.

– Rules describing the activity of $\beta$–Galactosidase.

$\beta$–galactosidase cleaves lactose into glucose and galactose so the bacterium can consume it. The production of glucose from lactose is not crucial in the regulation of the expression of the genes in the lac operon and it will not be specified in our work. Nevertheless, the production of allolactose is a key step in the regulation of the system as allolactose acts as an inducer. Allolactose is a product of the interaction between $\beta$–galactosidase and lactose. $\beta$–galactosidase is an enzyme which first forms a complex with lactose, complex formation rule $r_{12}$, and then interacts with it to produce allolactose, dissociation rule $r_{13}$.

$r_{12}$: [ $\beta$–Galac + Lact $]_c \overset{c_{12}}{\rightarrow}$ [ $\beta$–Galac–Lact $]_c$, $c_{12} = 3.92 \times 10^{-4}\, molec^{-1}sec^{-1}$

$r_{13}$: [ $\beta$–Galac–Lact $]_c \overset{c_{13}}{\rightarrow}$ [ $\beta$–Galac + Allolact $]_c$, $c_{13} = 3.92{\times}10^{-4}\, molec^{-1}sec^{-1}$

$\beta$–Galactosidase is degraded by the cell machinery in the cytoplasm. This process is described by the degradation rule, $r_{14}$.

$r_{14}$: [ $\beta$–Galac $]_c \overset{c_{14}}{\rightarrow}$ [ $]_c$, $c_{14} = 1.93 \times 10^{-4}\, sec^{-1}$

– Rules describing the activity of allolactose.

Allolactose is a signal of the presence of lactose in the environment. It acts as an inducer of the regulation system in the lac operon by inhibiting the repressor, LacI, through direct interaction, complex formation rule $r_{15}$. A conformational change is induced in the repressor when allolactose binds to it. This prevents the repressor from binding to the operator of the lac operon.

Allolactose is also degraded in the cytoplasm, degradation rule $r_{16}$.

$r_{15}$: $[\text{ LacI + Allolact }]_c \overset{c_{15}}{\rightarrow} [\text{ LacI-Allolact }]_c,\ c_{15} = 0.01 molec^{-1} sec^{-1}$

$r_{16}$: $[\text{ Allolact }]_c \overset{c_{16}}{\rightarrow} [\ ]_c,\ c_{16} = 5.58 \times 10^{-5} sec^{-1}$

– Rules describing the activity of Adenylate Cyclase.

Adenylate Cyclase (AC) regulates the production of cAMP molecules, whose number is inversely proportional to the number of glucose molecules in the environment. For the synthesis of cAMP, AC recruits reversibly the phosphorylated EIIA~P from the cytoplasm, recruitment rule $r_{17}$ and releasing rule $r_{18}$.

$r_{17}$: $\text{AC } [\text{ EIIA}{\sim}\text{P }]_c \overset{c_{17}}{\rightarrow} \text{AC–EIIA}{\sim}\text{P } [\ ]_c,\ c_{17} = 2.35 \times 10^{-5} molec^{-1} sec^{-1}$

$r_{18}$: $\text{AC–EIIA}{\sim}\text{P } [\ ]_c \overset{c_{18}}{\rightarrow} \text{AC } [\text{ EIIA}{\sim}\text{P }]_c,\ c_{18} = 0.01 sec^{-1}$

The complex AC–EIIA~P recruits ATP from the cytoplasm, recruitment rule, $r_{19}$, and transforms it into cAMP which subsequently is released into the cytoplasm, releasing rule $r_{20}$.

$r_{19}$: $\text{AC–EIIA}{\sim}\text{P } [\text{ ATP }]_c \overset{c_{19}}{\rightarrow} \text{AC}{\sim}\text{P–EIIA}{\sim}\text{P–ATP } [\ ]_c,$
$c_{19} = 2.35 \times 10^{-3} molec^{-1} sec^{-1}$

$r_{20}$: $\text{AC–EIIA}{\sim}\text{P–ATP } [\ ]_c \overset{c_{20}}{\rightarrow} \text{AC}{\sim}\text{P–EIIA}{\sim}\text{P } [\text{ cAMP }]_c,\ c_{20} = 0.02 sec^{-1}$

As mentioned before the non-phosphorylated number of molecules of EIIA is a signal of the presence of glucose in the environment. As part of the mechanism of catabolite repression, EIIA inhibits the production of cAMP by binding reversibly to AC in the cell membrane, recruitment rule $r_{21}$ and releasing rule $r_{22}$. This inhibits the activation of the lac operon by repressing the activity of AC and therefore the production of cAMP and the activator CRP-cAMP$_2$.

$r_{21}$: $\text{AC } [\text{ EIIA }]_c \overset{c_{21}}{\rightarrow} \text{AC–EIIA } [\ ]_c\, c_{21} = 2.35 \times 10^{-3} molec^{-1} sec^{-1}$

$r_{22}$: $\text{AC–EIIA } [\ ]_c \overset{c_{22}}{\rightarrow} \text{AC } [\text{ EIIA }]_c,\ c_{22} = 0.02 sec^{-1}$

– Rules describing the formation of the activator CRP-cAMP$_2$

The protein CRP can not bind to the promoter, unless it interacts with cAMP to produce the complex CRP-cAMP which in turn will form a dimer CRP-cAMP$_2$, complex formation rules $r_{23}$ and $r_{24}$. This last dimer is able to bind to the promoter and increase the rate of transcription of the genes encoded in the lac operon.

$r_{23}$: $[\ \text{CRP} + \text{cAMP}\ ]_c \overset{c_{23}}{\rightarrow} [\ \text{CRP-cAMP}\ ]_c$, $c_{23} = 5 \times 10^{-3} molec^{-1} sec^{-1}$

$r_{24}$: $[\ \text{CRP-cAMP} + \text{CRP-cAMP}\ ]_c \overset{c_{24}}{\rightarrow} [\ \text{CRP-cAMP}_2\ ]_c$,
$c_{24} = 5 \times 10^{-3} molec^{-1} sec^{-1}$

– Rules describing the activation and repression of the lac operon

The rate of transcription of the lac operon is determined by the state of the so called *lac operon switch*. The lac operon switch is made of the CAP site, $\langle \text{cap} \rangle$, where the activator binds and the operator, $\langle \text{op} \rangle$, where the repressor binds. The lac operon switch has four different configurations depending on the occupation of the CAP site and operator: $\langle \text{cap.op} \rangle$, $\langle \text{cap.op}^{LacI} \rangle$, $\langle \text{cap}^{CRP-cAMP_2}.\text{op} \rangle$ and $\langle \text{cap}^{CRP-cAMP_2}.\text{op}^{LacI} \rangle$. These configurations can be reached by applying the binding and debinding of transcription factor rules introduced in (4.9).

In the absence of lactose the repressor LacI will be active and it will bind the operator inhibiting the transcription of the lac operon, transcription factor binding rule $r_{25}$. Nonetheless, occasionally LacI drops from the operator allowing a basal production of the proteins, transcription factor dissociation rule $r_{26}$.

$r_{25}$: $[\ \text{LacI} + \langle\ \text{op}\ \rangle\ ]_c \overset{c_{25}}{\rightarrow} [\ \langle\ \text{op}^{LacI} \rangle\ ]_c$, $c_{25} = 0.2 molec^{-1} sec^{-1}$

$r_{26}$: $[\ \langle\ \text{op}^{LacI} \rangle\ ]_c \overset{c_{26}}{\rightarrow} [\ \text{LacI} + \langle\ \text{op}\ \rangle\ ]_c$, $c_{26} = 5 sec^{-1}$

In absence of glucose there is a high number of cAMP molecules that will produce a high number of activators, CRP-cAMP$_2$. The activator binds reversibly to the CAP site $\langle$ cap $\rangle$, transcription factor binding and debinding rules $r_{27}$ and $r_{28}$ respectively.

$r_{27}$: $[\ \text{CRP-cAMP}_2 + \langle\ \text{cap}\ \rangle\ ]_c \overset{c_{27}}{\rightarrow} [\ \langle\ \text{cap}^{CRP-cAMP_2}\ \rangle\ ]_c$,
$c_{27} = 0.01 molec^{-1} sec^{-1}$

$r_{28}$: $[\ \langle\ \text{cap}^{CRP-cAMP_2}\ \rangle\ ]_c \overset{c_{28}}{\rightarrow} [\ \text{CRP-cAMP}_2 + \langle\ \text{cap}\ \rangle\ ]_c$, $c_{28} = 5 sec^{-1}$

– Rules describing transcription initiation in the lac operon

The first step in the transcription of the lac operon consists in the binding of the RNA polymerase (RNAP) to the lac operon switch. The affinity between the lac operon switch and the RNA polymerase depends on the configuration of the switch. On the one hand, when the CAP site is free, $\langle$ cap $\rangle$, the RNAP seldomly binds to the lac operon switch producing a basal rate of transcription, rule $r_{29}$. On the other hand, when the activator CRP-cAMP$_2$ occupies the CAP site, $\langle$ cap$^{CRP-cAMP_2}$ $\rangle$, it produces an increase in the rate of transcription of around 40-fold, rule $r_{30}$.

$r_{29}$: $[\ \text{RNAP} + \langle\ \text{cap}\ \rangle\ ]_c \overset{c_{29}}{\rightarrow} [\ \langle\ \text{cap.RNAP}\ \rangle\ ]_c$, $c_{29} = 5 \times 10^{-4} molec^{-1}sec^{-1}$

$r_{30}$: $[\ \text{RNAP} + \langle\ \text{cap}^{CRP-cAMP_2}\ \rangle\ ]_c \overset{c_{30}}{\rightarrow} [\ \langle\ \text{cap}^{CRP-cAMP_2}.\ \text{RNAP}\ \rangle\ ]_c$,
$c_{30} = 0.02 molec^{-1}sec^{-1}$

RNAP starts transcription by producing the complementary ribonucleotides of the operator site $\langle\ \overline{\text{op}}\ \rangle$, transcription initiation rule $r_{31}$. The site $\langle\ \overline{\text{op}}\ \rangle$ marks the beginning of a mRNA transcript. Note that after the application of rule $r_{31}$ the substring $\langle$ op $\rangle$ is left free so another RNAP can start transcription before the first RNAP finishes transcribing the operon. Therefore we are explicitly simulating transcription by different polymerase as a concurrent process.

$r_{31}$: $[\ \langle\ \text{RNAP.op}\ \rangle\ ]_c \overset{c_{31}}{\rightarrow} [\ \langle\ \text{op.}\ \overline{\text{op}}.\ \text{RNAP}\ \rangle\ ]_c$, $c_{31} = 2sec^{-1}$

– Rules describing mRNA elongation in the lac operon

During the first stages of mRNA elongation RNAP moves along the lacZ gene transcribing it into mRNA. RNAP starts transcribing the first nucleotides of the lacZ gene represented by $\langle$ lacZ$_s$ $\rangle$ and attaches the complementary ribonucleotides specified by $\langle\ \overline{\text{lacZ}_s}\ \rangle$ to the growing mRNA, $\langle\ \overline{\text{op}}\ \rangle$, transcription elongation rule $r_{32}$. The substring $\langle\ \overline{\text{lacZ}_s}\ \rangle$ represents the RBS (ribosome binding site) for this gene. Once this site is produced a ribosome binding rule can be applied and translation can start before translation is over. Therefore in our approach transcription and translation take place in parallel.

$r_{32}$: $[\ \langle\ \overline{\text{op}}.\ \text{RNAP.}\ \text{lacZ}_s\ \rangle\ ]_c \overset{c_{32}}{\rightarrow} [\ \langle\ \text{lacZ}_s.\overline{\text{op}}.\overline{\text{lacZ}_s}.\text{RNAP}\ \rangle\ ]_c$, $c_{32} = 2sec^{-1}$

During transcription a complementary strand of mRNA is produced. This is described by the production of the RNA site $\langle\ \overline{\text{lacZ}_m}\ \rangle$ which is attached to the growing mRNA represented by the substring $\langle\ \overline{\text{op}}.w\ \rangle$, $w \in O^*_{rna}$ , whereas the DNA just transcribed $\langle$ lacZ$_m$ $\rangle$ is left behind, transcription elongation rule $r_{33}$.

$r_{33}$: $[\ \langle\ \overline{\text{op}}.w.\text{RNAP.lacZ}_m\ \rangle\ ]_c \overset{c_{33}}{\rightarrow} [\ \langle\ \text{lacZ}_m.\overline{\text{op}}.w.\overline{\text{lacZ}_m}.\text{RNAP}\ \rangle\ ]_c$,
$c_{33} = 2sec^{-1}$

When the RNAP reaches the end of the LacZ gene it attaches the site $\langle\ \overline{\text{lacZ}_e}\ \rangle$ to the substring $\langle\ \overline{\text{op}}.w\ \rangle$, $w \in O^*_{rna}$ , which describes the growing mRNA, transcription elongation rule $r_{34}$. The string $\langle\ \overline{\text{lacZ}_e}\ \rangle$ represents a translation termination site, therefore when ribosomes reach this site they dissociate releasing the protein encoded by the lacZ gene, $\beta$-galactosidase.

$r_{34}$: $[\ \langle\ \overline{\text{op}}.\text{w.RNAP.lacZ}_e\ \rangle\ ]_c \overset{c_{34}}{\rightarrow} [\ \langle\ \text{lacZ}_e.\overline{\text{op}}.\text{w}.\overline{\text{lacZ}_e}.\text{RNAP}\ \rangle\ ]_c$,
$c_{34} = 2sec^{-1}$

The following transcription elongation rules describe the transcription of the genes lacY and lacA in a similar way as in the case of the gene lacZ.

$r_{35}$: $[\ \langle\ \overline{\text{op}}.w.\ \text{RNAP. lacY}_s\ \rangle\ ]_c \overset{c_{35}}{\rightarrow} [\ \langle\ \text{lacY}_s.\overline{\text{op}}.w.\overline{\text{lacY}_s}.\text{RNAP}\ \rangle\ ]_c$,
$c_{35} = 2sec^{-1}$

$r_{36}$: $[\ \langle\ \overline{\text{op}}.w.\ \text{RNAP. lacY}_m\ \rangle\ ]_c \overset{c_{36}}{\rightarrow} [\ \langle\ \text{lacY}_m.\overline{\text{op}}.w.\overline{\text{lacY}_m}.\text{RNAP}\ \rangle\ ]_c$,
$c_{36} = 2sec^{-1}$

$r_{37}$: $[\ \langle\ \overline{\text{op}}.w.\ \text{RNAP. lacY}_e\ \rangle\ ]_c \overset{c_{37}}{\rightarrow} [\ \langle\ \text{lacY}_e.\overline{\text{op}}.w.\overline{\text{lacY}_e}.\text{RNAP}\ \rangle\ ]_c$,
$c_{37} = 2sec^{-1}$

$r_{38}$: $[\ \langle\ \overline{\text{op}}.w.\ \text{RNAP. lacA}_s\ \rangle\ ]_c \overset{c_{38}}{\rightarrow} [\ \langle\ \text{lacA}_s.\overline{\text{op}}.w.\overline{\text{lacA}_s}.\text{RNAP}\ \rangle\ ]_c$,
$c_{38} = 2sec^{-1}$

$r_{39}$: $[\ \langle\ \overline{\text{op}}.w.\ \text{RNAP. lacA}_m\ \rangle\ ]_c \overset{c_{39}}{\rightarrow} [\ \langle\ \text{lacA}_m.\overline{\text{op}}.w.\overline{\text{lacA}_m}.\text{RNAP}\ \rangle\ ]_c$,
$c_{39} = 2sec^{-1}$

– Rule describing transcription termination in the lac operon

The transcription of the lac operon terminates when the RNA polymerase reaches a transcription termination site at the end of the lacA gene. This site is represented by the string $\langle\ \text{lacA}_e\ \rangle$. When the RNAP reaches this termination site it attaches the final ribonucleotides, $\langle\ \overline{\text{lacA}_e}\ \rangle$ to the growing mRNA $\langle\ \overline{\text{op}}.w.\ \rangle$, $w \in O^*_{rna}$, and it dissociates from the operon releasing the fully transcribed mRNA, transcription termination rule $r_{40}$.

$r_{40}$: $[\ \langle\ \overline{\text{op}}.w.\text{RNAP.lacA}_e\ \rangle\ ]_c \overset{c_{40}}{\rightarrow} [\ \text{RNAP} + \langle\ \text{lacA}_e\ \rangle\ ;\ \langle\ \overline{\text{op}}.w.\overline{\text{lacA}_e}\ \rangle\ ]_c$,
$c_{40} = 2sec^{-1}$

– Rules describing translation initiation

Translation starts when ribosomes, Rib, recognise the RBS of the genes lacZ, lacY and lacA, represented by the strings $\langle\ \overline{\text{lacZ}_s}\ \rangle$, $\langle\ \overline{\text{lacY}_s}\ \rangle$ and $\langle\ \overline{\text{lacA}_s}\ \rangle$ and attach to the mRNA to start translation, translation initiation rules $r_{41}, r_{42}$ and $r_{43}$.

$r_{41}$: $[\ \text{Rib} + \langle\ \overline{\text{lacZ}_s}\ \rangle\ ]_c \xrightarrow{c_{41}} [\ \langle\ \text{Rib}.\overline{\text{lacZ}_s}\ \rangle\ ]_c,\ c_{41} = 0.16 molec^{-1}sec^{-1}$

$r_{42}$: $[\ \text{Rib} + \langle\ \overline{\text{lacY}_s}\ \rangle\ ]_c \xrightarrow{c_{42}} [\ \langle\ \text{Rib}.\overline{\text{lacY}_s}\ \rangle\ ]_c,\ c_{42} = 0.16 molec^{-1}sec^{-1}$

$r_{43}$: $[\ \text{Rib} + \langle\ \overline{\text{lacA}_s}\ \rangle\ ]_c \xrightarrow{c_{43}} [\ \langle\ \text{Rib}.\overline{\text{lacA}_s}\ \rangle\ ]_c,\ c_{43} = 0.16 molec^{-1}sec^{-1}$

– Rules describing translation and ribosome dissociation rules

During translation ribosomes move along the sites in the mRNA represented by the substrings $\langle\ \overline{\text{lacZ}_m}\ \rangle$, $\langle\ \overline{\text{lacY}_m}\ \rangle$ and $\langle\ \overline{\text{lacA}_m}\ \rangle$, translation elongation rules $r_{44}$, $r_{45}$, $r_{47}$, $r_{48}$, $r_{50}$ and $r_{51}$. When ribosomes reach the termination site, $\overline{\text{lacZ}_e}$, $\overline{\text{lacY}_e}$ or $\overline{\text{lacA}_e}$, they dissociate from the mRNA releasing the proteins, $\beta$-galactosidase, LacY or LacA, translation termination rules $r_{46}$, $r_{49}$ and $r_{52}$.

$r_{44}$: $[\ \langle\ \text{Rib}.\overline{\text{lacZ}_s}\ \rangle\ ]_c \xrightarrow{c_{44}} [\ \langle\ \overline{\text{lacZ}_s}.\text{Rib}\ \rangle\ ]_c,\ c_{44} = 0.3 sec^{-1}$

$r_{45}$: $[\ \langle\ \text{Rib}.\overline{\text{lacZ}_m}\ \rangle\ ]_c \xrightarrow{c_{45}} [\ \langle\ \overline{\text{lacZ}_m}.\text{Rib}\ \rangle\ ]_c,\ c_{45} = 0.3 sec^{-1}$

$r_{46}$: $[\ \langle\ \text{Rib}.\overline{\text{lacZ}_e}\ \rangle\ ]_c \xrightarrow{c_{46}} [\ \beta{-}\text{Galac} + \text{Rib} + \langle\ \overline{\text{lacZ}_e}\ \rangle\ ]_c,\ c_{46} = 0.3 sec^{-1}$

$r_{47}$: $[\ \langle\ \text{Rib}.\overline{\text{lacY}_s}\ \rangle\ ]_c \xrightarrow{c_{47}} [\ \langle\ \overline{\text{lacY}_s}.\text{Rib}\ \rangle\ ]_c,\ c_{47} = 0.3 sec^{-1}$

$r_{48}$: $[\ \langle\ \text{Rib}.\overline{\text{lacY}_m}\ \rangle\ ]_c \xrightarrow{c_{48}} [\ \langle\ \overline{\text{lacY}_m}.\text{Rib}\ \rangle\ ]_c,\ c_{48} = 0.3 sec^{-1}$

$r_{49}$: $[\ \langle\ \text{Rib}.\overline{\text{lacY}_e}\ \rangle\ ]_c \xrightarrow{c_{49}} [\ \text{LacY} + \text{Rib} + \langle\ \overline{\text{lacY}_e}\ \rangle\ ]_c,\ c_{49} = 0.3 sec^{-1}$

$r_{50}$: $[\ \langle\ \text{Rib}.\overline{\text{lacA}_s}\ \rangle\ ]_c \xrightarrow{c_{50}} [\ \langle\ \overline{\text{lacA}_s}.\text{Rib}\ \rangle\ ]_c,\ c_{50} = 0.3 sec^{-1}$

$r_{51}$: $[\ \langle\ \text{Rib}.\overline{\text{lacA}_m}\ \rangle\ ]_c \xrightarrow{c_{51}} [\ \langle\ \overline{\text{lacA}_m}.\text{Rib}\ \rangle\ ]_c,\ c_{51} = 0.3 sec^{-1}$

$r_{52}$: $[\ \langle\ \text{Rib}.\overline{\text{lacA}_e}\ \rangle\ ]_c \xrightarrow{c_{52}} [\ \text{LacA} + \text{Rib} + \langle\ \overline{\text{lacA}_e}\ \rangle\ ]_c,\ c_{52} = 0.3 sec^{-1}$

– Rules describing mRNA degradation

mRNA is degraded when specific enzymes bind to the RBS, $\langle\ \overline{\text{lacZ}_s}\ \rangle$, $\langle\ \overline{\text{lacY}_s}\ \rangle$ and $\langle\ \overline{\text{lacA}_s}\ \rangle$, and remove them preventing new ribosomes from initiating translation, degradation rules $r_{53}$, $r_{54}$ and $r_{55}$.

$r_{53}$: $[\ \langle\ \overline{\text{lacZ}_s}\ \rangle\ ]_c \xrightarrow{c_{53}} [\ \ ]_c,\ c_{53} = 0.2 sec^{-1}$

$r_{54}$: $[\ \langle\ \overline{\text{lacY}_s}\ \rangle\ ]_c \xrightarrow{c_{54}} [\ \ ]_c,\ c_{54} = 0.2 sec^{-1}$

$r_{55}$: $[\ \langle\ \overline{\text{lacA}_s}\ \rangle\ ]_c \xrightarrow{c_{55}} [\ \ ]_c,\ c_{55} = 0.2sec^{-1}$

Summing up, our P system specification of the gene expression regulation system in the lac operon consist in the following construct:

$$\Pi_{lac} = (O, \{e, s, c\}, [\ [\ [\ ]_3\ ]_2\ ]_1, M_1, M_2, M_3, R_e, R_s, R_c) \tag{6.1}$$

where:

- The alphabet $O$ collects the objects representing the molecular entities, proteins, DNA sites and RNA sites, involved in the lac operon system:

$$O = O_{prot} \cup O_{dna} \cup O_{rna}$$

- The labels $\{e, s, c\}$ identify the type of the compartments defined by the membrane structure in $\Pi_{lac}$, these being the environment, cell surface and cytoplasm respectively.

- The membrane structure consists of three membranes defining the three relevant regions in the lac operon system, namely, the environment identified with the number 1, the cell surface identified with the number 2 and the cytoplasm identified with the number 3. A Venn diagram representation of the membrane structure in our P system specification $\Pi_{lac}$ is depicted in Figure 6.2.

- The initial multisets $M_1$, $M_2$ and $M_3$ are part of the parameters of our P system specification $\Pi_{lac}$. They associate the label $e$ with membrane 1 which represents the environment, the label $s$ with membrane 2 which represents the cell surface and the label $c$ with membrane 3 which represents the cytoplasm.

- The set of rewriting rules on multisets of objects and strings $R_e, R_s$ and $R_c$ are associated with the compartments representing the environment, cell surface and cytoplasm respectively. These rules describe the molecular interactions that take place in the specific compartment they are associated with. Next we enumerate the rules associated with each compartment.

  $R_e = \{r_4, r_8\}$

  $R_s = \{r_2, r_3, r_5, r_6, r_9, r_{11}, r_{18}, r_{20}, r_{22}\}$

  $R_c = \{r_1, r_7, r_{10}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{19}, r_{21}, r_{23}, \ldots, r_{55}\}$

### 6.2.2. P System Models of the Lac operon Regulation System

In this section we introduce a family of P system models which will allow us to study the behaviour of the lac operon gene regulation system under different initial conditions.

Once our P system specification of the lac operon system, $\Pi_{lac}$, has been designed it is necessary to identify the parameters associated with it, $\mathcal{P}(\Pi_{lac})$, and specify a set of possible values for them to generate a family of P system models. This family of P system models will allow us to study the behaviour of the lac operon system under different conditions.

According to Definition 3.2 the parameters of our P system specification $\mathcal{P}(\Pi_{lac})$ consists of the initial multisets and stochastic constants associated with the rewriting rules:

$$\mathcal{P}(\Pi_{lac}) = (\mathcal{M}_0(\Pi_{lac}), \mathcal{C}(\Pi_{lac}))$$

- In our case the stochastic constants associated with the rewriting rules $\mathcal{C}(\Pi_{lac})$ will have a fixed value collected in $C$. The values in $C$ were presented before during the enumeration of the rewriting rules. It has been possible to determine good estimates for every stochastic constant because the lac operon is one of the most studied and characterised gene regulation system, being considered the canonical example in prokaryotic gene regulation. The stochastic constants used in this work were deduced from the literature or were computed, according to the discussion in section 2.6, from some deterministic kinetic constants used in ODE models, [58, 66, 69, 77, 106].

- The initial multisets $\mathcal{M}_0(\Pi_{lac})$ constitute the actual parameters of our system. In the initial multisets we specify the initial number of objects and strings present in each compartment and the label associated with them. Different initial conditions will be described using different initial multisets. Each choice from these possible initial multisets will produce a different model which will allow us to study the behaviour of the lac operon regulation system under the initial condition represented by the chosen initial multisets.

  The possible values associated with the initial multisets $\mathbb{M}_0 = (\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3)$ will describe four different initial conditions in the environment, $\mathbf{M}_1 = \{M_1^1, M_1^2, M_1^3, M_1^4\}$, and a single initial condition for the cell surface $\mathbf{M}_2 = \{M_2\}$ and cytoplasm $\mathbf{M}_3 = \{M_3\}$.

  More specifically, the four initial multisets associated with the environment are:

- Initial multiset representing the absence of glucose and lactose in the environment:

$$M_1^1 = (e, \lambda, \lambda)$$

- Initial multiset representing abundant glucose and absence of lactose in the environment:

$$M_1^2 = (e, \mathrm{Gluc}^{300000}, \lambda)$$

- Initial multiset representing the absence of glucose but abundant lactose in the environment:

$$M_1^3 = (e, \mathrm{Lact}^{300000}, \lambda)$$

- Initial multiset representing abundant glucose and lactose in the environment:

$$M_1^4 = (e, \mathrm{Gluc}^{300000} + \mathrm{Lact}^{300000}, \lambda)$$

For the cell surface there is a single initial configuration representing in the initial multiset $M_2$:

$$M_2 = (s, \mathrm{EIICB}^{2500} + \mathrm{EIICB} \sim \mathrm{P}^{15000} + \mathrm{AC}^{10000} + \mathrm{LacY}^{3000}\}, \lambda)$$

For the cytoplasm there is also a single initial configuration representing in the initial multiset $M_3$:

$$M_3 = (c, w_3, s_3)$$

where $w_3$ represents the initial objects

$w_3 = \mathrm{RNAP}^{300} + \mathrm{Rib}^{3000} + \mathrm{EIIA}^{2000} + \mathrm{EIIA} \sim \mathrm{P}^{13000} + \mathrm{ATP}^{1000000} + \beta - \mathrm{Galac}^{3000}$ $+ \mathrm{LacI}^{1500} + \mathrm{CRP}^{10000}$

and $s_3$ the initial string representing the lac operon:

$$\langle\ \mathrm{cap.\ op}\ .\ \overbrace{\mathrm{lacZ}_s.\mathrm{lacZ}_m.\cdots.\mathrm{lacZ}_m.\mathrm{lacZ}_e}^{30}\ .\ \overbrace{\mathrm{lacY}_s.\mathrm{lacY}_m.\cdots.\mathrm{lacY}_m.\mathrm{lacY}_e}^{12}\ .$$
$$\overbrace{\mathrm{lacA}_s.\mathrm{lacA}_m.\cdots.\mathrm{lacA}_m.\mathrm{lacA}_e}^{6}\ \rangle$$

These parameters produce a family, $\mathcal{F}_{lac}(\Pi_{lac}; (\mathbb{M}_0, C))$, consisting of four different P system models associate to our P system specification of the lac operon, $\Pi_{lac}$:

1. $PSM_1 = (\Pi_{lac}; (M_1^1, M_2, M_3), C)$
2. $PSM_2 = (\Pi_{lac}; (M_1^2, M_2, M_3), C)$
3. $PSM_3 = (\Pi_{lac}; (M_1^3, M_2, M_3), C)$
4. $PSM_4 = (\Pi_{lac}; (M_1^4, M_2, M_3), C)$

In the following section we will analyse the behaviour of the lac operon under different initial conditions by running simulations of the previous four P systems models, $PSM_1$, $PSM_2$, $PSM_3$ and $PSM_4$.

## 6.3. Analysis of the Gene Expression in the Lac Operon

In this section we will present an analysis of the gene expression regulation system of the lac operon under different conditions in the media or environment. The analysis will be carried out through simulation using the Multicompartmental Gillespie's Algorithm described in section 3.3 and a simulator which implements this algorithm is available from [134].

The initial conditions that are going to be studied are absent of lactose and glucose; no glucose but abundant lactose; abundant glucose but no lactose and abundant glucose and lactose. These four conditions are described in the four P systems models, $PSM_1, PSM_2, PSM_3$ and $PSM_4$ designed in the previous section. The behaviour of each P system model will be analysed by presenting the evolution over time of the number of molecules of some key proteins and enzymes, as well as, by describing the configuration of the switch of the lac operon which determines the level of expression of the genes codified in the lac operon.

- Behaviour of the system with **no glucose** and **no lactose** in the environment, P system model $PSM_1$.

  When glucose is not present in the environment EIIA∼P is not consumed by the glucose transport system and it activates AC on the cell surface. Once active AC produces a high number of cAMP molecules, as shown in Figure 6.5. cAMP binds to the protein CRP to produce the complex CRP-cAMP which in turns through dimerisation yields the activator CRP-cAMP$_2$.

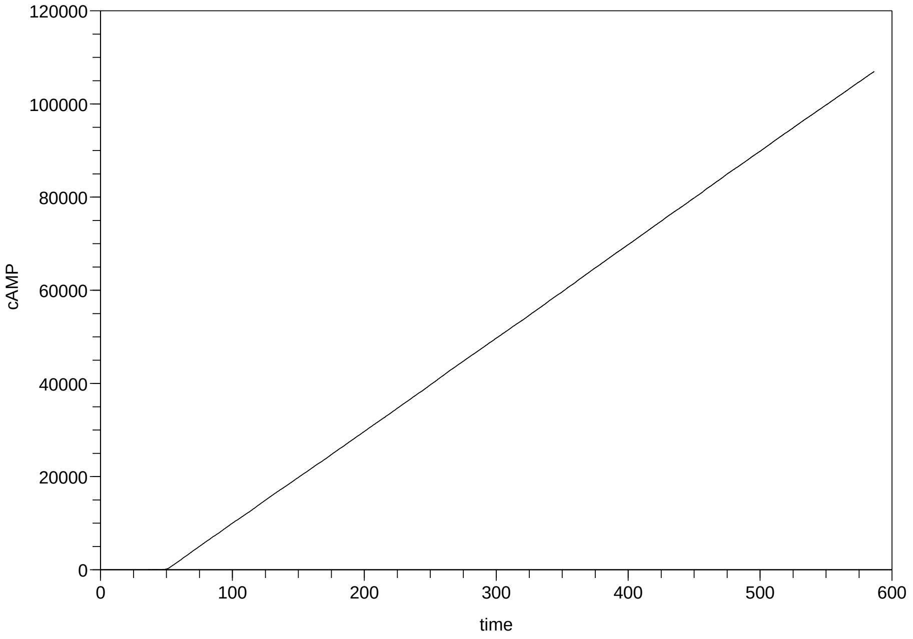

Figure 6.5.: Number of cAMP molecules over time in absence of glucose and lactose

With a high number of activators, the CAP site will be occupied by a CRP-cAMP$_2$ molecule, which will assist RNA polymerase in binding to the promoter. Nonetheless, since there is no lactose in the environment no allolactose will be produced in the cytoplasm and therefore the repressor will be active and bound to the operator. In these conditions a characteristic state of the lac operon is presented in the string in (6.2) obtained using our simulator.

*cap-CRP-cAMP_2* **RNAP** *op-LacI* lacZ_s lacZ_m **–op Rib**
**–lacZ_s –lacZ_m RNAP** lacZ_m lacZ_m lacZ_m lacZ_m
lacZ_m lacZ_m lacZ_m **–op Rib –lacZ_s –lacZ_m –lacZ_m**
**–lacZ_m –lacZ_m –lacZ_m –lacZ_m–lacZ_m –lacZ_m RNAP**
lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m
lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m
lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_e
lacY_s lacY_m lacY_m lacY_m lacY_m lacY_m lacY_m lacY_m
lacY_m lacY_m lacY_m lacY_m lacY_e lacA_s lacA_m lacA_m
lacA_m lacA_m lacA_e (6.2)

Note that CRP-cAMP$_2$ is bound to the CAP site assisting the RNA polimerase, RNAP (in bold), to bind to the promoter of the lac operon. Once bound to the promoter RNA is ready to start transcription whenever the repressor drops from the operator. This event happens occasionally permitting the transcription of the lac operon, as it can be seen in (6.2) where two polimerases are transcribing. Nevertheless, the repressor will be bound to the operator most of the time blocking transcription initiation.

Summing up, the configuration of the lac operon switch under the conditions represented in the P system model $PSM_1$, $\langle$ cap$^{\text{CRP-cAMP}_2}$.RNAP.op$^{\text{LacI}}$ $\rangle$, produces a slight increase in the expression of the genes encoded in the lac operon. This makes physiological sense, with no sugar in the environment the bacterium sets the operon such that it can respond immediately and efficiently to the presence of lactose.

- Behaviour of the system with **abundant glucose** but **no lactose** in the environment, P system model $PSM_2$.

  When glucose is abundant in the environment EIIA$\sim$P is depleted rapidly from the cytoplasm by the glucose transport system which in turn produces a high number of EIIA molecules. EIIA inhibits the activity of AC preventing it from assisting in the synthesis of cAMP and consequently there is a low number of activators, see Figure 6.6. Therefore, in these conditions, no activator will be bound at the lac operon CAP site.

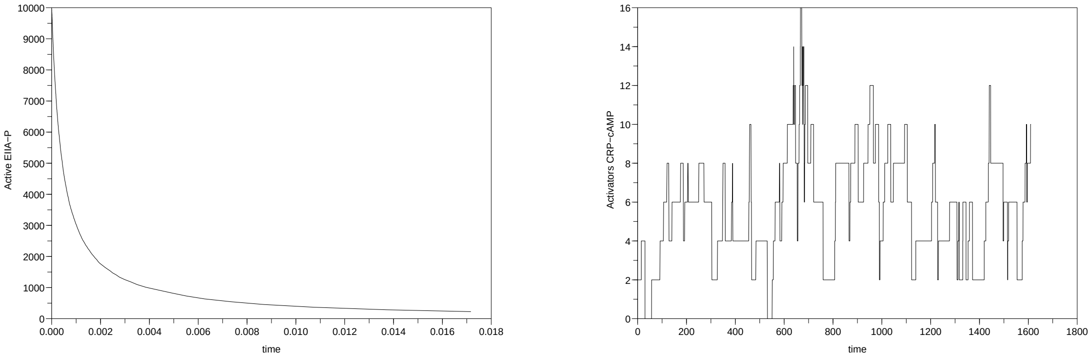

Figure 6.6.: Number of active EIIA$\sim$P and activator CRP-cAMP

Since there is no lactose in the environment no allolactose will appear in the cytoplasm and the repressor will be active and bound to the operator. Therefore in these conditions the configuration of the switch will be $\langle$ cap.op$^{\text{LacI}}$ $\rangle$. This configuration only allows transcription of the lac operon at a very low rate, as can be seen in the low number of RNA polymerases transcribing the lac operon in Figure 6.7.

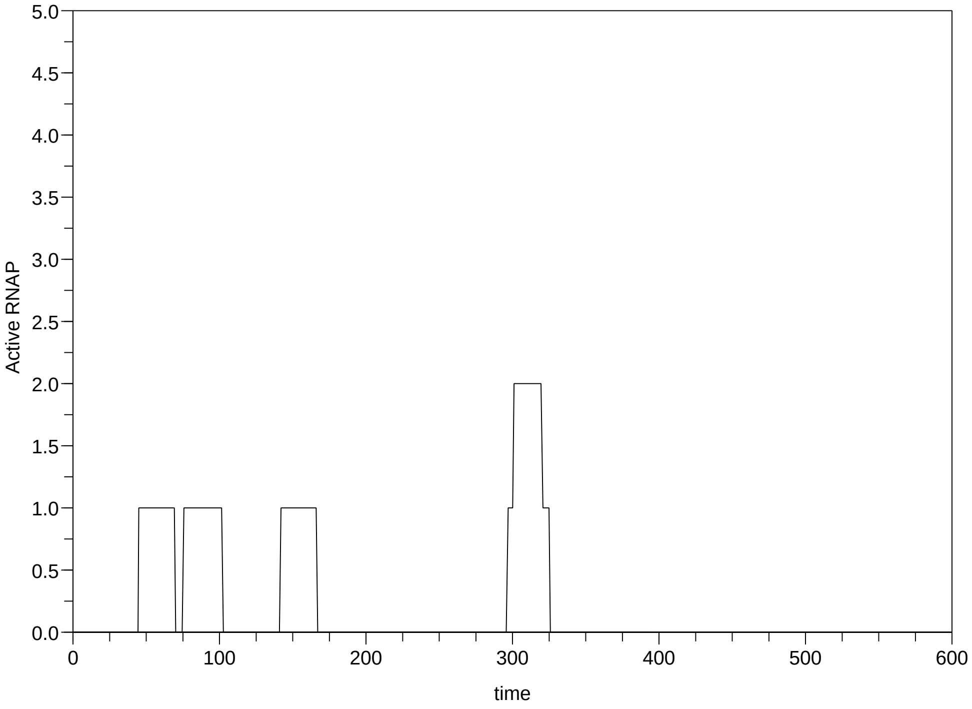

Figure 6.7.: Number of polymerases transcribing the lac operon in presence of glucose

This makes physiological sense. As long as glucose is present in the growth medium there is little need to metabolise lactose and the lactose operon is switched off.

- Behaviour of the system with **no glucose** but **abundant lactose** in the environment, P system model $PSM_3$.

  In these conditions, on the one hand, the absence of glucose in the environment allows the enzyme EIIA∼P to interact with AC in the synthesis of a high number of cAMP. This in turn produces a high number of activators molecules CRP-cAMP$_2$. Due to the high number of activator molecules the CAP site will be occupied which enhances transcription by recruiting RNA polymerase.

  On the other hand, since lactose is abundant in the environment it will be transported by the permease LacY, expressed at a basal level, into the cytoplasm.

$\beta$-galactosidase is also present at a basal level in the cytoplasm and as soon as lactose is present it starts to cleave it into galactose and glucose, and occasionally allolactose appears as a product of the interaction between $\beta$-galactosidase and lactose. Allolactose acts as an inducer binding to the repressor LacI and preventing it from binding to the operator. In Figure 6.8 it is depicted how the number of active repressors is rapidly inhibited when lactose is abundant.

Under these conditions the lac operon will be both induced (no repressor will be bound to the operator) and activated (the activator will be bound to the CAP site). Therefore, the configuration of the switch will be $\langle\ \text{cap}^{\text{CRP-cAMP}_2}.\text{op}\ \rangle$ and the genes encoded in the operon will be transcribed massively, as can be deduced by the number of RNAP transcribing the operon, Figure 6.8. This will result in a drastic increase in the number of $\beta-$galactosidase and LacY molecules, see Figure 6.9.

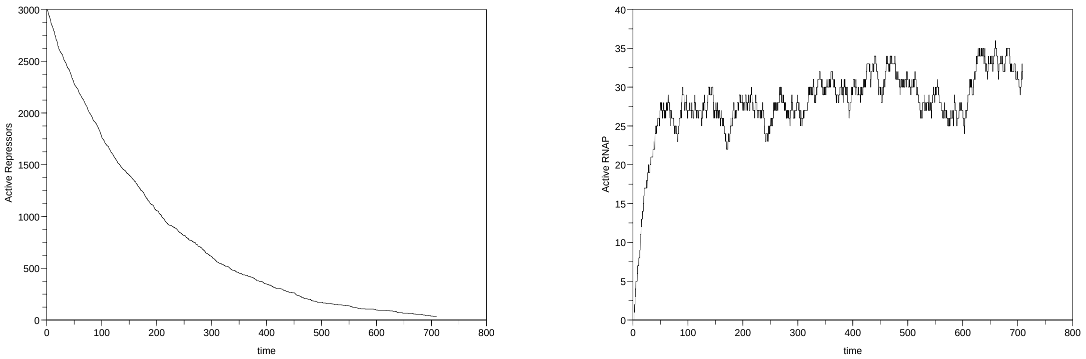

Figure 6.8.: Number of active repressors, left graph, and active RNA polymerases, right graph

- Behaviour of the system with **abundant glucose** and **abundant lactose** in the environment, P system model $PSM_4$.

Again the presence of glucose produces a low number of activators and the lac operon will not be activated by the binding of a CRP-cAMP$_2$ molecule to the CAP site. Therefore, even in the presence of lactose the genes encoded in the lac operon will be very rarely transcribed. This phenomenon is known as catabolite repression. There will also be little uptake of lactose from the environment as the permease LacY will be inhibited. Observe in Figure 6.10 that the number of glucose molecules decreases in the environment whereas lactose remains almost

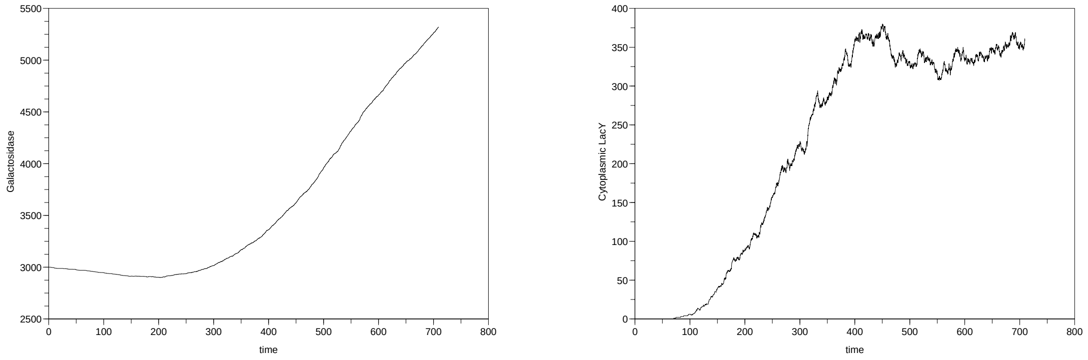

Figure 6.9.: Number of $\beta-$galactosidase, left graph, and permease LacY, right graph, over time

constant which indicates that almost no lactose is transported inside the bacterial cell.

Nevertheless, some lactose will be present in the cytoplasm which will produce allolactose which is able to inhibit the repressor to some extent producing a low transcription rate of the operon.

Summing up, in these conditions the configuration of the lac operon switch will be $\langle$ cap.op $\rangle$. That is, the lac operon will be induced, no repressor will be bound to the operator but, since no activator CRP-cAMP$_2$ will be bound to the CAP site, the lac operon will not be activated. This configuration of the lac operon switch produces a relatively low rate of transcription of the genes encoded in the lac operon. This can be seen in figure 6.11 where the number of galactosidase and LacY start to increase slowly.

Again, this makes physiological sense. As long as the bacterium can metabolise glucose there is little need to metabolise lactose. However, since lactose is now present, the cell will not ignore a sugar supply completely.

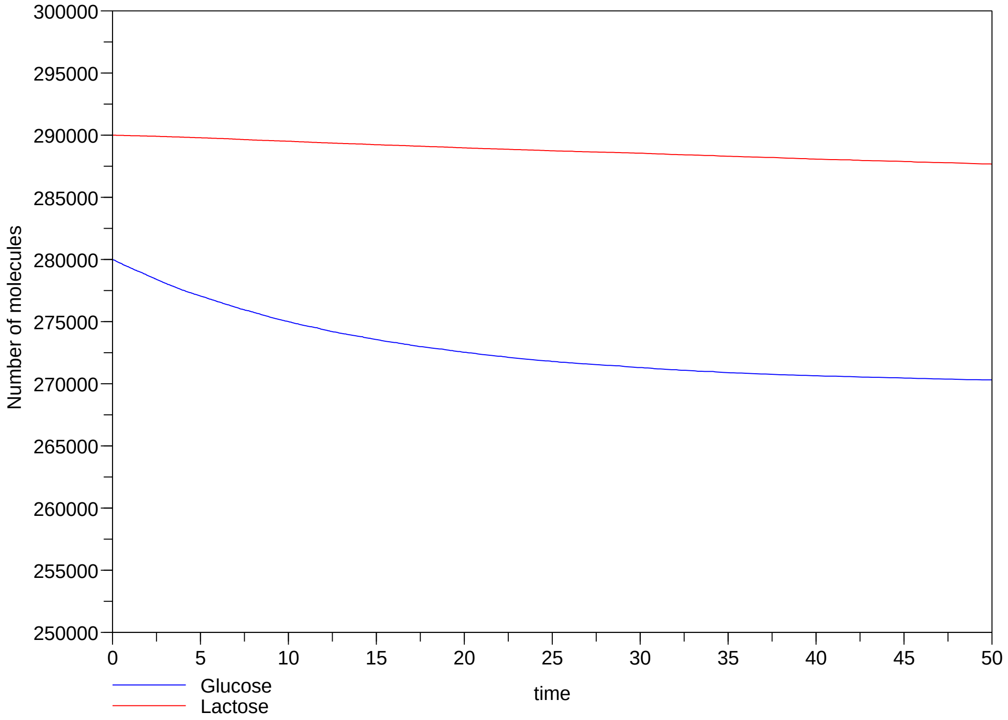

Figure 6.10.: Glucose and lactose molecules in the environment

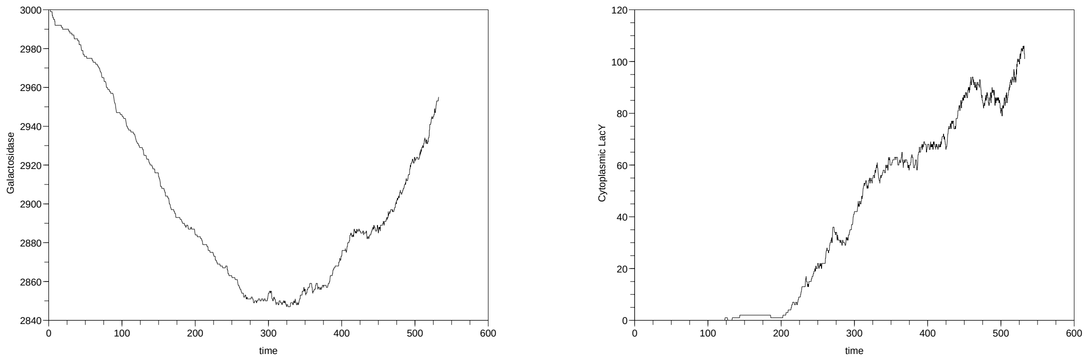

Figure 6.11.: Number of $\beta-$galactosidase, left graph, and permease LacY, right graph, over time

# 7. Modelling Signal Transduction

One of the most important cellular processes consists of the sensing of external signals by transmembrane receptors located on the cell surface. Cells monitor the state of the changing environment through these processes and produce the correct response adapting their functioning according to the signals they receive. These processes are the subject of very intensive research as they are involved in many key cellular systems like cell division, apoptosis, cell differentiation etc. Malfunctioning in the signalling pathways that constitute signal transduction systems can cause many cancer related processes like tumourgenesis, angiogenesis, uncontrolled cell proliferation, etc. The importance of membranes in the functioning and structure of signal transduction systems is obvious making P systems a suitable framework for the development of models of such systems.

In this chapter we present a brief description of the general principles of signal transduction systems in section 7.1. A deterministic version of the Multicompartmental Gillespie's algorithm introduced in section 3.3 will be introduced in section 7.2. This algorithm will be referred to as the Deterministic Waiting Times Algorithm and will be used in the evolution of the P systems models developed in this chapter. Two different signal transduction systems will be studied in this chapter, namely the Epidermal Growth Factor signalling cascade, section 7.3, and FAS induced apoptosis, section 7.4. A P system specification of each of these systems will be developed as well as a family of P systems models which will allow us to study the robustness of the system in the case of the EGFR signalling cascade and to check the validity of various hypotheses about different protein-protein interactions in the case of the FAS-induced apoptosis.

## 7.1. Signal Transduction

Signal transduction is a collection of cellular processes by which cells translate extracellular signals into specific cellular responses. Typically, cells sense extra cellular signals through direct binding of these signals to receptors placed on the cell surface. Signal transduction involves a large diversity of molecular processes, including binding and debinding of signals to extracellular domains of transmembrane receptors, recruitment and releasing of proteins and enzymes from and to the cytoplasm, molecular degradation and transformation (mainly phosphorylation), etc. Signalling pathways involve molecular entities of different types, such as receptors, enzymes, signals etc. These molecules can assemble dynamically into highly organised complexes.

Modelling of the dynamic behaviour of signalling pathways is not straight forward. Knowledge about components of the pathway and their interactions are still limited and incomplete. Furthermore, the effect of a signal often changes the state of the whole cell, and this implies difficulties for determination of the system limits. This makes the development of computational/mathematical models that can help in elucidating properties of signal transduction systems a necessity.

A common sequence of events in the signalling pathways which constitute signal transduction systems is shown in Figure 7.1 and proceeds as follows. The signal (a substance acting as a ligand) approaches the cell surface. Cells have developed two different modes of importing a signal.

On the one hand, the stimulus may penetrate the cell membrane and bind to its respective receptor in the cell interior. An example of this first system for sensing an extracellular signal is the case of lactose in the lac operon gene regulation system in the previous chapter.

On the other hand, the signal can be perceived by a transmembrane receptor. In this case the signal does not cross the membrane. The signal acts as a ligand whose target is a receptor located on the cell surface. This chapter is focused on the specification and analysis of this type of signal transduction system.

Once the signal is bound to the receptor it produces a conformational change in its cytosolic domains which yields the activation of the receptor. The active receptor in turn triggers subsequent processes within the cell consisting typically of signalling cascades. These cascades frequently include a series of changes in protein phosphorylation states. Finally, the sequence of state changes crosses the nuclear membrane producing a change in the activation state of one or more transcription factors. Commonly, the transcription factors affected by a signalling cascade change their binding properties to regulatory regions on the DNA sequence of a set of genes resulting in a change in the transcription rates of these genes, normally increasing them. These newly produced proteins constitute the actual response of the cell to the extracellular signal.

In addition to this downstream program, signalling pathways are regulated by a number of control mechanisms including feedback loops, feed-forward loops, cross-talk between different pathways, etc.

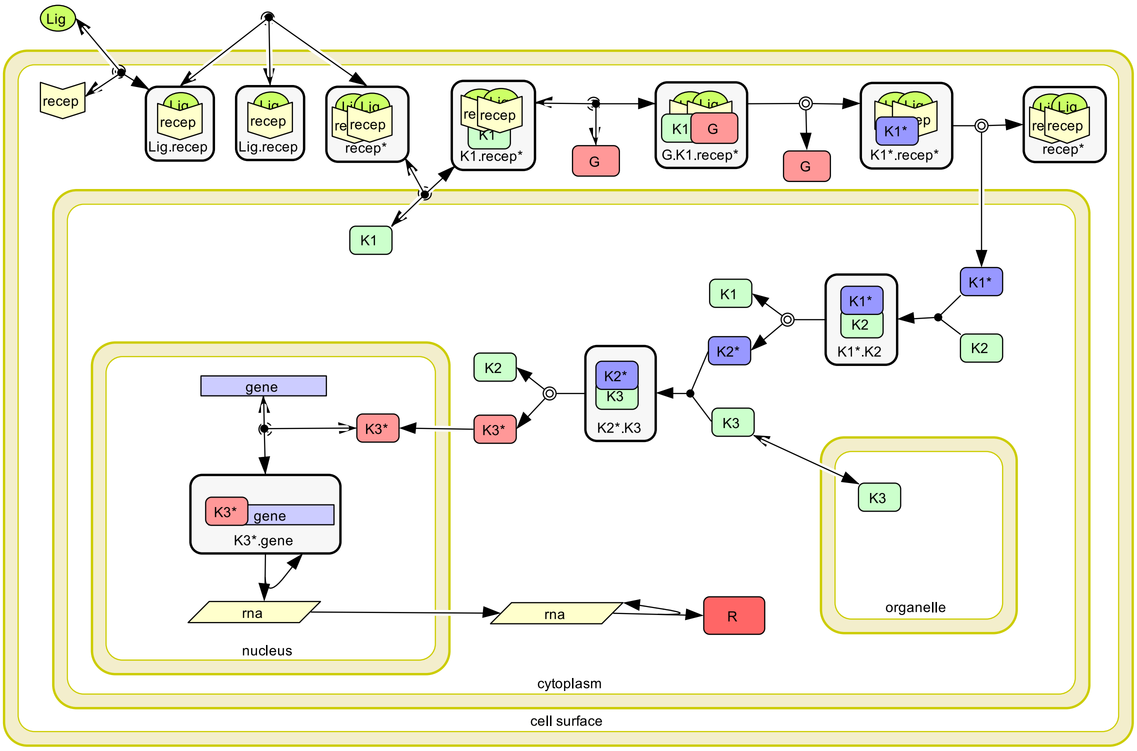

Figure 7.1.: A common sequence of events in signal transduction. A signal binds to a transmembrane receptor inducing its activation, commonly through the formation of multimers made of different receptors. Once active the receptors recruit cytoplasmic proteins and other proteins associated with the cellular membrane. This yields the activation of the cytoplasmic proteins which are released back into the cytoplasm where they start a signalling cascade involving other proteins that may interact with different organelles in the cell. Eventually, the signal reaches the nucleus where it activates the transcription of specific proteins which constitutes the actual cellular response to the extracellular signal.

## 7.2. Deterministic Waiting Times Algorithm

In section 3.3 an adaption of the classical Gillespie algorithm to the compartmentalised structure of P system models was introduced. This algorithm is referred to as the Multicomparmental Gillespie algorithm and its use was motivated by the fact that systems with a low number of molecules are not accurately modelled with the classical deterministic and continuous approach based on ODEs. Nevertheless, for systems with high number of molecules the deterministic approaches are valid to some extent. In this section, in order to study the possibility of applying deterministic strategies to simulate some cellular systems, we present the deterministic version of the Multicomparmental Gillespie algorithm which will be referred to as the *Deterministic Waiting Times Algorithm.*

Given the state of a compartment $i$, $M_i = (l_i, w_i, s_i)$, from a P system model, the next rule to be applied and its waiting time is computed as follows:

1. Compute for each rule in $r_j \in R_{l_i}$ its velocity, $v_j(M_i)$, by multiplying the stochastic constant $c_j^{l_i}$ associated specifically with rule $r_j \in R_{l_i}$ by the number of distinct possible combinations of the objects and substrings present on the left-side of the rule with respect to the current contents of membranes involved in the rule.

2. Compute the waiting time associated with rule $r_j \in R_{l_i}$ as follows:

$$\tau_j = \frac{1}{v_j(M_i)} \tag{7.1}$$

Then all the rules associated with the membranes of the P system are ordered in a priority queue according to when they are scheduled to be applied. The first rule to be applied in the whole system is the one with the shortest waiting time. Depending on the type of rule that has been applied the state of a single compartment or of two compartments is changed. Therefore the waiting time for the rules associated with these compartments must be recalculated. The algorithm stops when a prefixed simulation time is reached.

Next, we give a detailed description of the *Deterministic Waiting Times Algorithm*:

- **Initialisation**
    - set the time of the simulation $t = 0$;
    - for every rule $r_j$ associated with each membrane $i$ in $\mu$ compute the triple $(\tau_j, j, i)$ by using the procedure described before; construct a list containing all such triples;

  - ○ sort the list of triple $(\tau_j, j, i)$ according to $\tau_j$;

- **Iteration**
  - ○ extract the first triple, $(\tau_{i_0}, j_{i_0}, i_0)$ from the list;
  - ○ set the time of the simulation $t = t + \tau_{i_0}$;
  - ○ update the waiting time for the rest of the triples in the list by subtracting $\tau_{i_0}$;
  - ○ apply the rule $r_{j_{i_0}}$ in membrane $i_0$ only once changing the number of objects and sites in the membranes affected by the application of the rule;
  - ○ for each membrane $i'$ affected by the application of the rule remove all the triples corresponding to rules associated with $i'$, $(\tau'_{i'}, j_{i'}, i')$ from the list;
  - ○ for each rule associated with each membrane $i'$ affected by the application of the rule $r_{j_{i_0}}$ compute its corresponding triple as discussed above;
  - ○ add the new triples in the list and sort this list according to each waiting time and iterate the process.

- **Termination**
  - ○ Terminate simulation when the time of the simulation $t$ reaches or exceeds a preset maximal time of simulation.

Note that in this algorithm instead of associating a waiting time to a single rule in each membrane (as is the case in the *Multi-compartmental Gillespie's Algorithm*) every rule in each membrane has a waiting time computed in a deterministic way that is used to determine the order in which the rules are applied. We also highlight the fact that this is an *exact* method in the sense that we do not approximate infinitesimal intervals of time by $\Delta t$ as is the case in ODEs, but the time step varies across the evolution of the system and is computed in each step, being dependent on the current state of the system.

## 7.3. Modelling the EGFR Signalling Cascade

The epidermal growth factor receptor (EGFR) belongs to the tyrosine kinase family of receptors. Binding of the epidermal growth factor (EGF) to the extracellular domain of EGFR induces receptor dimerisation and autophosphorylation of intracellular domains. Then a multitude of proteins are recruited starting a complex signalling cascade and the receptor follows a process of internalisation and degradation in endosomals. Two principal pathways lead to activation of Ras-GTP by hydrolisation of Ras-GDP. One of these pathways depends on the Src homology and collagen domain protein (Shc) and the other one is Shc-independent. Ras-GTP acts like a *switch* that stimulates the Mitogen Activated Protein (MAP) kinase cascade by phosphorylating the proteins Raf, MEK and ERK. Subsequently phosphorylated MEK and ERK regulate several cellular proteins and nuclear transcription factors. Disregulated EGFR expression, ligand production and signalling have been proved to have a strong association with tumourgenesis. As a result of this, EGFR has been identified as a key biological target for the development of novel anticancer therapies.

Figure 7.2 depicts a detailed graphical representation of the EGFR signalling cascade.

In this section we present a P system specification and a family of models for the EGFR signalling cascade. Our specification and models will consist only of protein-protein interactions. Therefore only objects will be used to represent molecular entities and rewriting rules on multisets of objects will describe the molecular interactions of the system. In the same manner as in the previous chapter the relevant regions of the system will be specified using membranes.

The models developed in this section will allow us to study the robustness of the systems with regard to the number of extracellular signals and with regard to the number of receptors located in the cell surface.

### 7.3.1. A P System Specification for the EGFR Signalling Cascade

In what follows we present a detailed description of the P system specification for the EGFR signalling cascade in this chapter.

- **Specification of the relevant regions in the EGFR signalling cascade:**

  The role played by the cell membrane in the EGFR signalling cascade is crucial as it is the region of the cell where the receptors are located and where all the interactions producing the assembly of molecular complexes involved in signalling take place. These molecular complexes are made of transmembrane receptors, extracellular signals and cytoplasmic proteins.

  The membranes of our P system specification of the EGFR signalling cascade describe the three relevant regions of the system:

  1. The environment where the signal is located will be specified using the membrane identified with the number 1. A label $e$ will be associated with this membrane. It will constitute the root of the membrane structure describing the compartments involved in the system.

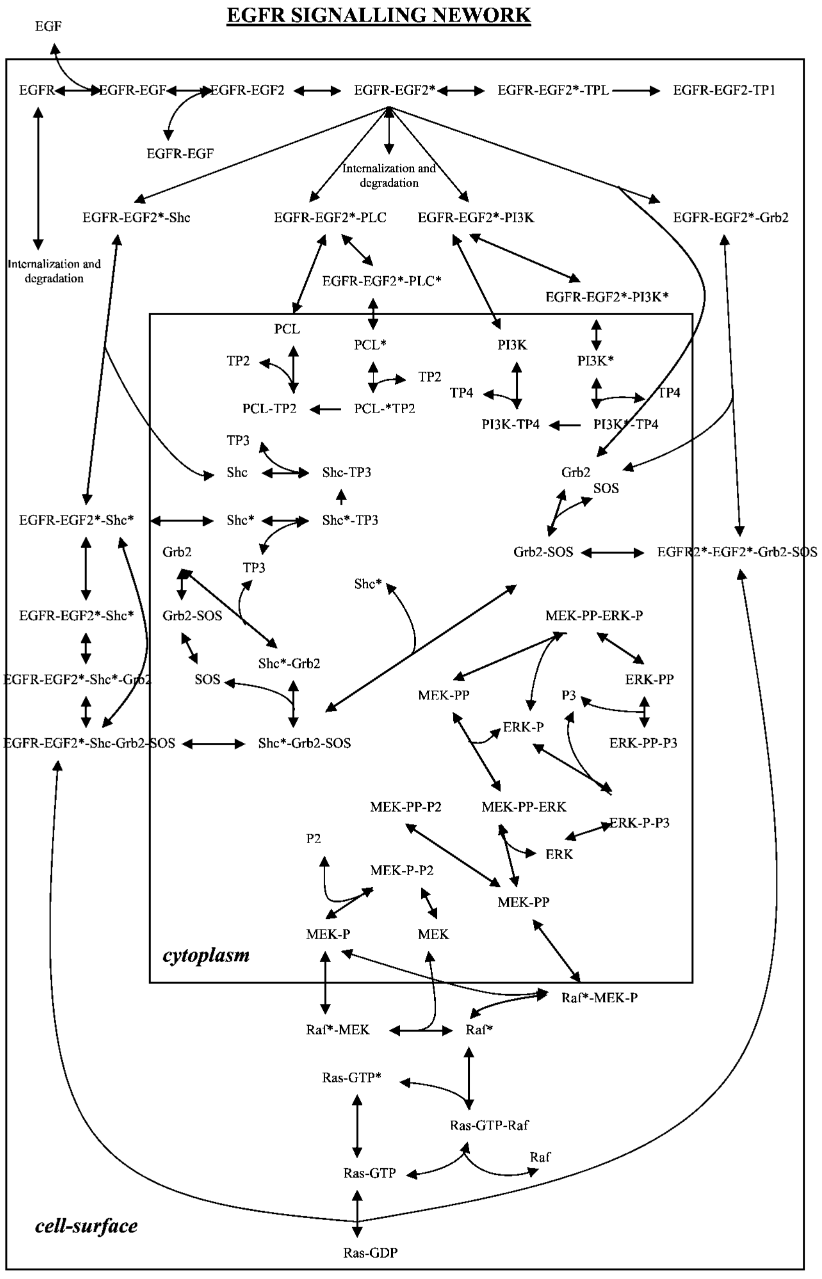

Figure 7.2.: Graphical representation of the EGFR signalling cascade

2. The cell surface where the receptor and the complexes involved in signalling as well as other proteins associated with the cell membrane are located.

This membrane is identified with the number 2. A label *s* will be associated with this membrane. This membrane will be embedded inside the previous membrane to represent the fact that the environment surrounds the cell surface.

3. The cytoplasm where the actual signalling cascade takes places. This membrane is identified with the number 3. A label *c* will be associated with this membrane. This membrane will be located inside the previous membrane to represent the fact that the cell surface wraps the cytoplasm.

A Venn diagram representation of the membrane structure can be seen in Figure 7.3.

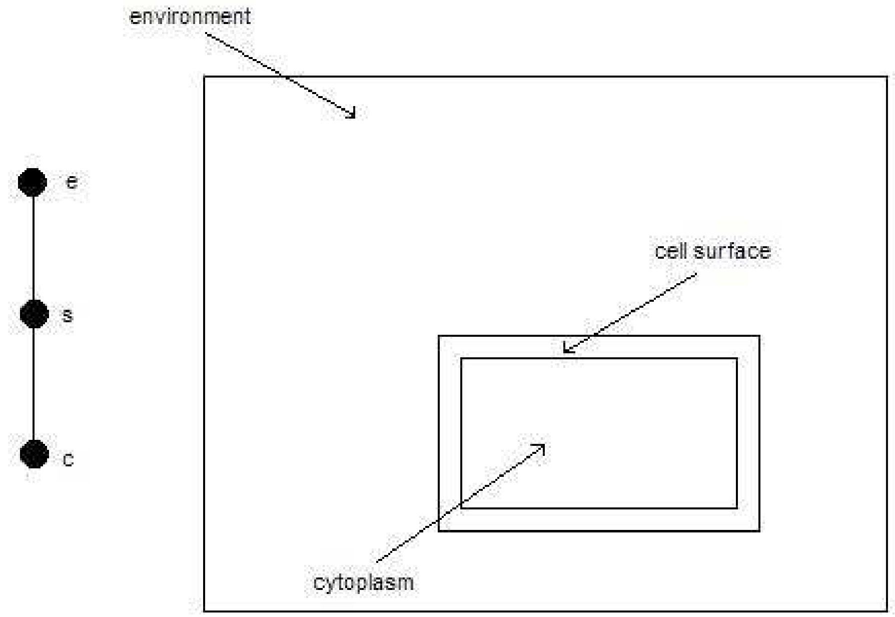

Figure 7.3.: Membrane structure in the P System Specification of the EGFR signalling cascade

- **Specification of the molecular entities in the EGFR signalling cascade:**

  As mentioned earlier we only consider molecular entities that can be represented using single objects. In this respect, the alphabet $O$ collects all the objects used to specify the signals, receptors, proteins and complexes of proteins that take part in the signalling cascade. Figure 7.4 presents the objects in $O$ and the molecular entities they represent.

- **Specification of the molecular interactions in the EGFR signalling cascade:**

| **Object** | **Molecular Entity** |
|---|---|
| EGF | Epidermal Growth Factor |
| EGFR | Epidermal Growth Factor Receptor |
| EGFR-EGF | EGFR and EGF complex |
| EGFR-EGF2, EGFR-EGF2* | Dimerised and Phosphorylated Receptor |
| TP1, TP2, TP3, TP4 | Tyrosine Phosphatase 1,2,3,4 |
| EGFR-EGF2*-TP1 | Phosphorylated Receptor and TP1 complex |
| EGFR-EGF2-TP1 | Receptor, TP1 complex |
| PLC, PCL* | Phospholipase C-$\gamma$ and its phosphorylated state |
| EGFR-EGF2*-PLC | Phosphorylated Receptor and PLC complex |
| EGFR-EGF2*-PLC* | Phosphorylated Receptor and PLC* complex |
| PLC*-TP2 | PLC* and TP2 complex |
| PLC-TP2 | PLC and TP2 complex |
| PI3K, PI3K* | Phosphatidylinositol 3-kinase and its phosphorylated state |
| EGFR-EGF2*-PI3K | Phosphorylated receptor and PI3K complex |
| EGFR-EGF2*-PI3K* | Phosphorylated receptor and PI3K* complex |
| PI3K*-TP4 | PI3K* and TP4 complex |
| PI3K-TP4 | PI3K and TP4 complex |
| Grb2 | Growth factor receptor binding protein 2 |
| EGFR-EGF2*-Grb2 | Phosphorylated receptor and Grb2 complex |
| SOS | Son of sevenless homologue protein |
| EGFR-EGF2*-Grb2-SOS | Phosphorylated receptor, Grb2, SOS complex |
| Grb2-SOS | Grb2 and SOS complex |
| Shc, Shc* | src homology 2 domain and its phosphorylated state |
| EGFR-EGF2*-Shc | Phosphorylated receptor and Shc complex |
| EGFR-EGF2*-Shc* | Phosphorylated receptor and Shc* complex |
| Shc*-TP3 | Shc* and TP3 complex |
| Shc-TP3 | Shc and TP3 complex |
| EGFR-EGF2*-Shc*-Grb2 | Phosphorylated receptor, Shc* and Grb2 complex |
| EGFR-EGF2*-Shc*-Grb2-SOS | Phosphorylated receptor, Shc*, Grb2 and SOS complex |
| Shc*-Grb2-SOS | Shc*, Grb2 and SOS complex |
| Shc*-Grb2-SOS | Shc* and Grb2complex |
| Ras-GDP, Ras-GTP, Ras-GTP* | Ras protein and its hydrolised state |
| Raf, Raf* | Raf protein and its phosphorylated state |
| Ras-GTP-Raf | Ras-GTP and Raf complex |
| MEK, MEK-P, MEK-PP | Mitogen-activated Protein Kinase and its phosphorylated states |
| ERK, ERK-P, ERK-PP | External Regulated Kinase and its phosphorylated states |
| Raf*-MEK | Raf* and MEK complex |
| Raf*-MEK-P | Raf* and MEK-P complex |
| MEK-PP-ERK | MEK-PP and ERK complex |
| MEK-PP-ERK-P | MEK-PP and ERK-P complex |
| P2, P3 | Phosphatases 2,3 |
| MEK-P-P2 | MEK-P and P2 complex |
| MEK-PP-P2 | MEK-PP and P2 complex |
| ERK-P-P3 | ERK-P and P3 complex |
| ERK-PP-P3 | ERK-PP and P3 complex |

Figure 7.4.: Objects in the EGFR P system specification

As mentioned previously in our P system specification of the EGFR signalling cascade only rewriting rules on multisets of objects will be used since we are only specifying protein-protein interactions.

In what follows we present the 160 rewriting rules used to describe the interactions between the different molecular entities involved in the signalling cascade. For each rule we also present the deterministic macroscopic constant associated with it. All these constants were obtained from ODE models enumerated in the references, [32, 68, 81, 115, 117, 119, 130]. These constants could not be used directly in our P system model, instead they were first converted into their mesoscopic and stochastic counterparts according to the discussion in section 2.6.

A brief description of the interactions represented by the rules will be presented as well.

– Receptor activation

The first step in the signalling cascade consists of the reversible binding of the signal, the Epidermal Growth Factor (EGF), to its receptor, binding and debinding rules $r_1$ and $r_2$. Once bound, EGF assists in dimerisation of the receptor, the formation of a complex made up of two receptors, complex formation and dissociation rules $r_3$ and $r_4$. The dimerisation of the receptor produces the reversible phosphorylation of both receptors yielding the active form of the receptor, transformation rules $r_5$ and $r_6$.

Phosphate $\text{TP}_1$ is involved in one of the processes of deactivation of the receptor. Specifically, $\text{TP}_1$ is recruited reversibly from the cytoplasm by the active receptor, recruitment and releasing rules $r_7$ and $r_8$. Once bound to the receptor $\text{TP}_1$ dephosphorylate it, transformation rule $r_9$ and subsequently is released reversibly back into the cytoplasm, releasing rule $r_{10}$ and recruitment rule $r_{11}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_1 : \text{EGF [ EGFR }_s \xrightarrow{c_1} \text{[ EGFR-EGF ]}_s$ | $k_1 = 3 \times 10^{-3}$ |
| $r_2 : \text{[ EGFR-EGF ]}_s \xrightarrow{c_2} \text{EGF [ EGFR ]}_s$ | $k_2 = 6 \times 10^{-2}$ |
| $r_3 : \text{[ EGFR-EGF + EGFR-EGF ]}_s \xrightarrow{c_3} \text{[ EGFR-EGF2 ]}_s$ | $k_3 = 1.1 \times 10^{-2}$ |
| $r_4 : \text{[ EGFR-EGF2 ]}_s \xrightarrow{c_4} \text{[ EGFR-EGF + EGFR-EGF ]}_s$ | $k_4 = 0.814$ |
| $r_5 : \text{[ EGFR-EGF2 ]}_s \xrightarrow{c_5} \text{[ EGFR-EGF2*]}_s$ | $k_5 = 2.71$ |
| $r_6 : \text{[ EGFR-EGF2*]}_s \xrightarrow{c_6} \text{[ EGFR-EGF2 ]}_s$ | $k_6 = 0.0271$ |
| $r_7 : \text{EGFR-EGF2*[ TP}_1\text{ ]}_c \xrightarrow{c_7} \text{EGFR-EGF2*-TP}_1\text{[ ]}_c$ | $k_7 = 0.03$ |
| $r_8 : \text{EGFR-EGF2*-TP}_1\text{[ ]}_c \xrightarrow{c_8} \text{EGFR-EGF2*[ TP}_1\text{]}_c$ | $k_8 = 5.91$ |
| $r_9 : \text{[ EGFR-EGF2*-TP}_1\text{]}_s \xrightarrow{c_9} \text{[ EGFR-EGF2-TP}_1\text{]}_s$ | $k_9 = 7.44$ |
| $r_{10} : \text{EGFR-EGF2-TP}_1\text{[ ]}_c \xrightarrow{c_{10}} \text{EGFR-EGF2 [ TP}_1\text{]}_c$ | $k_{10} = 5.1 \times 10^{-3}$ |
| $r_{11} : \text{EGFR-EGF2 [ TP}_1\text{]}_c \xrightarrow{c_{11}} \text{EGFR-EGF2-TP}_1\text{[ ]}_c$ | $k_{11} = 2 \times 10^{-4}$ |

– Receptor Internalisation and degradation

The main mechanism of receptor deactivation consists of receptor internalisation in the cytoplasm where subsequently it is degraded. Receptor internalisation can take place without the aid of any transport proteins, diffusion in rules $r_{12}$ and $r_{13}$

or after interaction with the protein CPP, complex formation and dissociation rules $r_{18}, r_{19}$ and diffusion in rule $r_{20}$. The internalisation of the receptor is reversible, diffusion out rule $r_{14}$. Once in the cytoplasm the receptor can be degraded, degradation rules, $r_{15}$ and $r_{16}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{12}$ : EGFR [ ]$_c \xrightarrow{c_{12}}$ [ EGFR ]$_c$ | $k_{12} = 5 \times 10^{-5}$ |
| $r_{13}$ : [ EGFR ]$_c \xrightarrow{c_{13}}$ EGFR [ ]$_c$ | $k_{13} = 5 \times 10^{-3}$ |
| $r_{14}$ : EGFR-EGF2*[ ]$_c \xrightarrow{c_{14}}$ [EGFR-EGF2*]$_c$ | $k_{14} = 5 \times 10^{-5}$ |
| $r_{15}$ : [ EGFR ]$_c \xrightarrow{c_{15}}$ [ ]$_c$ | $k_{15} = 6.64 \times 10^{-4}$ |
| $r_{16}$ : [ EGFR-EGF2* ] $_c \xrightarrow{c_{16}}$ [ ] $_c$ | $6.64 \times 10^{-4}$ |
| $r_{17}$ : [ EGF ] $_c \xrightarrow{c_{17}}$ [ ] $_c$ | $k_{17} = 1.67 \times 10^{-4}$ |
| $r_{18}$ : [ EGFR-EGF2*-Grb2 CPP ] $_s \xrightarrow{c_{18}}$ [ EGFR-EGF2*-Grb2-CPP ] $_s$ | $k_{18} = 1.73 \times 10^{-7}$ |
| $r_{19}$ : [ EGFR-EGF2*-Grb2-CPP ] $_s \xrightarrow{c_{19}}$ [ EGFR-EGF2*-Grb2 CPP ] $_s$ | $k_{19} = 1.66 \times 10^{-3}$ |
| $r_{20}$ : EGFR-EGF2*-Grb2-CPP [ ] $_c \xrightarrow{c_{20}}$ [ EGFR-EGF2*-Grb2 CPP ] $_c$ | $k_{20} = 0.003$ |
| $r_{21}$ : [ CPP ] $_c \xrightarrow{c_{21}}$ CPP [ ] $_c$ | $k_{21} = 10^{-5}$ |

Once the receptor is active it recruits several cytoplasmic proteins assembling different complexes which trigger some signalling pathways in the cytoplasm. When these proteins are recruited they are phosphorylated and released back in the cytoplasm to initiate different signalling pathways. Next we present the P system specification of these pathways.

– PLC$_\gamma$ pathway

One of the first cytoplasmic pathways starts with the reversible recruitment of the protein PLC$_\gamma$, rules $r_{22}$ and $r_{23}$. Once recruited to the cell membrane PLC$_\gamma$ is reversibly phosphorylated, rules $r_{24}$ and $r_{25}$, and released back to the cytoplasm, rules $r_{26}$ and $r_{27}$. Once in the cytoplasm PLC$_\gamma$* can translocate to the membrane, rules $r_{28}$ and $r_{29}$, or be desphosphorylated by TP$_2$, rules $r_{30}$, $r_{31}$, $r_{32}$, $r_{33}$ and $r_{34}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{22}$ : EGFR-EGF2* [ PLC$_\gamma$ ] $_c \xrightarrow{c_{22}}$ EGFR-EGF2*-PLC$_\gamma$ [ ] $_c$ | $k_{22} = 0.1$ |
| $r_{23}$ : EGFR-EGF2*-PLC$_\gamma$ [ ] $_c \xrightarrow{c_{23}}$ EGFR-EGF2* [ PLC$_\gamma$ ] $_c$ | $k_{23} = 1$ |
| $r_{24}$ : [ EGFR-EGF2*-PLC$_\gamma$ ] $_s \xrightarrow{c_{24}}$ [ EGFR-EGF2*-PLC$^*_\gamma$ ] $_s$ | $k_{24} = 10$ |
| $r_{25}$ : [ EGFR-EGF2*-PLC$^*_\gamma$ ] $_s \xrightarrow{c_{25}}$ [ EGFR-EGF2*-PLC$_\gamma$ ] $_s$ | 0.1 |
| $r_{26}$ : EGFR-EGF2*-PLC$^*_\gamma$ [ ] $_c \xrightarrow{c_{26}}$ EGFR-EGF2* [ PLC$^*_\gamma$ ] $_c$ | $k_{26} = 1.5 \times 10^{-4}$ |
| $r_{27}$ : EGFR-EGF2* [ PLC$^*_\gamma$ ] $_c \xrightarrow{c_{27}}$ EGFR-EGF2*-PLC$^*_\gamma$ [ ] $_c$ | $k_{27} = 7.5 \times 10^{-8}$ |
| $r_{28}$ : PLC$^*_\gamma$ [ ] $_s \xrightarrow{c_{28}}$ [ PLC$^*_\gamma$ ] $_s$ | $k_{28} = 1$ |
| $r_{29}$ : [ PLC$^*_\gamma$ ] $_s \xrightarrow{c_{29}}$ PLC$^*_\gamma$ [ ] $_s$ | $k_{29} = 0.03$ |
| $r_{30}$ : [ PLC$^*_\gamma$ + TP$_2$ ] $_c \xrightarrow{c_{30}}$ [ PLC$^*_\gamma$-TP$_2$ ] $_c$ | $k_{30} = 1.5 \times 10^{-4}$ |
| $r_{31}$ : [ PLC$^*_\gamma$-TP$_2$ ] $_c \xrightarrow{c_{31}}$ [ PLC$^*_\gamma$ + TP$_2$ ] $_c$ | $k_{31} = 1.5 \times 10^{-3}$ |
| $r_{32}$ : [ PLC$^*_\gamma$-TP$_2$ ] $_c \xrightarrow{c_{32}}$ [ PLC$_\gamma$-TP$_2$ ] $_c$ | $k_{32} = 0.1$ |
| $r_{33}$ : [ PLC$_\gamma$-TP$_2$ ] $_c \xrightarrow{c_{33}}$ [ PLC$_\gamma$ + TP$_2$ ] $_c$ | $k_{33} = 10^{-5}$ |
| $r_{34}$ : [ PLC$_\gamma$ TP$_2$ ] $_c \xrightarrow{c_{34}}$ [ PLC$_\gamma$-TP$_2$ ] $_c$ | $k_{34} = 1.8 \times 10^{-7}$ |

– Grb2 pathway

One of the most important signalling pathways starts with the recruitment of the Grb2 protein, rules $r_{35}$ and $r_{36}$. Once Grb2 is recruited the protein SOS can also be recruited, rules $r_{37}$ and $r_{38}$. This situation allows the complex Grb2-SOS to be released reversibly in the cytoplasm,rules $r_{39}$ and $r_{40}$, where it can dissociate, rules $r_{40}$ and $r_{41}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{35}$ : EGFR-EGF2* [ Grb2 ] $_c \xrightarrow{c35}$ EGFR-EGF2*-Grb2 [ ] $_c$ | $k_{35} = 1.5 \times 10^{-3}$ |
| $r_{36}$ : EGFR-EGF2*-Grb2 [ ] $_c \xrightarrow{c36}$ EGFR-EGF2* [ Grb2 ] $_c$ | $k_{36} = 0.2$ |
| $r_{37}$ : EGFR-EGF2*-Grb2 [ SOS ] $_c \xrightarrow{c37}$ EGFR-EGF2*-Grb2-SOS [ ] $_c$ | $k_{37} = 0.01$ |
| $r_{38}$ : EGFR-EGF2*-Grb2-SOS [ ] $_c \xrightarrow{c38}$ EGFR-EGF2*-Grb2 [ SOS ] $_c$ | $k_{38} = 0.06$ |
| $r_{39}$ : EGFR-EGF2*-Grb2-SOS [ ] $_c \xrightarrow{c39}$ EGFR-EGF2* [ Grb2-SOS ] $_c$ | $k_{39} = 2.8 \times 10^{-3}$ |
| $r_{40}$ : EGFR-EGF2* [ Grb2-SOS ] $_c \xrightarrow{c40}$ EGFR-EGF2*-Grb2-SOS [ ] $_c$ | $k_{40} = 5.3 \times 10^{-5}$ |
| $r_{41}$ : [ Grb2-SOS ] $_c \xrightarrow{c41}$ [ Grb2 + SOS ] $_c$ | $k_{41} = 10^{-4}$ |
| $r_{42}$ : [ Grb2 + SOS ] $_c \xrightarrow{c42}$ [ Grb2-SOS ] $_c$ | $k_{42} = 6.7 \times 10^{-6}$ |

– Shc pathway

Another important pathway depends on the protein Shc. This protein is recruited by the active receptor, phosphorylated and released back into the cytoplasm, rules $r_{43}$ - $r_{48}$. Once in the cytoplasm it can be dephosphorylated by TP$_3$, rules $r_{49}$ - $r_{53}$. Once bound to the receptor the phosphorylated state of Shc can assist in the recruitment of the proteins Grb2 and SOS, rules $r_{54}$ - $r_{57}$, $r_{60}$ - $r_{63}$, $r_{68}$ and $r_{69}$. The complexes formed by the proteins Grb2, SOS and Shc are released into the cytoplasm, rules $r_{58}$, $r_{59}$, $r_{64}$ - $r_{67}$, where they can dissociate and interact with the Grb2 pathway. These interactions constitute the cross-talk between these two pathways.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{43}$ : EGFR-EGF2* [ Shc ] $_c \xrightarrow{c43}$ EGFR-EGF2*-Shc [ ] $_c$ | $k_{43} = 0.1$ |
| $r_{44}$ : EGFR-EGF2*-Shc [ ] $_c \xrightarrow{c44}$ EGFR-EGF2* [ Shc ] $_c$ | $k_{44} = 1$ |
| $r_{45}$ : [ EGFR-EGF2*-Shc ] $_s \xrightarrow{c45}$ [ EGFR-EGF2*-Shc* ] $_s$ | $k_{45} = 20$ |
| $r_{46}$ : [ EGFR-EGF2*-Shc* ] $_s \xrightarrow{c46}$ [ EGFR-EGF2*-Shc ] $_s$ | $k_{46} = 0.2$ |
| $r_{47}$ : EGFR-EGF2*-Shc* [ ] $_c \xrightarrow{c47}$ EGFR-EGF2* [ Shc* ] $_c$ | $k_{47} = 5 \times 10^{-4}$ |
| $r_{48}$ : EGFR-EGF2* [ Shc* ] $_c \xrightarrow{c48}$ EGFR-EGF2*-Shc* [ ] $_c$ | $k_{48} = 3.56 \times 10^{-7}$ |
| $r_{49}$ : [ Shc* + TP$_3$ ] $_c \xrightarrow{c49}$ [ Shc*-TP$_3$ ] $_c$ | $k_{49} = 5 \times 10^{-3}$ |
| $r_{50}$ : [ Shc*-TP$_3$ ] $_c \xrightarrow{c50}$ [ Shc* + TP$_3$ ] $_c$ | $k_{50} = 9.5 \times 10^{-2}$ |
| $r_{51}$ : [ Shc*-TP$_3$ ] $_c \xrightarrow{c51}$ [ Shc-TP$_3$ ] $_c$ | $k_{51} = 0.15$ |
| $r_{52}$ : [ Shc-TP$_3$ ] $_c \xrightarrow{c52}$ [ Shc + TP$_3$ ] $_c$ | $k_{52} = 2 \times 10^{-4}$ |
| $r_{53}$ : [ Shc + TP$_3$ ] $_c \xrightarrow{c53}$ [ Shc-TP$_3$ ] $_c$ | $k_{53} = 4 \times 10^{-6}$ |
| $r_{54}$ : EGFR2*-Shc* [ Grb2 ] $_c \xrightarrow{c54}$ EGFR2*-Shc*-Grb2 [ ] $_c$ | $k_{54} = 1.5 \times 10^{-3}$ |

| Rule | Kinetic Constant |
|---|---|
| $r_{55}$ : EGFR2*-Shc*-Grb2 [ ] $_c \xrightarrow{c_{55}}$ EGFR2*-Shc* [ Grb2 ] $_c$ | $k_{55} = 4.95 \times 10^{-2}$ |
| $r_{56}$ : EGFR2*-Shc*-Grb2 [ ] $_c \xrightarrow{c_{56}}$ EGFR2* [ Shc*-Grb2 ] $_c$ | $k_{56} = 6.5 \times 10^{-4}$ |
| $r_{57}$ : EGFR2* [ Shc*-Grb2 ] $_c \xrightarrow{c_{57}}$ EGFR2*-Shc*-Grb2 [ ] $_c$ | $k_{57} = 1.4 \times 10^{-6}$ |
| $r_{58}$ : [ Shc*-Grb2 ] $_c \xrightarrow{c_{58}}$ [ Shc* + Grb2 ] $_c$ | $k_{58} = 1 \times 10^{-3}$ |
| $r_{59}$ : [ Shc* + Grb2 ] $_c \xrightarrow{c_{59}}$ [ Shc*-Grb2 ] $_c$ | $k_{59} = 1 \times 10^{-5}$ |
| $r_{60}$ : EGFR2*-Shc*-Grb2 [ SOS ] $_c \xrightarrow{c_{60}}$ EGFR2*-Shc*-Grb2-SOS [ ] $_c$ | $k_{60} = 0.015$ |
| $r_{61}$ : EGFR2*-Shc*-Grb2-SOS [ ] $_c \xrightarrow{c_{61}}$ EGFR2*-Shc*-Grb2 [ SOS ] $_c$ | $k_{61} = 0.03$ |
| $r_{62}$ : EGFR2*-Shc*-Grb2-SOS [ ] $_c \xrightarrow{c_{62}}$ EGFR2* [ Shc*-Grb2-SOS ] $_c$ | $k_{62} = 1.1 \times 10^{-3}$ |
| $r_{63}$ : EGFR2* [ Shc*-Grb2-SOS ] $_c \xrightarrow{c_{63}}$ EGFR2*-Shc*-Grb2-SOS [ ] $_c$ | $k_{63} = 2.37 \times 10^{-6}$ |
| $r_{64}$ : [ Shc*-Grb2-SOS ] $_c \xrightarrow{c_{64}}$ [ Shc*-Grb2 SOS ] $_c$ | $k_{64} = 0.06$ |
| $r_{65}$ : [ Shc*-Grb2 SOS ] $_c \xrightarrow{c_{65}}$ [ Shc*-Grb2-SOS ] $_c$ | $k_{65} = 0.03$ |
| $r_{66}$ : [ Shc*-Grb2-SOS ] $_c \xrightarrow{c_{66}}$ [ Shc* Grb2-SOS ] $_c$ | $k_{66} = 0.035$ |
| $r_{67}$ : [ Shc* Grb2-SOS ] $_c \xrightarrow{c_{67}}$ [ Shc*-Grb2-SOS ] $_c$ | $k_{67} = 2.5 \times 10^{-3}$ |
| $r_{68}$ : EGFR2*-Shc* [ Grb2-SOS ] $_c \xrightarrow{c_{68}}$ EGFR2*-Shc*-Grb2-SOS [ ] $_c$ | $k_{68} = 0.5$ |
| $r_{69}$ : EGFR2*-Shc*-Grb2-SOS [ ] $_c \xrightarrow{c_{69}}$ EGFR2*-Shc* [ Grb2-SOS ] $_c$ | $k_{69} = 0.1$ |

– Internalized Receptor Activity

We assume that receptors at the cell surface and internalised receptors in endosomal compartments induce identical signalling cascades, except for the PLC$_\gamma$ pathway which is turned off upon receptor internalisation.

| Rule | Kinetic Constant |
|---|---|
| $r_{70}$ : [ EGF + EGFR ] $_c \xrightarrow{c_{70}}$ [ EGFR-EGF ] $_c$ | $k_{70} = 3 \times 10^{-3}$ |
| $r_{71}$ :[ EGFR-EGF ] $_c \xrightarrow{c_{71}}$ [ EGF + EGFR ] $_c$ | $k_{71} = 0.06$ |
| $r_{72}$ : [ EGFR-EGF + EGFR-EGF ] $_c \xrightarrow{c_{72}}$ [ EGFR-EGF2 ] $_c$ | $k_{72} = 1.1 \times 10^{-2}$ |
| $r_{73}$ : [ EGFR-EGF2 ] $_c \xrightarrow{c_{73}}$ [ EGFR-EGF + EGFR-EGF ] $_c$ | $k_{73} = 0.814$ |
| $r_{74}$ : [ EGFR-EGF2 ] $_c \xrightarrow{c_{74}}$ [ EGFR-EGF2* ] $_c$ | $k_{74} = 2.71$ |
| $r_{75}$ : [ EGFR-EGF2* ] $_c \xrightarrow{c_{75}}$ [ EGFR-EGF2 ] $_c$ | $k_{75} = 0.0271$ |
| $r_{76}$ : [ EGFR-EGF2* + TP$_1$ ] $_c \xrightarrow{c_{76}}$ [ EGFR-EGF2*-TP$_1$ ] $_c$ | $k_{76} = 5.1 \times 10^{-3}$ |
| $r_{77}$ : [ EGFR-EGF2*-TP$_1$ ] $_c \xrightarrow{c_{77}}$ [ EGFR-EGF2* + TP$_1$ ] $_c$ | $k_{77} = 5.91$ |
| $r_{78}$ : [ EGFR-EGF2*-TP$_1$ ] $_c \xrightarrow{c_{78}}$ [ EGFR-EGF2-TP$_1$ ] $_c$ | $k_{78} = 7.44$ |
| $r_{79}$ : [ EGFR-EGF2-TP$_1$ ] $_c \xrightarrow{c_{79}}$ [ EGFR-EGF2 + TP$_1$ ] $_c$ | $k_{79} = 5.1 \times 10^{-3}$ |
| $r_{80}$ : [ EGFR-EGF2 + TP$_1$ ] $_c \xrightarrow{c_{80}}$ [ EGFR-EGF2-TP$_1$ ] $_c$ | $k_{80} = 2 \times 10^{-4}$ |
| $r_{81}$ : [ EGFR-EGF2* + Grb2 ] $_c \xrightarrow{c_{81}}$ [ EGFR-EGF2*-Grb2 ] $_c$ | $k_{81} = 1.5 \times 10^{-3}$ |
| $r_{82}$ : [ EGFR-EGF2*-Grb2 ] $_c \xrightarrow{c_{82}}$ [ EGFR-EGF2* + Grb2 ] $_c$ | $k_{82} = 0.2$ |
| $r_{83}$ : [ EGFR-EGF2*-Grb2 + SOS ] $_c \xrightarrow{c_{83}}$ [ EGFR-EGF2*-Grb2-SOS ] $_c$ | $k_{83} = 0.01$ |
| $r_{84}$ : [ EGFR-EGF2*-Grb2-SOS ] $_c \xrightarrow{c_{84}}$ [ EGFR-EGF2*-Grb2 + SOS ] $_c$ | $k_{84} = 0.06$ |
| $r_{85}$ : [ EGFR-EGF2*-Grb2-SOS ] $_c \xrightarrow{c_{85}}$ [ EGFR-EGF2* + Grb2-SOS ] $_c$ | $k_{85} = 2.8 \times 10^{-3}$ |
| $r_{86}$ : [ EGFR-EGF2* + Grb2-SOS ] $_c \xrightarrow{c_{86}}$ [ EGFR-EGF2*-Grb2-SOS ] $_c$ | $k_{86} = 5.3 \times 10^{-5}$ |
| $r_{87}$ : [ EGFR-EGF2* + Shc ] $_c \xrightarrow{c_{87}}$ [ EGFR-EGF2*-Shc ] $_c$ | $k_{87} = 0.1$ |
| $r_{88}$ : [ EGFR-EGF2*-Shc ] $_c \xrightarrow{c_{88}}$ [ EGFR-EGF2* + Shc ] $_c$ | $k_{88} = 1$ |
| $r_{89}$ : [ EGFR-EGF2*-Shc ] $_c \xrightarrow{c_{89}}$ [ EGFR-EGF2*-Shc* ] $_c$ | $k_{89} = 20$ |
| $r_{90}$ : [ EGFR-EGF2*-Shc* ] $_c \xrightarrow{c_{90}}$ [ EGFR-EGF2*-Shc ] $_c$ | $k_{90} = 0.2$ |
| $r_{91}$ : [ EGFR-EGF2*-Shc* ] $_c \xrightarrow{c_{91}}$ [ EGFR-EGF2* + Shc* ] $_c$ | $5 \times 10^{-4}$ |
| $r_{92}$ : [ EGFR-EGF2* + Shc* ] $_c \xrightarrow{c_{92}}$ [ EGFR-EGF2*-Shc* ] $_c$ | $3.56 \times 10^{-7}$ |

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{93} : [\ \text{EGFR2}^*\text{-Shc}^* + \text{Grb2}\ ]_c \xrightarrow{c_{93}} [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2}\ ]_c$ | $1.5 \times 10^{-3}$ |
| $r_{94} : [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2}\ ]_c \xrightarrow{c_{94}} [\ \text{EGFR2}^*\text{-Shc}^* + \text{Grb2}\ ]_c$ | $4.95 \times 10^{-2}$ |
| $r_{95} : [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2}\ ]_c \xrightarrow{c_{95}} [\ \text{EGFR2}^* + \text{Shc}^*\text{-Grb2}\ ]_c$ | $6.5 \times 10^{-4}$ |
| $r_{96} : [\ \text{EGFR2}^* + \text{Shc}^*\text{-Grb2}\ ]_c \xrightarrow{c_{96}} [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2}\ ]_c$ | $1.4 \times 10^{-6}$ |
| $r_{97} : [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2} + \text{SOS}\ ]_c \xrightarrow{c_{97}} [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2-SOS}\ ]_c$ | $k_{97} = 0.015$ |
| $r_{98} : [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2-SOS}\ ]_c \xrightarrow{c_{98}} [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2} + \text{SOS}\ ]_c$ | $k_{98} = 0.03$ |
| $r_{99} : [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2-SOS}\ ]_c \xrightarrow{c_{99}} [\ \text{EGFR2}^* + \text{Shc}^*\text{-Grb2-SOS}\ ]_c$ | $k_{99} = 1.1 \times 10^{-3}$ |
| $r_{100} : [\ \text{EGFR2}^* + \text{Shc}^*\text{-Grb2-SOS}\ ]_c \xrightarrow{c_{100}} [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2-SOS}\ ]_c$ | $k_{100} = 2.37 \times 10^{-6}$ |
| $r_{101} : [\ \text{EGFR2}^*\text{-Shc}^* + \text{Grb2-SOS}\ ]_c \xrightarrow{c_{101}} [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2-SOS}\ ]_c$ | $k_{101} = 0.5$ |
| $r_{102} : [\ \text{EGFR2}^*\text{-Shc}^*\text{-Grb2-SOS}\ ]_c \xrightarrow{c_{102}} [\ \text{EGFR2}^*\text{-Shc}^* + \text{Grb2-SOS}\ ]_c$ | $k_{102} = 0.1$ |

– PI3K pathway

A marginal pathway consists of the recruitment of the protein PI3K, rules $r_{103}$ and $r_{104}$, its phosphorylation, rules $r_{104}$ and $r_{105}$; its release back to the cytoplasm, rules $r_{106}$ and $r_{107}$ where it can be dephosphorylated through interactions with $\text{TP}_4$, rules $r_{109}$ - $r_{113}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{103} :\text{EGFR-EGF2}^*\ [\ \text{PI3K}\ ]_c \xrightarrow{c_{103}} \text{EGFR-EGF2}^*\text{-PI3K}\ [\,]_c$ | $k_{103} = 0.1$ |
| $r_{104} : \text{EGFR-EGF2}^*\text{-PI3K}\ [\,]_c \xrightarrow{c_{104}} \text{EGFR-EGF2}^*\ [\ \text{PI3K}\ ]_c$ | $k_{104} = 2$ |
| $r_{105} : [\ \text{EGFR-EGF2}^*\text{-PI3K}\ ]_s \xrightarrow{c_{105}} [\ \text{EGFR-EGF2}^*\text{-PI3K}^*\ ]_s$ | $k_{105} = 9.85$ |
| $r_{106} : [\ \text{EGFR-EGF2}^*\text{-PI3K}^*\ ]_s \xrightarrow{c_{106}} [\ \text{EGFR-EGF2}^*\text{-PI3K}\ ]_s$ | $k_{106} = 0.985$ |
| $r_{107} : \text{EGFR-EGF2}^*\text{-PI3K}^*\ [\,]_c \xrightarrow{c_{107}} \text{EGFR-EGF2}^*\ [\ \text{PI3K}^*\ ]_c$ | $k_{107} = 0.047$ |
| $r_{108} : \text{EGFR-EGF2}^*\ [\ \text{PI3K}^*\ ]_c \xrightarrow{c_{108}} \text{EGFR-EGF2}^*\text{-PI3K}^*\ [\,]_c$ | $k_{108} = 4.82 \times 10^{-5}$ |
| $r_{109} : [\ \text{PI3K}^*\ \text{TP}_4\ ]_c \xrightarrow{c_{109}} [\ \text{PI3K}^*\text{-TP}_4\ ]_c$ | $k_{109} = 7 \times 10^{-4}$ |
| $r_{110} : [\ \text{PI3K}^*\text{-TP}_4\ ]_c \xrightarrow{c_{110}} [\ \text{PI3K}^*\ \text{TP}_4\ ]_c$ | $k_{110} = 7 \times 10^{-3}$ |
| $r_{111} : [\ \text{PI3K}^*\text{-TP}_4\ ]_c \xrightarrow{c_{111}} [\ \text{PI3K-TP}_4\ ]_c$ | $k_{111} = 0.03$ |
| $r_{112} : [\ \text{PI3K-TP}_4\ ]_c \xrightarrow{c_{112}} [\ \text{PI3K}\ \text{TP}_4\ ]_c$ | $k_{112} = 8 \times 10^{-5}$ |
| $r_{113} : [\ \text{PI3K}\ \text{TP}_4\ ]_c \xrightarrow{c_{113}} [\ \text{PI3K-TP}_4\ ]_c$ | $k_{113} = 6.7 \times 10^{-6}$ |

– Ras and Raf activation

The Grb2 and Shc pathway converge in the activation of the Ras protein. Once the proteins Grb2 and SOS have been recruited in one of the two pathways they can interact with Ras-GDP, rules $r_{114}$ - $r_{125}$, to produce Ras-GTP. Once Ras-GTP appears in the cell surface it can interact with Raf, rules $r_{126}$ - $r_{129}$, producing the phosphorylated state of both molecules Ras-GTP* and Raf*. Ras-GTP* can interact with the Shc pathways, rules $r_{130}$ - $r_{133}$ and Raf* can be dephosphorylated through interactions with $\text{Phosp}_1$, rules $r_{134}$ - $r_{136}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{114}$ : [ EGFR-EGF2*-Grb2-SOS + Ras-GDP ]$_s \xrightarrow{c_{114}}$ [ EGFR-EGF2*-Grb2-SOS-Ras-GDP ]$_s$ | $k_{114} = 0.015$ |
| $r_{115}$ : [ EGFR-EGF2*-Grb2-SOS-Ras-GDP ]$_s \xrightarrow{c_{115}}$ [ EGFR-EGF2*-Grb2-SOS + Ras-GDP ]$_s$ | $k_{115} = 1.3$ |
| $r_{116}$ : [ EGFR-EGF2*-Grb2-SOS-Ras-GDP ]$_s \xrightarrow{c_{116}}$ [ EGFR-EGF2*-Grb2-SOS + Ras-GTP ]$_s$ | $k_{116} = 0.5$ |
| $r_{117}$ : [ EGFR-EGF2*-Grb2-SOS + Ras-GTP ]$_s \xrightarrow{c_{117}}$ [ EGFR-EGF2*-Grb2-SOS-Ras-GDP ]$_s$ | $k_{117} = 1 \times 10^{-4}$ |
| $r_{118}$ : [ EGFR-EGF2*-Grb2-SOS + Ras-GTP* ]$_s \xrightarrow{c_{118}}$ [ EGFR-EGF2*-Grb2-SOS-Ras-GTP ]$_s$ | $k_{118} = 2.1 \times 10^{-3}$ |
| $r_{119}$ : [ EGFR-EGF2*-Grb2-SOS-Ras-GTP ]$_s \xrightarrow{c_{119}}$ [ EGFR-EGF2*-Grb2-SOS + Ras-GTP* ]$_s$ | $k_{119} = 0.4$ |
| $r_{120}$ : [ EGFR-EGF2*-Grb2-SOS-Ras-GTP ]$_s \xrightarrow{c_{120}}$ [ EGFR-EGF2*-Grb2-SOS + Ras-GDP ]$_s$ | $k_{120} = 0.023$ |
| $r_{121}$ : [ EGFR-EGF2*-Grb2-SOS + Ras-GDP ]$_s \xrightarrow{c_{121}}$ [ EGFR-EGF2*-Grb2-SOS-Ras-GTP ]$_s$ | $k_{121} = 2.2 \times 10^{-4}$ |
| $r_{122}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GDP ]$_s \xrightarrow{c_{122}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GDP ]$_s$ | $k_{122} = 0.015$ |
| $r_{123}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GDP ]$_s \xrightarrow{c_{123}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GDP ]$_s$ | $k_{123} = 1.3$ |
| $r_{124}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GDP ]$_s \xrightarrow{c_{124}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GTP ]$_s$ | $k_{124} = 0.5$ |
| $r_{125}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GTP ]$_s \xrightarrow{c_{125}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GDP ]$_s$ | $k_{125} = 1 \times 10^{-4}$ |
| $r_{126}$ : [ Ras-GTP + Raf ]$_s \xrightarrow{c_{126}}$ [ Ras-GTP-Raf ]$_s$ | $k_{126} = 0.001$ |
| $r_{127}$ : [ Ras-GTP-Raf ]$_s \xrightarrow{c_{127}}$ [ Ras-GTP + Raf ]$_s$ | $k_{127} = 0.0053$ |
| $r_{128}$ : [ Ras-GTP-Raf ]$_s \xrightarrow{c_{128}}$ [ Ras-GTP* Raf* ]$_s$ | $k_{128} = 1$ |
| $r_{129}$ : [ Ras-GTP* + Raf* ]$_s \xrightarrow{c_{129}}$ [ Ras-GTP-Raf ]$_s$ | $k_{129} = 7 \times 10^{-4}$ |
| $r_{130}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GTP* ]$_s \xrightarrow{c_{130}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GTP ]$_s$ | $k_{130} = 7.9 \times 10^{-3}$ |
| $r_{131}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GTP ]$_s \xrightarrow{c_{131}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GTP* ]$_s$ | $k_{131} = 0.3$ |
| $r_{132}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GTP ]$_s \xrightarrow{c_{132}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GDP ]$_s$ | $k_{132} = 0.023$ |
| $r_{133}$ : [ EGFR-EGF2*-Shc*-Grb2-SOS + Ras-GDP ]$_s \xrightarrow{c_{133}}$ [ EGFR-EGF2*-Shc*-Grb2-SOS-Ras-GTP ]$_s$ | $k_{133} = 2.2 \times 10^{-4}$ |
| $r_{134}$ : Raf* [ Phosp$_1$ ]$_c \xrightarrow{c_{134}}$ Raf*-Phosp$_1$ [ ]$_c$ | $k_{134} = 0.0717$ |
| $r_{135}$ : Raf*-Phosp$_1$ [ ]$_c \xrightarrow{c_{135}}$ Raf* [ Phosp$_1$ ]$_c$ | $k_{135} = 0.2$ |
| $r_{136}$ : Raf*-Phosp$_1$ [ ]$_c \xrightarrow{c_{136}}$ Raf [ Phosp$_1$ ]$_c$ | $k_{136} = 1$ |

– MEK and ERK activation

The activation of the Raf protein in the cell surface triggers the last pathway specified in this work which consists of the activation of MEK and ERK. First, MEK is recruited to the cell surface by Raf* twice where it is doubly phosphorylated and released back into the cytoplasm, rules $r_{137}$ - $r_{142}$. In the cytoplasm MEK-PP can be desphosphorylated by Phosp$_2$, rules $r_{143}$ - $r_{148}$, or it can interact with ERK, rules $r_{149}$ - $r_{154}$, to produce its doubly phosphorylated state ERK-PP. In turn ERK-PP is desphosphorylated by Phosp$_3$, rules $r_{155}$ - $r_{160}$.

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{137}$ : Raf* [ MEK ]$_c \xrightarrow{c_{137}}$ Raf*-MEK [ ]$_c$ | $k_{137} = 0.0111$ |
| $r_{138}$ : Raf*-MEK [ ]$_c \xrightarrow{c_{138}}$ Raf* [ MEK ]$_c$ | $k_{138} = 0.01833$ |
| $r_{139}$ : Raf*-MEK [ ]$_c \xrightarrow{c_{139}}$ Raf* [ MEK-P ]$_c$ | $k_{139} = 3.5$ |
| $r_{140}$ : Raf* [ MEK-P ]$_c \xrightarrow{c_{140}}$ Raf*-MEK-P [ ]$_c$ | $k_{140} = 0.0111$ |
| $r_{141}$ : Raf*-MEK-P [ ]$_c \xrightarrow{c_{141}}$ Raf* [ MEK-P ]$_c$ | $k_{141} = 0.01833$ |
| $r_{142}$ : Raf*-MEK-P [ ]$_c \xrightarrow{c_{142}}$ Raf* [ MEK-PP ]$_c$ | $k_{142} = 2.9$ |
| $r_{143}$ : [ MEK-PP + Phosp$_2$ ]$_c \xrightarrow{c_{143}}$ [ MEK-PP-Phosp$_2$ ]$_c$ | $k_{143} = 1.43 \times 10^{-2}$ |
| $r_{144}$ : [ MEK-PP-Phosp$_2$ ]$_c \xrightarrow{c_{144}}$ [ MEK-PP + Phosp$_2$ ]$_c$ | $k_{144} = 0.8$ |
| $r_{145}$ : [ MEK-PP-Phosp$_2$ ]$_c \xrightarrow{c_{145}}$ [ MEK-P + Phosp$_2$ ]$_c$ | $k_{145} = 0.058$ |
| $r_{146}$ : [ MEK-P + Phosp$_2$ ]$_c \xrightarrow{c_{146}}$ [ MEK-P-Phosp$_2$ ]$_c$ | $k_{146} = 2.5 \times 10^{-4}$ |
| $r_{147}$ : [ MEK-P-Phosp$_2$ ]$_c \xrightarrow{c_{147}}$ [ MEK-P + Phosp$_2$ ]$_c$ | $k_{147} = 0.5$ |
| $r_{148}$ : [ MEK-P-Phosp$_2$ ]$_c \xrightarrow{c_{148}}$ [ MEK + Phosp$_2$ ]$_c$ | $k_{148} = 0.058$ |

| **Rule** | **Kinetic Constant** |
|---|---|
| $r_{149}$ : [ MEK-PP + ERK ] $_c \xrightarrow{c_{149}}$ [ MEK-PP-ERK ] $_c$ | $k_{149} = 1.1 \times 10^{-4}$ |
| $r_{150}$ : [ MEK-PP-ERK ] $_c \xrightarrow{c_{150}}$ [ MEK-PP + ERK ] $_c$ | $k_{150} = 0.033$ |
| $r_{151}$ : [ MEK-PP-ERK ] $_c \xrightarrow{c_{151}}$ [ MEK-PP + ERK-P ] $_c$ | $k_{151} = 16$ |
| $r_{152}$ : [ MEK-PP + ERK-P ] $_c \xrightarrow{c_{152}}$ [ MEK-PP-ERK-P ] $_c$ | $k_{152} = 1.1 \times 10^{-4}$ |
| $r_{153}$ : [ MEK-PP-ERK-P ] $_c \xrightarrow{c_{153}}$ [ MEK-PP + ERK-P ] $_c$ | $k_{153} = 0.033$ |
| $r_{154}$ : [ MEK-PP-ERK-P ] $_c \xrightarrow{c_{154}}$ [ MEK-PP + ERK-PP ] $_c$ | $k_{154} = 5.7$ |
| $r_{155}$ : [ ERK-PP + Phosp3 ] $_c \xrightarrow{c_{155}}$ [ ERK-PP-Phosp3 ] $_c$ | $k_{155} = 0.0145$ |
| $r_{156}$ : [ ERK-PP-Phosp3 ] $_c \xrightarrow{c_{156}}$ [ ERK-PP + Phosp3 ] $_c$ | $k_{156} = 0.6$ |
| $r_{157}$ : [ ERK-PP-Phosp3 ] $_c \xrightarrow{c_{157}}$ [ ERK-P + Phosp3 ] $_c$ | $k_{157} = 0.27$ |
| $r_{158}$ : [ ERK-P + Phosp3 ] $_c \xrightarrow{c_{158}}$ [ ERK-P-Phosp3 ] $_c$ | $k_{158} = 0.005$ |
| $r_{159}$ : [ ERK-P-Phosp3 ] $_c \xrightarrow{c_{159}}$ [ ERK-P + Phosp3 ] $_c$ | $k_{159} = 0.5$ |
| $r_{160}$ : [ ERK-P-Phosp3 ] $_c \xrightarrow{c_{160}}$ [ ERK + Phosp3 ] $_c$ | $k_{160} = 0.3$ |

Summing up our P system specification of the EGFR signalling cascade consists of the following construct:

$$\mathbf{\Pi}_{EGFR} = (O, \{e, s, c\}, \mu, M_1, M_2, M_3, \mathcal{R}_e, \mathcal{R}_s, \mathcal{R}_c)$$

where:

- The alphabet $O$ collects all the objects representing the proteins and complexes of proteins involved in the system, see Figure 7.4.

- The set of labels, $L = \{e, s, c\}$, is used to identify the different compartment types where the signalling transduction takes place, namely, the environment $e$, the cell surface $s$ and the cytoplasm $c$.

- The membrane structure, $\mu$, consists of three membranes identifying the three relevant regions in the system, namely, the environment, the cell surface and the cytoplasm. A Venn diagram representation of the membrane structure in our P system specification $\Pi_{EGFR}$ is depicted in Figure 7.3.

- The initial multisets $M_1$, $M_2$ and $M_3$ are part of the parameters of our P system specification $\Pi_{EGFR}$. They associate the label $e$ with membrane 1 which represents the environment, the label $s$ with membrane 2 which represents the cell surface and the label $c$ with membrane 3 which represents the cytoplasm.

- The sets of rewriting rules $R_e, R_s$ and $R_c$ are associated with the compartments representing the environment, cell surface and cytoplasm respectively. These rules describe the molecular interactions that take place in the specific compartment they are associated with. Next we enumerate the rules associated with each compartment.

  - $R_e = \{r_1\}$

- $R_s = \{r_2 - r_6, r_8 - r_9, r_{12}, r_{14}, r_{18} - r_{20}, r_{23} - r_{26}, r_{29}, r_{36}, r_{38}, r_{39}, r_{44}, r_{45}, r_{46}$ $r_{47}, r_{55}, r_{56}, r_{62}, r_{69}, r_{104} - r_{107}, r_{114} - r_{133}, r_{135}, r_{136}, r_{138}, r_{139}, r_{141}, r_{142}\}$
- $R_c = \{r_7, r_{11}, r_{13}, r_{15} - r_{17}, r_{21}, r_{22}, r_{27}, r_{28}, r_{30} - r_{35}, r_{37}, r_{40} - r_{43}, r_{48} - r_{54},$ $r_{57} - r_{61}, r_{63} - r_{68}, r_{70} - r_{103}, r_{108} - r_{113}, r_{134}, r_{137}, r_{140}, r_{143} - r_{160}\}$

### 7.3.2. P Systems Models for the Analysis of the EGFR Signalling Cascade

Once we have developed a P system specification of the EGFR signalling cascade, $\Pi_{EGFR}$, we can study the behaviour of the system under different conditions. In this section we will study the robustness of the signalling cascade with regard to the number of signals, EGF, in the environment and the number of receptors, EGFR, in the cell surface. This will be achieved by designing a suitable family of P system models.

The first step in the design of a family of P system models associated with our P system specification consists of identifying the parameters, $\mathcal{P}(\Pi_{EGFR}) = (\mathcal{M}_0(\Pi_{EGFR}), \mathcal{C}(\Pi_{EGFR}))$.

The EGFR signalling cascade is one of the best studied and understood signal transduction systems, and thus it was possible to determine good estimates for the constants associated with the rules $\mathcal{C}(\Pi_{EGFR})$. These values were enumerated previously during the introduction of the rules.

Therefore, the actual parameters of our P system specification are the initial multisets $\mathcal{M}_0(\Pi_{EGFR}) = (\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3)$. As mentioned earlier in this section the robustness of the EGFR signalling cascade will be studied with regard to the number of signals in the environment and the number of receptors in the cell surface. The different number of initial signals will be represented in the initial multisets associated with the environment $\mathbf{M}_1$ and the different number of receptors in the cell surface will represented in the initial multisets associated with the cell surface $\mathbf{M}_2$. Different initial conditions in the cytoplasm will not be studied, thus there will be a single initial multiset associated with the cytoplasm $\mathbf{M}_3 = \{M_3\}$.

More specifically, the initial multisets associated with each membrane or compartment will be:

- The initial multisets associated with the environment $\mathbf{M}_1 = \{M_1^{100}, M_1^{200}, M_1^{300}, M_1^{400}, M_1^{1000}, M_1^{2000}\}$ will represent the initial conditions corresponding to a concetration of 100nM, 200nM, 300nM, 400nM, 1000nM and 2000nM of EGF signals, in the environment.
- The initial multisets associated with the cell surface $\mathbf{M}_2 = \{M_2^{100}, M_2^{1000}\}$ will represent the initial conditions corresponding to a concentration of 100nM and

1000nM of receptors, EGFR, in the cell surface. The same initial concentration of 200 nM of Ras-GDP will be placed in both initial multisets $M_2^{100}$ and $M_2^{1000}$.

- Finally, there will be a single initial multiset associted with the cytoplasm, $M_3$. This initial multiset will represent the initial conditions corresponding to a concetration of 250nM of ShC, 150nM of PLC$_\gamma$, 50nM of PI3K, 40nM of SOS, 80nM of Grb2, 100nM of TP$_1$, 450nM of TP$_2$, 450nM of TP$_3$, 125nM of TP$_4$, 80nM of Raf, 400nM of MEK, 400nM of ERK, 80nM of P$_1$, 80nM of P$_2$ and 300nM of P$_3$.

By combining these parameters with the P system specification $\Pi_{EGFR}$ we obtain a family of P system models $\mathcal{F}_{EGFR}(\Pi_{EGFR}; ((\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3), C))$ consisting of twelve different P systems models. These models will be denoted by $PSM_{EGFR}^{ij}$ where $PSM_{EGFR}^{ij} = (\Pi_{EGFR}; (M_1^i, M_2^j, M_3), C)$.

In what follows we present an analysis of the robustness of the EGFR signalling cascade using the previous models. The analysis will be carried out through simulation using the Deterministic Waiting Times algorithm introduced in section 7.2. The behaviour of each P system model will be analysed by presenting the evolution over time of the number of molecules of some key proteins and enzymes involved in the signalling cascade. For example, in Figure 7.5 it depicts the evolution over time of the number of autophosphorylated receptors and the number of doubled phosphorylated MEK (Mitogen External Kinase), one of the target proteins of the signalling cascade that regulates some nuclear transcription factors involved in the cell division. Note that the activation of the receptor is very fast reaching its maximum within the first 5 seconds and then it decays fast to very low levels; on the other hand the number of doubled phosphorylated MEK is more sustained around 3 nM. These results agree well with empirical observations, see [81, 115].

The EGFR signalling cascade is known to be involved in processes related to cancer development. In particular, it has been reported that an overexpression of signals EGF in the environment and of receptors, EGFR, in the cell surface occurs in cancerous cells. As mentioned before in this work these two situations, namely an overexpression of signals and receptor, will be studied separately in order to determine which of the two is really distorting the function of the signalling cascade.

- **Robustness of the system with regard to extracellular EGF signals:**

  First, we analyse the effect of the number of extracellular signals on the signalling cascade. In particular, we will show the evolution over time of the number of autophosphorylated receptors and double phosphorylated MEK in the models $PSM_{EGFR}^{100,100}$, $PSM_{EGFR}^{200,100}$, $PSM_{EGFR}^{300,100}$, $PSM_{EGFR}^{400,100}$, $PSM_{EGFR}^{1000,100}$ and $PSM_{EGFR}^{2000,100}$.

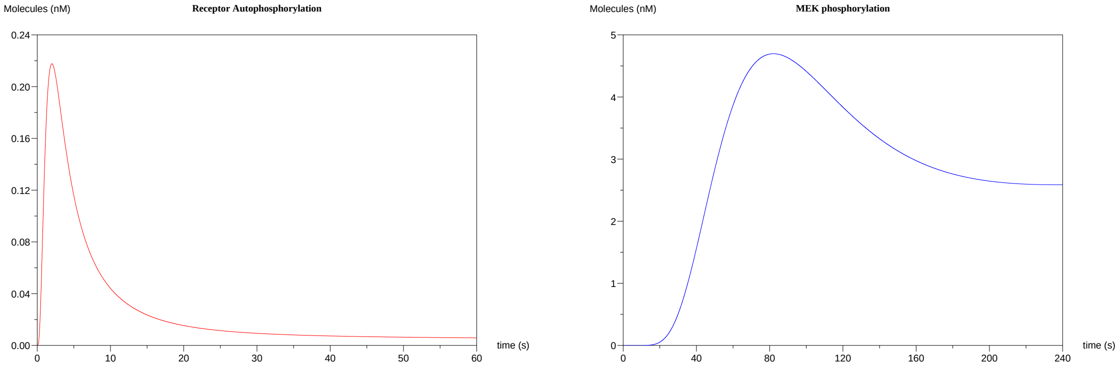

Figure 7.5.: Evolution of the number of autophosphorylated receptors, left, and the number of doubled phosphorylated MEK, right

Recall, that these models represent an experiment where the extracellular signal EGF varies over a range from 100 nM to 2000 nM.

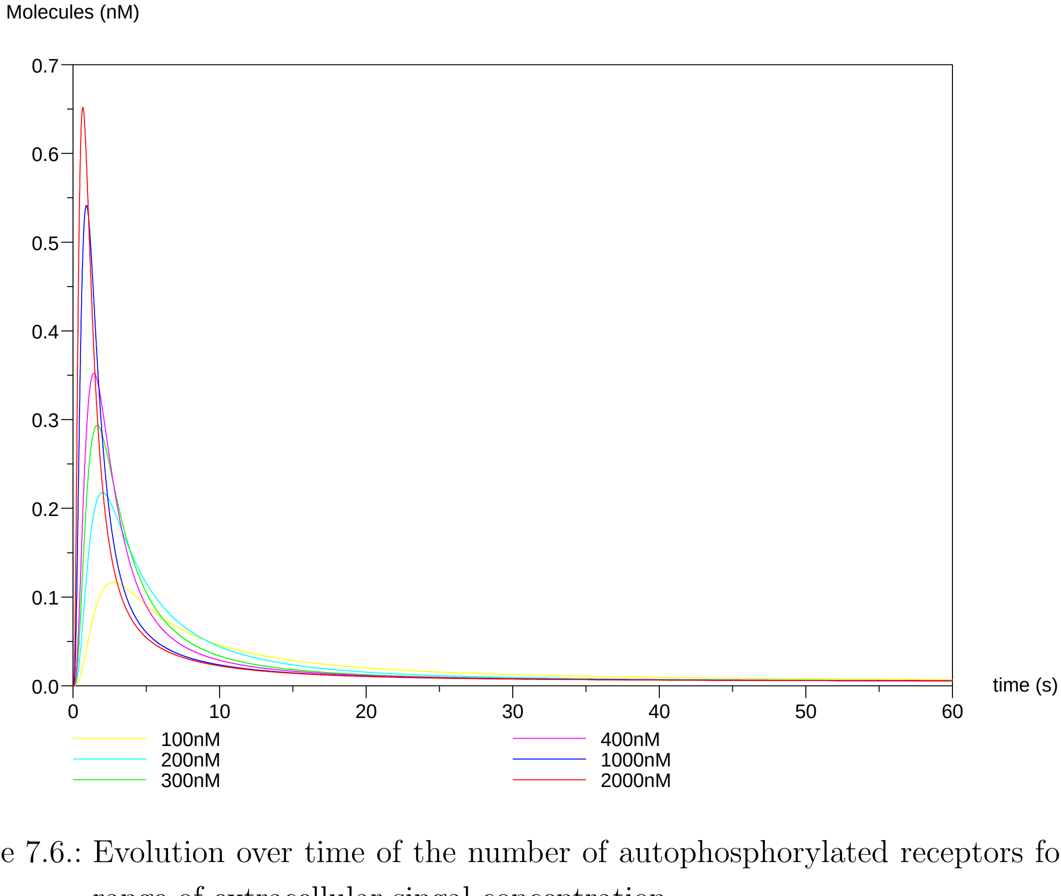

Figure 7.6.: Evolution over time of the number of autophosphorylated receptors for a range of extracellular singal concentration

In Figure 7.6 it can be seen that the receptor autophosphorylation is clearly concentration dependent showing different peaks for a differing number of signals in the environment. According to the variance in the receptor activation it is intuitive to expect different cell responses to different EGF concentrations. In what follows we show that, interestingly, this is not the case.

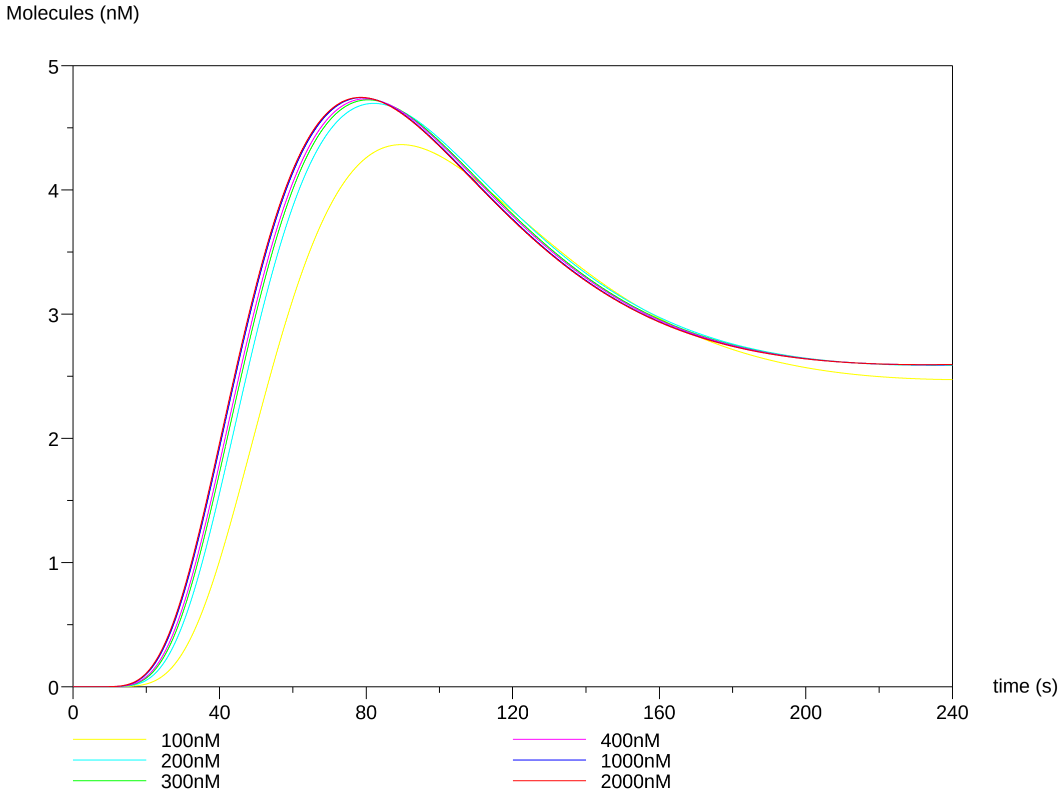

Figure 7.7.: Evolution over time of the number of double phosphorylated MEK for a range of extracellular singal concentration

Observe, in Figure 7.7 , that the number of doubled phosphorylated MEK does not depend on the number of signals in the environment. This shows the surprising robustness of the signalling cascade with regard to the number of extracellular EGF. The signal is either attenuated or amplified to obtain the same concentration of one of the most relevant kinases in the signalling cascade, MEK. Note that after 100 seconds, when the response is sustained, the lines representing the response to different external EGF concentrations are identical.

- **Effect of the number of receptor in the cell surface**

Now we analyse the effect on the dynamics of the signalling cascade on different numbers of receptors in the cell surface. More specifically, we will show the evolution over time of the number of double phosphorylated MEK in the models

$PSM_{EGFR}^{400,100}$ and $PSM_{EGFR}^{400,1000}$. Recall, that these two models represent an experiment where the number of receptors in the cell surface is changed from 100 nM to 1000 nM whereas the number of extracellular signals EGF is kept constant at 400 nM.

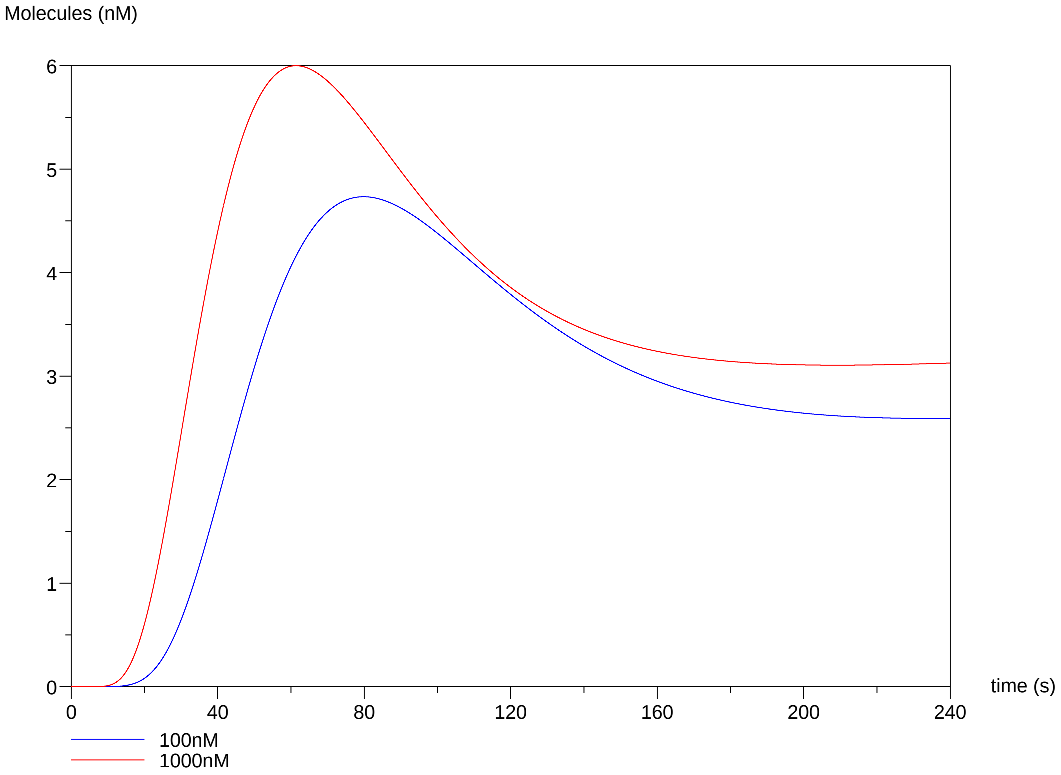

Figure 7.8.: Evolution over time of the number of double phosphorylated MEK for different number of receptors in the cell surface

Figure 7.8 shows the evolution of the number of doubled phosphorylated MEK when there is 100 nM and 1000 nM of receptors in the cell surface. Note that now the response is considerably different, the number of activated MEK being greater when there is an overexpression of receptors in the cell surface. As a consequence of this high number of activated MEK the cells will undergo an uncontrolled process of proliferation.

The key role played by the overexpression of EGFR on the uncontrolled growth of tumours has been reported before, making EGFR one of the main biological targets for the development of novel anticancer therapies.

## 7.4. Modelling FAS Induced Apoptosis

There are basically two mechanisms of cell death, *necrosis* and *apoptosis.* Necrosis is a form of cell death that usually occurs when cells are damaged by injury. A disruption of the cell membrane occurs and intracellular materials are released. In contrast to necrosis, apoptosis is carried out in an ordered sequence of events that culminates in the suicide of the cell, and without releasing intracellular materials from the dying cell.

The term *apoptosis* (also known as *programmed cell death*) was coined by Kerr, Wyllie and Currie [67] as a means of distinguishing a morphologically distinctive form of cell death which was associated with normal physiology.

Apoptosis occurs during organ development, plays an important role in cellular homeostasis [62], and is a cellular response to a *cellular insult* that starts a cascade of apoptotic signals, both intracellular and extracellular, which converge on the activation of a group of apoptotic–specific proteases called caspases. The apoptotic mechanism include condensation of cell contents, DNA fragmentation into nucleosomal fragments, nuclear membrane breakdown, and the formation of apoptotic bodies, small membrane–bound vesicles phagocytosed by neighboring cells [84]. Apoptosis protects the rest of the organism from a potentially harmful agent and dysregulation of apoptosis can contribute to the development of autoimmune diseases and cancers. Apoptosis can also be induced by anticancer drugs, group factor deprivation, and irradiation.

The family of proteases that mediates apoptosis is divided into two subgroups.

- The first group consists of caspase 8, caspase 9, and caspase 10, and they function as initiators of the cell death process.

- The second group contains caspase 3, caspase 6, and caspase 7, and they work as effectors. The other effector molecule in apoptosis is Apaf-1, which, together with cytochrome c, stimulates the processing of pro-caspase 9 to the mature enzyme.

The other regulators of apoptosis are the Bcl2 family members, divided into three subgroups based on their structure.

- Members of the first subgroup, represented by Bcl2 and Bcl-xL, have an anti-apoptotic function.

- The second subgroup, represented by Bax and Bak.

- The third subgroup, represented by Bid and Bad, are pro-apoptotic molecules.

Apoptotic death can be triggered by a wide variety of stimuli. Among the more studied death stimuli are DNA damage which in many cells leads to apoptotic death via a pathway dependent on p53, and the signalling pathways for FAS-induced apoptosis that was shown to be one of the most relevant processes for understanding and combating many forms of human diseases such as cancer, neurodegenerative diseases (Parkinson's disease, Alzheimer's disease, etc.), AIDS and ischemic stroke.

Fas (also called CD95 or APO–1) is a cell surface receptor protein with an extracellular region, one transmembrane domain, and an intracellular region. Fas belongs to the tumour necrosis factor/nerve growth factor (TNT/NGF) cytokine receptor family. Activation of Fas through binding to its ligands induces apoptosis in the Fas bearing cell. Fas induced–apoptosis starts from the Fas ligand binding to Fas receptors and ends in the fragmentation of genomic DNA, which is used as a hallmark of apoptosis.

Fas ligands usually exist as trimers and bind and activate their receptors by inducing receptor trimerisation. This creates a clustering of Fas that is necessary for signalling. In its intracellular region, Fas contains a conserved sequence called a *death domain*. Activated receptors recruit adaptor molecules (such as FADD, Fas–associating protein with death domain) which interacts with the death domain on the Fas receptor and recruit procaspase 8 to the receptor complex, where it undergoes autocatalytic activation cleaving and releasing active caspase 8 molecules intracellularly. Activated caspase 8 can activate caspase 3 through two different pathways that have been identified by Scaffidi et al. [114], and are referred to as type I (*death receptor pathway*) and type II (*mitochondrial pathway*), where caspases play a crucial role for both the initiation and execution apoptosis.

The pathways diverge after activation of initiator caspases and converge at the end by activating executor caspases. In the type I pathway, the initiator caspase (caspase 8) cleaves procaspase 3 directly and activates the executor caspase (caspase 3).

In the type II pathway, a more complicated cascade is activated involving the disruption of mitochondrial membrane potential and is mediated by Bcl2 family proteins that regulate the passage of small molecules which activate caspase cascades through the mitochondrial transition pore. More specifically (see Figure 7.9), caspase 8 cleaves Bid (Bcl2 interacting protein) and its COOH–terminal part translocates to mitochondria where it triggers cytochrome c release. The released cytochrome c bind to Apaf–1 (apoplectic protease activating factor) together with dATP and procaspase 9 and activate caspase 9. The caspase 9 cleaves procaspase 3 and activates caspase 3.

The executor caspase 3 cleaves DFF (DNA fragmentation factor) in a heterodimeric factor of DFF40 and DFF45. Cleaved DFF45 dissociates from DFF40, inducing oligomerisation of DFF40. The active DFF40 oligomer causes the internucleosomal DNA fragmentation.

Despite many molecular components of these apoptotic pathways having been iden-

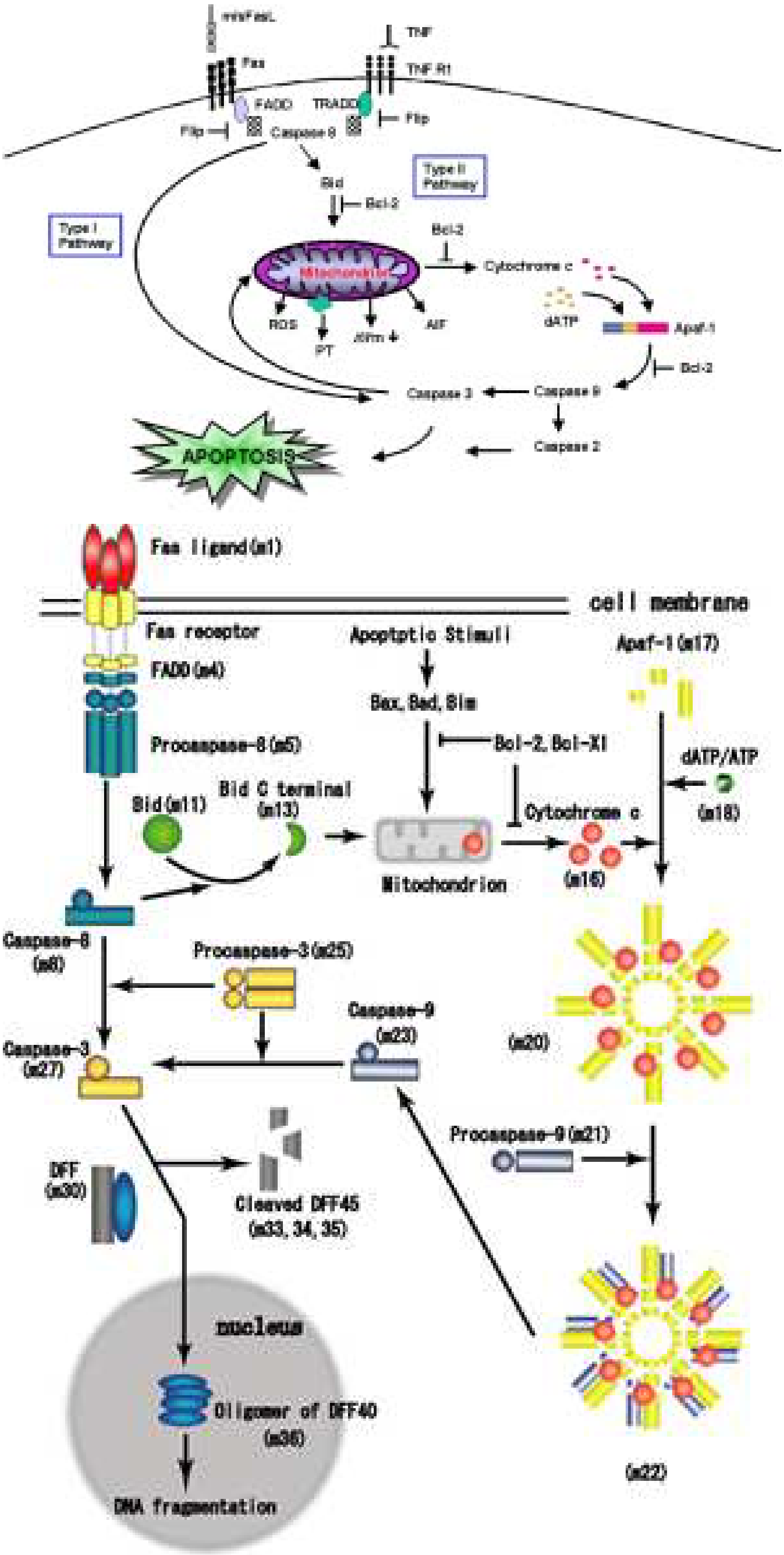

Figure 7.9.: FAS signalling pathways, from [2, 84]

tified, a better understanding of how they work together in a consistent network is necessary. A way to understand complex biological processes, in general, and the complex signalling behaviour of these pathways, in particular, is achieved by modelling them in a computational framework and simulating them in electronic computers.

In [55] the two pathways activated by FAS starting with the stimulation of FASL (FAS ligand) until the activation of the effector caspase 3, have been modelled using ordinary differential equations in which biochemical reactions were used to describe molecular interactions. Here we present an alternative modelling approach to the FAS induced apoptosis based in P systems.

### 7.4.1. A P System Specification of the FAS Induced Apoptosis

In this section we present a P system specification of the FAS induced apoptosis system described previously. The specification principles in this section will follow those used in the case of the EGFR signalling cascade in section 7.3.

- **Specification of the relevant regions in the FAS induced apoptosis system:**

  Similar to the case of the EGFR signalling cascade, membranes play a key role in the Fas induced apoptosis system. The relevant regions in the Fas induced apoptosis system include those used in the EGFR signalling cascade, namely, the environment, the cell surface and the cytoplasm. Nevertheless, in this case there is an extra relevant region, the mitochondria.

  More specifically the membrane structure of our P system specification for the Fas induced apopotosis system consists of the following four membranes:

  1. The environment where the signal FASL appear. This membrane is identified with the number 1 and will be labelled $e$. It constitutes the root of the membrane structure describing the compartments involved in the system.
  2. The cell surface where the receptor FAS is located and where the assembly of the the complexes involved in signalling takes place. This membrane is identified with the number 2 and will be labelled $s$. This membrane will be embedded inside the previous one to represent the fact that the environment surrounds the cell surface.
  3. The cytoplasm where the signalling pathways take place. This membrane is identified with the number 3 and will be labelled $c$. This membrane will be located inside the previous membrane to represent the fact that the cell surface wraps the cytoplasm.

4. Finally, a region is used to describe all mitochondria in the cell. All the molecules associated with the mitochondria will be placed in this region identified with the number 4 and labelled $m$. This last membrane will be embedded inside the membrane representing the cytoplasm.

A Venn diagram representation of the membrane structure can be seen in Figure 7.10.

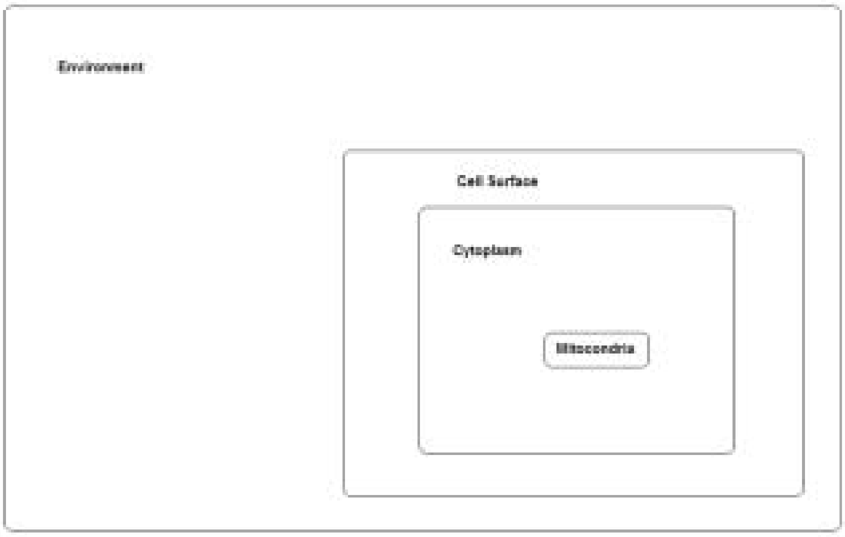

Figure 7.10.: Membrane structure in the P System Specification of the Fas induced apoptosis system

- **Specification of the molecular entities in the FAS induced apoptosis system:**

Our model consist of 53 proteins and complexes of proteins. All of them are represented by single objects collected in the alphabet $O$, presented below:

$$
\begin{aligned}
O = \{&\text{FASL, FAS, FASC, FADD, FASC-FADD, FASC-FADD}_2\text{,,}\\
&\text{FASC-FADD}_3\text{, FASC-FADD}_2\text{-CASP8, FASC-FADD}_3\text{-CASP8,}\\
&\text{FASC-FADD}_2\text{-FLIP, FASC-FADD}_3\text{-FLIP,}\\
&\text{FASC-FADD}_2\text{-CASP8}_2\text{, FASC-FADD}_3\text{-CASP8}_2\text{,}\\
&\text{FASC-FADD}_2\text{-CASP8-FLIP, FASC-FADD}_3\text{-CASP8-FLIP,}\\
&\text{FASC-FADD}_2\text{-FLIP}_2\text{, FASC-FADD}_3\text{-FLIP}_2\text{,}\\
&\text{FASC-FADD-CASP8, FASC-FADD-FLIP, CASP8, FLIP,}\\
&\text{FASC-FADD}_3\text{-CASP8}_3\text{, FASC-FADD}_3\text{-CASP8}_2\text{-FLIP,}\\
&\text{FASC-FADD}_3\text{-CASP8-FLIP}_2\text{, FASC-FADD}_3\text{-FLIP}_3\text{,}\\
&\text{CASP8}_2^{P41}\text{, CASP8}_2^*\text{, CASP3, CASP8}_2^*\text{-CASP3,}\\
&\text{CASP3}^*\text{, CASP8}_2^*\text{-Bid, tBid, Bid, Bax, tBid-Bax,}\\
&\text{tBid-Bax}_2\text{, Smac, Smac}^*\text{, Cyto.c, Cyto.c}^*\text{, XIAP,}\\
&\text{Smac}^*\text{-XIAP, Apaf, Cyto.c}^*\text{-Apaf-ATP, CASP9,}\\
&\text{Cyto.c}^*\text{-Apaf-ATP-CASP9, Cyto.c}^*\text{-Apaf-ATP-CASP9}_2\text{,}\\
&\text{CASP9}^*\text{, CASP9}^*\text{-CASP3, CASP9-XIAP, CASP3}^*\text{-XIAP,}\\
&\text{Bcl2, Bcl2-Bax}\}
\end{aligned}
\tag{7.2}
$$

The most important molecular entities and their corresponding objects are presented in Figure 7.11.

| Object | Protein or Complex |
|---|---|
| FAS | Fas protein |
| FASL | Fas Ligand |
| FADD | Fas–associating protein with death domain |
| FLIP | Fas Ligand Inhibitory Protein |
| CASP3, CASP8, CASP9 | Caspases 3, 8 and 9 |
| Bcl2 | B-cell leukemia/lymphoma-2 |
| Bid, tBid | Domain death agonist and its truncated form |
| Bax | BCL-2-associated X protein |
| Apaf | Apoptotic protease activating factor |
| Smac | Second mitochondria–derived activator of caspase |
| XIAP | X–linked inhibitor of apoptosis protein |

Figure 7.11.: Molecular entities in the FAS induced apoptosis system

- **Specification of the molecular interactions in the FAS induced apoptosis system:**

  The molecular interactions in the FAS induced apoptosis systems are similar to those presented in the P system specification of the EGFR signalling cascade. Binding and debinding rules describe the binding and debinding of the signal FASL to its receptor FAS. Complex formation rules are used to specify the assembly in the cell surface of the complexes which triggers the cytoplasmic signalling pathways. The initiation of these signalling pathways is specified using recruitment and releasing rules involving the complexes in the cell surface and certain cytoplasmic proteins. The signalling pathways consists of complex formation and dissociation rules that specify the transfer of the signal to the target proteins.

All the rules of our P system specification of the FAS induced apoptosis system are presented in Figures 7.12 7.13 7.14 7.15. The constants associated with them are computed from those presented in Figure 7.16.

Regarding the specification of the molecular interactions in the FAS induced apoptosis systems it is known that Bcl2 blocks the mitochondrial pathway. Nevertheless, it is not clear that the mechanism through which Bcl2 can block the pathway is of type II.

In [85] and [127] four different mechanisms to block the mitochondrial are suggested:

(a) Bcl2 might bind to Bax

(b) Bcl2 might bind to Bid

(c) Bcl2 might bind to tBid

(d) Bcl2 might bind to both Bax and tBid

All these possible alternatives are presented in Figure 7.15.

Summing up our P system specification of the FAS induced apoptosis system consists of the following construct:

$$\mathbf{\Pi}_{FAS} = (O, \{e, s, c, m\}, \mu, M_1, M_2, M_3, M_4, \mathcal{R}_e, \mathcal{R}_s, \mathcal{R}_c, \mathcal{R}_m)$$

where:

- The alphabet $O$ collects all the objects representing the 53 proteins and complexes of proteins involved in the system, see (7.2).
- The set of labels, $L = \{e, s, c, m\}$, is used to identify the different compartment types where the signalling transduction takes place, namely, the environment $e$, the cell surface $s$, the cytoplasm $c$ and the mitochondria $m$.

| Rule | Constant |
|---|---|
| $r_1 : FASL\ [\ FAS\ ]_s \xrightarrow{c_1} [\ FASC\ ]_s$ | $k_{1f}$ |
| $r_2 : [\ FASC\ ]_s \xrightarrow{c_2} FASL\ [\ FASC\ ]_s$ | $k_{1r}$ |
| $r_3 : FASC\ [\ FADD\ ]_c \xrightarrow{c_3} FASC : FADD\ [\ ]_c$ | $k_{2f}$ |
| $r_4 : FASC\text{-}FADD\ [\ ]_c \xrightarrow{c_4} FASC\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_5 : FASC\text{-}FADD\ [\ FADD\ ]_c \xrightarrow{c_5} FASC\text{-}FADD_2\ [\ ]_c$ | $k_{2f}$ |
| $r_6 : FASC\text{-}FADD_2\ [\ ]_c \xrightarrow{c_6} FASC\text{-}FADD\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_7 : FASC\text{-}FADD_2\ [\ FADD\ ]_c \xrightarrow{c_7} FASC\text{-}FADD_3\ [\ ]_c$ | $k_{2f}$ |
| $r_8 : FASC\text{-}FADD_3\ [\ ]_c \xrightarrow{c_8} FASC\text{-}FADD_2\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_9 : FASC\text{-}FADD_2\text{-}CASP8\ [\ FADD\ ]_c \longrightarrow FASC\text{-}FADD_3\text{-}CASP8\ [\ ]_c$ | $k_{2f}$ |
| $r_{10} : FASC\text{-}FADD_3\text{-}CASP8\ [\ ]_c \xrightarrow{c_{10}} FASC\text{-}FADD_2\text{-}CASP8\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{11} : FASC\text{-}FADD_2\text{-}FLIP\ [\ FADD\ ]_c \xrightarrow{c_{11}} FASC\text{-}FADD_3\text{-}FLIP\ [\ ]_c$ | $k_{2f}$ |
| $r_{12} : FASC\text{-}FADD_3\text{-}FLIP\ [\ ]_c \xrightarrow{c_{12}} FASC\text{-}FADD_2\text{-}FLIP\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{13} : FASC\text{-}FADD_2\text{-}CASP8_2\ [\ FADD\ ]_c \xrightarrow{c_{13}} FASC\text{-}FADD_3\text{-}CASP8_2\ [\ ]_c$ | $k_{2f}$ |
| $r_{14} : FASC\text{-}FADD_3\text{-}CASP8_2\ [\ ]_c \xrightarrow{c_{14}} FASC\text{-}FADD_2\text{-}CASP8_2\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{15} : FASC\text{-}FADD_2\text{-}CASP8\text{-}FLIP\ [\ FADD\ ]_c \xrightarrow{c_{15}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ ]_c$ | $k_{2f}$ |
| $r_{16} : FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ ]_c \xrightarrow{c_{16}} FASC\text{-}FADD_2\text{-}CASP8\text{-}FLIP\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{17} : FASC\text{-}FADD_2\text{-}FLIP_2\ [\ FADD\ ]_c \xrightarrow{c_{17}} FASC\text{-}FADD_3\text{-}FLIP_2\ [\ ]_c$ | $k_{2f}$ |
| $r_{18} : FASC\text{-}FADD_3\text{-}FLIP_2\ [\ ]_c \xrightarrow{c_{18}} FASC\text{-}FADD_2\text{-}FLIP_2\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{19} : FASC\text{-}FADD\text{-}CASP8\ [\ FADD\ ]_c \xrightarrow{c_{19}} FASC\text{-}FADD_2\text{-}CASP8\ [\ ]_c$ | $k_{2f}$ |
| $r_{20} : FASC\text{-}FADD_2\text{-}CASP8\ [\ ]_c \xrightarrow{c_{20}} FASC\text{-}FADD\text{-}CASP8\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{21} : FASC\text{-}FADD\text{-}FLIP\ [\ FADD\ ]_c \xrightarrow{c_{21}} FASC\text{-}FADD_2\text{-}FLIP\ [\ ]_c$ | $k_{2f}$ |
| $r_{22} : FASC\text{-}FADD_2\text{-}FLIP\ [\ ]_c \xrightarrow{c_{22}} FASC\text{-}FADD\text{-}FLIP\ [\ FADD\ ]_c$ | $k_{2r}$ |
| $r_{23} : FASC\text{-}FADD_3\ [\ CASP8\ ]_c \xrightarrow{c_{23}} FASC\text{-}FADD_3\text{-}CASP8\ [\ ]_c$ | $k_{2f}$ |
| $r_{24} : FASC\text{-}FADD_3\text{-}CASP8\ [\ ]_c \xrightarrow{c_{24}} FASC\text{-}FADD_3\ [\ CASP8\ ]_c$ | $k_{2r}$ |
| $r_{25} : FASC\text{-}FADD_3\ [\ FLIP\ ]_c \xrightarrow{c_{25}} FASC\text{-}FADD_3\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{26} : FASC\text{-}FADD_3\text{-}FLIP\ [\ ]_c \xrightarrow{c_{26}} FASC\text{-}FADD_3\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{27} : FASC\text{-}FADD_3\text{-}CASP8\ [\ CASP8\ ]_c \xrightarrow{c_{27}} FASC\text{-}FADD_3\text{-}CASP8_2\ [\ ]_c$ | $k_{3f}$ |
| $r_{28} : FASC\text{-}FADD_3\text{-}CASP8_2\ [\ ]_c \xrightarrow{c_{28}} FASC\text{-}FADD_3\text{-}CASP8\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{29} : FASC\text{-}FADD_3\text{-}CASP8\ [\ FLIP\ ]_c \xrightarrow{c_{29}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{30} : FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ ]_c \xrightarrow{c_{30}} FASC\text{-}FADD_3\text{-}CASP8\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{31} : FASC\text{-}FADD_3\text{-}FLIP\ [\ CASP8\ ]_c \xrightarrow{c_{31}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{32} : FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ ]_c \xrightarrow{c_{32}} FASC\text{-}FADD_3\text{-}FLIP\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{33} : FASC\text{-}FADD_3\text{-}FLIP\ [\ FLIP\ ]_c \xrightarrow{c_{33}} FASC\text{-}FADD_3\text{-}FLIP_2\ [\ ]_c$ | $k_{3f}$ |
| $r_{34} : FASC\text{-}FADD_3\text{-}FLIP_2\ [\ ]_c \xrightarrow{c_{34}} FASC\text{-}FADD_3\text{-}FLIP\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{35} : FASC\text{-}FADD_3\text{-}CASP8_2\ [\ CASP8\ ]_c \xrightarrow{c_{35}} FASC\text{-}FADD_3\text{-}CASP8_3\ [\ ]_c$ | $k_{3f}$ |

Figure 7.12.: Rewriting rule for the P system specification of the FAS induced apoptosis system

| Rule | Constant |
|---|---|
| $r_{36}: FASC\text{-}FADD_3\text{-}CASP8_3\ [\ ]_c \xrightarrow{c_{36}} FASC\text{-}FADD_3\text{-}CASP8_2\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{37}: FASC\text{-}FADD_3\text{-}CASP8_2\ [\ FLIP\ ]_c \xrightarrow{c_{37}} FASC\text{-}FADD_3\text{-}CASP8_2\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{38}: FASC\text{-}FADD_3\text{-}CASP8_2\text{-}FLIP\ [\ ]_c \xrightarrow{c_{38}} FASC\text{-}FADD_3\text{-}CASP8_2\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{39}: FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ CASP8\ ]_c \xrightarrow{c_{39}} FASC\text{-}FADD_3\text{-}CASP8_2\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{40}: FASC\text{-}FADD_3\text{-}CASP8_2\text{-}FLIP\ [\ ]_c \xrightarrow{c_{40}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{41}: FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ FLIP\ ]_c \xrightarrow{c_{41}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP_2\ [\ ]_c$ | $k_{3f}$ |
| $r_{42}: FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP_2\ [\ ]_c \xrightarrow{c_{42}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{43}: FASC\text{-}FADD_3\text{-}FLIP_2\ [\ CASP8\ ]_c \xrightarrow{c_{43}} FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP_2\ [\ ]_c$ | $k_{3f}$ |
| $r_{44}: FASC\text{-}FADD_3\text{-}CASP8\text{-}FLIP_2\ [\ ]_c \xrightarrow{c_{44}} FASC\text{-}FADD_3\text{-}FLIP_2\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{45}: FASC\text{-}FADD_3\text{-}FLIP_2\ [\ FLIP\ ]_c \xrightarrow{c_{45}} FASC\text{-}FADD_3\text{-}FLIP_3\ [\ ]_c$ | $k_{3f}$ |
| $r_{46}: FASC\text{-}FADD_3\text{-}FLIP_3\ [\ ]_c \xrightarrow{c_{46}} FASC\text{-}FADD_3\text{-}FLIP_2\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{47}: FASC\text{-}FADD_2\ [\ CASP8\ ]_c \xrightarrow{c_{47}} FASC\text{-}FADD_2\text{-}CASP8\ [\ ]_c$ | $k_{3f}$ |
| $r_{48}: FASC\text{-}FADD_2\text{-}CASP8\ [\ ]_c \xrightarrow{c_{48}} FASC\text{-}FADD_2\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{49}: FASC\text{-}FADD_2\ [\ FLIP\ ]_c \xrightarrow{c_{49}} FASC\text{-}FADD_2\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{50}: FASC\text{-}FADD_2\text{-}FLIP\ [\ ]_c \xrightarrow{c_{50}} FASC\text{-}FADD_2\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{51}: FASC\text{-}FADD_2\text{-}CASP8\ [\ CASP8\ ]_c \xrightarrow{c_{51}} FASC\text{-}FADD_2\text{-}CASP8_2\ [\ ]_c$ | $k_{3f}$ |
| $r_{52}: FASC\text{-}FADD_2\text{-}CASP8_2\ [\ ]_c \xrightarrow{c_{52}} FASC\text{-}FADD_2\text{-}CASP8\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{53}: FASC\text{-}FADD_2\text{-}CASP8\ [\ FLIP\ ]_c \xrightarrow{c_{53}} FASC\text{-}FADD_2\text{-}CASP8 : FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{54}: FASC\text{-}FADD_2\text{-}CASP8\text{-}FLIP\ [\ ]_c \xrightarrow{c_{54}} FASC\text{-}FADD_2\text{-}CASP8\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{55}: FASC\text{-}FADD_2\text{-}FLIP\ [\ CASP8\ ]_c \xrightarrow{c_{55}} FASC\text{-}FADD_2\text{-}CASP8\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{56}: FASC\text{-}FADD_2\text{-}CASP8\text{-}FLIP\ [\ ]_c \xrightarrow{c_{56}} FASC\text{-}FADD_2\text{-}FLIP\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{57}: FASC\text{-}FADD_2\text{-}FLIP\ [\ FLIP\ ]_c \xrightarrow{c_{57}} FASC\text{-}FADD_2\text{-}FLIP_2\ [\ ]_c$ | $k_{3f}$ |
| $r_{58}: FASC\text{-}FADD_2\text{-}FLIP_2\ [\ ]_c \xrightarrow{c_{58}} FASC\text{-}FADD_2\text{-}FLIP\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{59}: FASC\text{-}FADD\ [\ CASP8\ ]_c \xrightarrow{c_{59}} FASC\text{-}FADD\text{-}CASP8\ [\ ]_c$ | $k_{3f}$ |
| $r_{60}: FASC\text{-}FADD\text{-}CASP8\ [\ ]_c \xrightarrow{c_{60}} FASC\text{-}FADD\ [\ CASP8\ ]_c$ | $k_{3r}$ |
| $r_{61}: FASC\text{-}FADD\ [\ FLIP\ ]_c \xrightarrow{c_{61}} FASC\text{-}FADD\text{-}FLIP\ [\ ]_c$ | $k_{3f}$ |
| $r_{62}: FASC\text{-}FADD\text{-}FLIP\ [\ ]_c \xrightarrow{c_{62}} FASC\text{-}FADD\ [\ FLIP\ ]_c$ | $k_{3r}$ |
| $r_{63}: FASC\text{-}FADD_2\text{-}CASP8_2\ [\ ]_c \xrightarrow{c_{63}} FASC\text{-}FADD_2\ [\ CASP8_2^{P41}\ ]_c$ | $k_4$ |
| $r_{64}: FASC\text{-}FADD_3\text{-}CASP8_3\ [\ ]_c \xrightarrow{c_{64}} FASC\text{-}FADD_3\text{-}CASP8\ [\ CASP8_2^{P41}\ ]_c$ | $k_4$ |
| $r_{65}: FASC\text{-}FADD_3\text{-}CASP8_2\text{-}FLIP\ [\ ]_c \xrightarrow{c_{65}} FASC\text{-}FADD_3\text{-}FLIP\ [\ CASP8_2^{P41}\ ]_c$ | $k_4$ |
| $r_{66}: FASC\text{-}FADD_3\text{-}CASP8_2\ [\ ]_c \xrightarrow{c_{66}} FASC\text{-}FADD_3\ [\ CASP8_2^{P41}\ ]_c$ | $k_4$ |
| $r_{67}:\ [\ CASP8_2^{P41}\ ]_c \xrightarrow{c_{67}}\ [\ CASP8_2^*\ ]_c$ | $k_5$ |
| $r_{68}:\ [\ CASP8_2^* + CASP3\ ]_c \xrightarrow{c_{68}}\ [\ CASP8_2^*\text{-}CASP3\ ]_c$ | $k_{6f}$ |
| $r_{69}:\ [\ CASP8_2^*\text{-}CASP3\ ]_c \xrightarrow{c_{69}}\ [\ CASP8_2^* + CASP3\ ]_c$ | $k_{6r}$ |
| $r_{70}:\ [\ CASP8_2^* + CASP3^*\ ]_c \xrightarrow{c_{70}}\ [\ CASP8_2^*\text{-}CASP3\ ]_c$ | $k_7$ |

Figure 7.13.: Rewriting rule for the P system specification of the FAS induced apoptosis system

| Rule | Constant |
|---|---|
| $r_{71}:\ [\,CASP8_2^* + Bid\,]_c \xrightarrow{c_{71}} [\,CASP8_2^*\text{-}Bid\,]_c$ | $k_{8f}$ |
| $r_{72}:\ [\,CASP8_2^*\text{-}Bid\,]_c \xrightarrow{c_{72}} [\,CASP8_2^* + Bid\,]_c$ | $k_{8r}$ |
| $r_{73}:\ [\,CASP8_2^* + tBid\,]_c \xrightarrow{c_{73}} [\,CASP8_2^*\text{-}Bid\,]_c$ | $k_7$ |
| $r_{74}:\ [\,tBid + Bax\,]_c \xrightarrow{c_{74}} [\,tBid\text{-}Bax\,]_c$ | $k_{9f}$ |
| $r_{75}:\ [\,tBid\text{-}Bax\,]_c \xrightarrow{c_{75}} [\,tBid + Bax\,]_c$ | $k_{9r}$ |
| $r_{76}:\ [\,tBid\text{-}Bax + Bax\,]_c \xrightarrow{c_{76}} [\,tBid\text{-}Bax_2\,]_c$ | $k_{9f}$ |
| $r_{77}:\ [\,tBid\text{-}Bax_2\,]_c \xrightarrow{c_{77}} [\,tBid\text{-}Bax + Bax\,]_c$ | $k_{9r}$ |
| $r_{78}: tBid\text{-}Bax_2\ [\,Smac\,]_m \xrightarrow{c_{78}} Smac^*\ [\ \ ]_m$ | $k_{10}$ |
| $r_{79}: tBid\text{-}Bax_2\ [\,Cyto.c\,]_m \xrightarrow{c_{79}} Cyto.c^*\ [\ \ ]_m$ | $k_{10}$ |
| $r_{80}:\ [\,Smac^* + XIAP\,]_c \xrightarrow{c_{80}} [\,Smac^*\text{-}XIAP\,]_c$ | $k_{11f}$ |
| $r_{81}:\ [\,Smac^*\text{-}XIAP\,]_c \xrightarrow{c_{81}} [\,Smac^* + XIAP\,]_c$ | $k_{11r}$ |
| $r_{82}:\ [\,Cyto.c^* + Apaf\,]_c \xrightarrow{c_{82}} [\,Cyto.c^*\text{-}Apaf\text{-}ATP\,]_c$ | $k_{12f}$ |
| $r_{83}:\ [\,Cyto.c^*\text{-}Apaf\text{-}ATP\,]_c \xrightarrow{c_{83}} [\,Cyto.c^* + Apaf\,]_c$ | $k_{12r}$ |
| $r_{84}:\ [\,Cyto.c^*\text{-}Apaf\text{-}ATP + CASP9\,]_c \xrightarrow{c_{84}} [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9\,]_c$ | $k_{13f}$ |
| $r_{85}:\ [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9\,]_c \xrightarrow{c_{85}} [\,Cyto.c^*\text{-}Apaf\text{-}ATP + CASP9\,]_c$ | $k_{13r}$ |
| $r_{86}:\ [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9 + CASP9\,]_c \xrightarrow{c_{86}} [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9_2\,]_c$ | $k_{14f}$ |
| $r_{87}:\ [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9_2\,]_c \xrightarrow{c_{87}} [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9 + CASP9\,]_c$ | $k_{14r}$ |
| $r_{88}:\ [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9_2\,]_c \xrightarrow{c_{88}} [\,Cyto.c^*\text{-}Apaf\text{-}ATP\text{-}CASP9 + CASP9^*\,]_c$ | $k_{15}$ |
| $r_{89}:\ [\,CASP9^* + CASP3\,]_c \xrightarrow{c_{89}} [\,CASP9^*\text{-}CASP3\,]_c$ | $k_{16f}$ |
| $r_{90}:\ [\,CASP9^*\text{-}CASP3\,]_c \xrightarrow{c_{90}} [\,CASP9^* + CASP3\,]_c$ | $k_{16r}$ |
| $r_{91}:\ [\,CASP9^*\text{-}CASP3\,]_c \xrightarrow{c_{91}} [\,CASP9^* + CASP3^*\,]_c$ | $k_{17}$ |
| $r_{92}:\ [\,CASP9 + XIAP\,]_c \xrightarrow{c_{92}} [\,CASP9\text{-}XIAP\,]_c$ | $k_{18f}$ |
| $r_{93}:\ [\,CASP9\text{-}XIAP\,]_c \xrightarrow{c_{93}} [\,CASP9 + XIAP\,]_c$ | $k_{18r}$ |
| $r_{94}:\ [\,CASP3^* + XIAP\,]_c \xrightarrow{c_{94}} [\,CASP3^*\text{-}XIAP\,]_c$ | $k_{19f}$ |
| $r_{95}:\ [\,CASP3^*\text{-}XIAP\,]_c \xrightarrow{c_{95}} [\,CASP3^* + XIAP\,]_c$ | $k_{19r}$ |

Figure 7.14.: Rewriting rule for the P system specification of the FAS induced apoptosis system

| Rule | Constant |
|---|---|
| $r_{96} : Bax\ [\ Bcl2\ ]_m \xrightarrow{c_{96}} [\ Bcl2\text{-}Bax\ ]_m$ | $k_{20f}$ |
| $r_{97} : [\ Bcl2\text{-}Bax\ ]_m \xrightarrow{c_{97}} Bax\ [\ Bcl2\ ]_m$ | $k_{20r}$ |
| $r_{96'} : Bid\ [\ Bcl2\ ]_m \xrightarrow{c_{96'}} [\ Bcl2\text{-}Bid\ ]_m$ | $k_{20f}$ |
| $r_{97'} : [\ Bcl2\text{-}Bid\ ]_m \xrightarrow{c_{97'}} Bid\ [\ Bcl2\ ]_m$ | $k_{20r}$ |
| $r_{96''} : tBid\ [\ Bcl2\ ]_m \xrightarrow{c_{96''}} [\ Bcl2\text{-}tBid\ ]_m$ | $k_{20f}$ |
| $r_{97''} : [\ Bcl2\text{-}tBid\ ]_m \xrightarrow{c_{97''}} tBid\ [\ Bcl2\ ]_m$ | $k_{20r}$ |

Figure 7.15.: Rewriting rule for the P system specification of the FAS induced apoptosis system

| | | | | | |
|---|---|---|---|---|---|
| $k_{1f}$ | $=$ | $9.09E-05\ nM^{-1}s^{-1}$ | $k_{11r}$ | $=$ | $2.21E-03\ s^{-1}$ |
| $k_{1r}$ | $=$ | $1.00E-04\ s^{-1}$ | $k_{12f}$ | $=$ | $2.78E-07\ nM^{-1}s^{-1}$ |
| $k_{2f}$ | $=$ | $5.00E-04\ nM^{-1}s^{-1}$ | $k_{12r}$ | $=$ | $5.70E-03\ s^{-1}$ |
| $k_{2r}$ | $=$ | $0.2\ s^{-1}$ | $k_{13f}$ | $=$ | $2.84E-04\ nM^{-1}s^{-1}$ |
| $k_{3f}$ | $=$ | $3.50E-03\ nM^{-1}s^{-1}$ | $k_{13r}$ | $=$ | $0.07493\ s^{-1}$ |
| $k_{3r}$ | $=$ | $0.018\ s^{-1}$ | $k_{14f}$ | $=$ | $4.41E-04\ nM^{-1}s^{-1}$ |
| $k_4$ | $=$ | $0.3\ s^{-1}$ | $k_{14r}$ | $=$ | $0.1\ s^{-1}$ |
| $k_5$ | $=$ | $0.1\ s^{-1}$ | $k_{15}$ | $=$ | $0.7\ s^{-1}$ |
| $k_{6f}$ | $=$ | $1.00E-05\ nM^{-1}s^{-1}$ | $k_{16f}$ | $=$ | $1.96E-05\ nM^{-1}s^{-1}$ |
| $k_{6r}$ | $=$ | $0.06\ s^{-1}$ | $k_{16r}$ | $=$ | $0.05707\ s^{-1}$ |
| $k_7$ | $=$ | $0.1\ s^{-1}$ | $k_{17}$ | $=$ | $4.8\ s^{-1}$ |
| $k_{8f}$ | $=$ | $5.00E-03\ nM^{-1}s^{-1}$ | $k_{18f}$ | $=$ | $1.06E-04\ nM^{-1}s^{-1}$ |
| $k_{8r}$ | $=$ | $0.005\ s^{-1}$ | $k_{18r}$ | $=$ | $1.00E-03\ s^{-1}$ |
| $k_{9f}$ | $=$ | $2.00E-04\ nM^{-1}s^{-1}$ | $k_{19f}$ | $=$ | $2.47E-03\ nM^{-1}s^{-1}$ |
| $k_{9r}$ | $=$ | $0.02\ s^{-1}$ | $k_{19r}$ | $=$ | $2.40E-03\ s^{-1}$ |
| $k_{10}$ | $=$ | $1.00E-03\ nM^{-1}s^{-1}$ | $k_{20f}$ | $=$ | $2.00E-03\ nM^{-1}s^{-1}$ |
| $k_{11f}$ | $=$ | $7.00E-03\ nM^{-1}s^{-1}$ | $k_{20r}$ | $=$ | $0.02\ s^{-1}$ |

Figure 7.16.: Kinetic constants of the FAS induced apoptosis systems

- The membrane structure, $\mu$, consists of four membranes identifying the four relevant regions in the system, namely, the environment, the cell surface, the cytoplasm and the mitochondria. A Venn diagram representation of the membrane structure in our P system specification $\Pi_{FAS}$ is depicted in Figure 7.10.

- The initial multisets $M_1$, $M_2$, $M_3$ and $M_4$ are part of the parameters of our P system specification $\Pi_{FAS}$. They associate the label $e$ with membrane 1 which represents the environment, the label $s$ with membrane 2 which represents the cell surface, the label $c$ with membrane 3 which represents the cytoplasm and the label $m$ with membrane 4 which represents the mitochondria.

- The set of rewriting rules on multisets of objects and strings $R_e, R_s$, $R_c$ and $R_m$ are associated with the compartments representing the environment, cell surface, cytoplasm and mitochondria respectively. These rules describe the molecular interactions that take place in the specific compartment they are associated with. Next we enumerate the rules associated with each compartment.

  - $R_e = \{r_1\}$
  - $R_s = \{r_2, r_4, r_6, r_8, r_{10}, r_{11}, r_{12}, r_{14}, r_{16}, r_{17}, r_{18}, r_{20}, r_{22}, r_{24}, r_{26}, r_{28}, r_{30}, r_{32}, r_{34}, r_{36}, r_{38}, r_{40}, r_{42}, r_{44}, r_{46}, r_{48}, r_{50}, r_{52}, r_{54}, r_{56}, r_{58}, r_{60}, r_{62}, r_{63}, r_{64}, r_{65}, r_{66}\}$
  - $R_c = \{r_3, r_5, r_7, r_9, r_{11}, r_{13}, r_{15}, r_{17}, r_{19}, r_{21}, r_{23}, r_{25}, r_{27}, r_{29}, r_{31}, r_{33}, r_{35}, r_{37}, r_{39}, r_{41}, r_{43}, r_{45}, r_{47}, r_{49}, r_{51}, r_{53}, r_{55}, r_{57}, r_{59}, r_{61}, r_{67}, r_{68}, r_{69}, r_{70}, r_{71}, r_{72}, r_{73}, r_{74}, r_{75}, r_{76}, r_{77}, r_{78}, r_{79}, r_{80}, r_{81}, r_{82}, r_{83}, r_{84}, r_{85}, r_{86}, r_{87}, r_{88}, r_{89}, r_{90}, r_{91}, r_{92}, r_{93}, r_{94}, r_{95}, r_{96}, r_{96'}, r_{96''}\}$
  - $R_m = \{r_{97}, r_{97'}, r_{97''}\}$

### 7.4.2. Analysis of the FAS Induced Apoptosis using P systems Models

In a similar manner as in the previous cases once we have developed our P system specification of the Fas induced apoptosis system, $\Pi_{FAS}$, it is necessary to identify the parameters associated with it and give them specific values that allow us to study or analyse different properties of the system.

In all the previous cases the analysis of the systems refer to the study of their behaviour for different initial conditions represented in the initial multisets. In the case of our analysis of the FAS induced apoptosis system in addition to studying the behaviour for different initial coditions we will analyse the effect of different mechanisms, described using rewriting rules, on the signalling cascade.

First we study the activation of one of the major proteins in the pathway, caspase 3. We compare our results obtained using the Deterministic Waiting Times algorith to experimental data and to the results obtained using ODEs in [55]. The parameters corresponding to the initial multisets are computed from the initial condition of the system of differential equations presented in [55].

$$\begin{aligned}
M_1 &= (e, FASL^{12500}, \lambda) \\
M_2 &= (s, FAS^{6023}, \lambda) \\
M_3 &= (c, FADD^{10040} + CASP8^{20074} + FLIP^{48786} + CASP3^{120460} + Bid^{15057} + \\
&\quad Bax^{50189} + XIAP^{18069} + Apaf^{60230} + CASP9^{12046}, \lambda) \\
M_4 &= (m, Smac^{60230} + Cyto.c^{60230} + Bcl2^{45172}, \lambda)
\end{aligned} \tag{7.3}$$

The rule constants used in this first experiment are those computed according to the discussion in section 2.6 from the deterministic macroscopic constants presented in Figure 7.16 except for rules $r_{96'}, r_{97'}, r_{96''}$ and $r''_{97}$ whose constants will be set to zero since in this first experiment the mechanisms they represent are assumed not to take place within the system. This first set of rule constants is denoted by $C^a$.

Therefore, the first P system model to be analysed is $PSM_a = (\Pi_{FAS}; (M_1, M_2, M_3, M_4), C^a)$. The results from the ODE model in [55] show that caspase 3 is activated after 4 hours, which was considered close to the experimental results where caspase 3 was found to be activated after 6 hours (see Figure 7.17). By running our model $PSM_a$ using the Deterministic Waiting Times algorithm the caspase 3 activation dynamics are studied when Bcl2 is at baseline value. Caspase 3 is activated in our simulations after about 7 hours which is a very good approximation to the experimental data and improves on the results obtained in the ODE model [55].

There are cells (as thymocytes and fibroblasts) which are not sensitive to Bcl2 overexpression as described in [114]. In these cells caspase 8 directly activates caspase 3.

Scaffidi et al. has suggested in [114] that the type of pathway activated by Fas is chosen based on the concentration of caspase 8 generated in active form following FASL binding. If the concentration of active caspase 8 is high, then the caspase 3 is activated directly, on the other hand, if the concentration of activated caspase 8 is low, the type II pathway is chosen. In this last case, the system amplifies the death signal through the mitochondria to be able to induce cell death.

To check this hypothesis, the active caspase 8 formation was increased by making the initial concentration of caspase 8 a value 20 times greater than its baseline value while keeping everything else the same. The initial multiset associated with the cytoplasm in this case is:

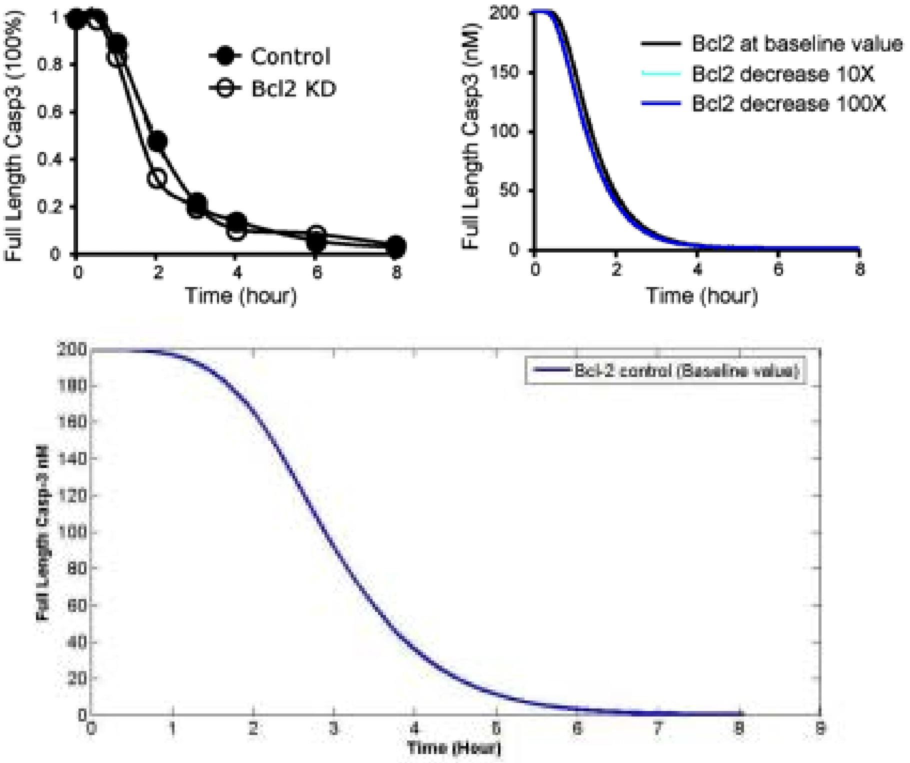

Figure 7.17.: Comparison between experimental data (top left, from [55]), previous ODE simulation data (top right, [55]) and the P system simulation data (down).

$$\begin{aligned} M_3' \;=\; & (c, FADD^{10040} + CASP8^{401480} + FLIP^{48786} + CASP3^{120460} + Bid^{15057} + \\ & Bax^{50189} + XIAP^{18069} + Apaf^{60230} + CASP9^{12046}, \lambda) \end{aligned} \tag{7.4}$$

The P system model used in this experiment is $PSM_a' = (\Pi_{FAS}; (M_1, M_2, M_3', M_4), C^a)$. The results obtained after running simulations of this P system model show a faster activation of caspase 3 which agrees well with the results obtained in [114].

We also study the sensitivity of the caspase 3 activation with regard to the number of Bcl2 molecules. This was done by incresing the number of molecules 100 times with respect to the initial multisets $M_3$ to obtain a new initial multiset associated with the cytoplasm that will be denoted by $M_3''$. The P system model used in this experiment is $PSM_a'' = (\Pi_{FAS}; (M_1, M_2, M_3'', M_4), C^a)$. Simulations of this model together with the previous model representing Bcl2 at baseline, $PSM_a$ are presented in Figure 7.18.

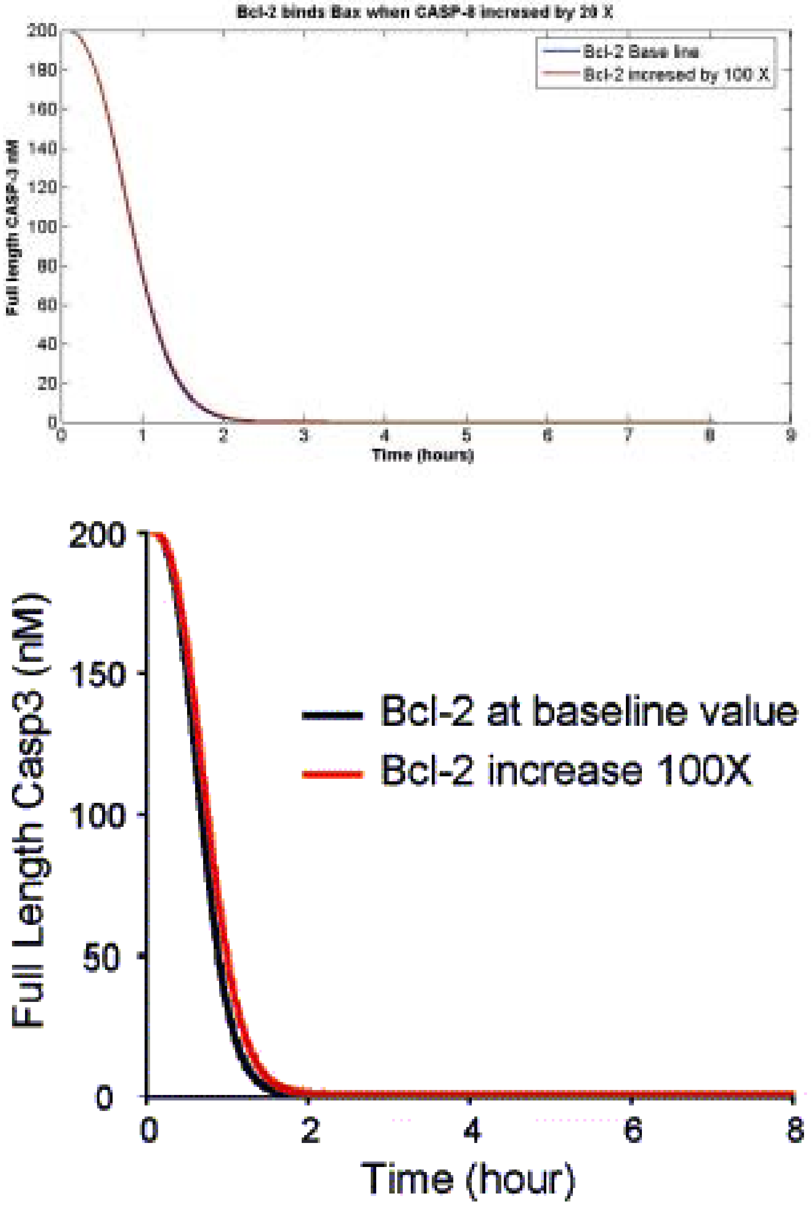

Figure 7.18.: Left the P system simulation, right the ODE simulation, from [55], for the change in caspase 8 initial concentration.

These simulations show that caspase 3 activation is not sensitive to increases in Bcl2 concentration, when the pathway of type I is chosen.

Bcl2 is known to block the mitochondrial pathway; however, the mechanism through which Bcl2 can block the pathway of type II is not clear. Next, we analyse caspase 3 activation kinetics in this type of pathway by considering different mechanisms for blocking the mitochondrial pathway proposed in [85] and [127].

These blocking mechanisms and the corresponding P system models used to repre-

| | Activation Caspase 3<br>(with overexpression of Bcl2) |
|---|---|
| Type I (death receptor pathway) | *Insensitivity* |
| Type II (mitochondrial pathway) | *Sensitivity* |

Figure 7.19.: Sensitivity of caspase 3 activation to overexpression of Bcl2

sent them are presented next.

(a) Bcl2 might bind to Bax. This situation is represented in the P system model, $PSM_a$ presented earlier where the rules $r_{96}$ and $r_{97}$ were associated with their corresponding constants as presented in Figure 7.16 but the rules $r'_{96}, r'_{97}, r''_{96}$ and $r''_{97}$ were *turned off* by associating a constant with value zero with them.

(b) Bcl2 might bind to Bid. This situation is described by setting to zero the constants associated with rules $r_{96}, r_{97}, r''_{96}$ and $r''_{97}$ while rules $r'_{96}$ and $r'_{97}$ are associated with their corresponding constants, as presented in Figure 7.16. This set of possible values for the rule constants are collected in $C_b$. Therefore the P system model describing this situation is $PSM_b = (\Pi_{FAS}; (M_1, M_2, M_3, M_4), C^b)$.

(c) Bcl2 might bind to tBid. This situation is described by setting to zero the constants associated with rules $r_{96}, r_{97}, r'_{96}$ and $r'_{97}$ while rules $r''_{96}$ and $r''_{97}$ are associated with their corresponding constants, as presented in Figure 7.16. This set of possible values for the rule constants are collected in $C_c$. Therefore the P system model describing this situation is $PSM_c = (\Pi_{FAS}; (M_1, M_2, M_3, M_4), C^c)$.

(d) Bcl2 might bind to both Bax and tBid. This situation is described by setting to zero the constants associated with rules $r'_{96}$ and $r'_{97}$ while rules $r_{96}, r_{97}, r''_{96}$ and $r''_{97}$ are associated with their corresponding constants, as presented in Figure 7.16. This set of possible values for the rule constants are collected in $C_d$. Therefore the P system model describing this situation is $PSM_d = (\Pi_{FAS}; (M_1, M_2, M_3, M_4), C^d)$.

Using the previous P system models the dynamics of caspase 3 activation were studied. It was found that the binding of Bcl2 to both Bax and tBid is the most efficient mechanism out of the four presented previously. Figure 7.20 compares the results obtained using the ODE model in [55] with the P system models presented in this section.

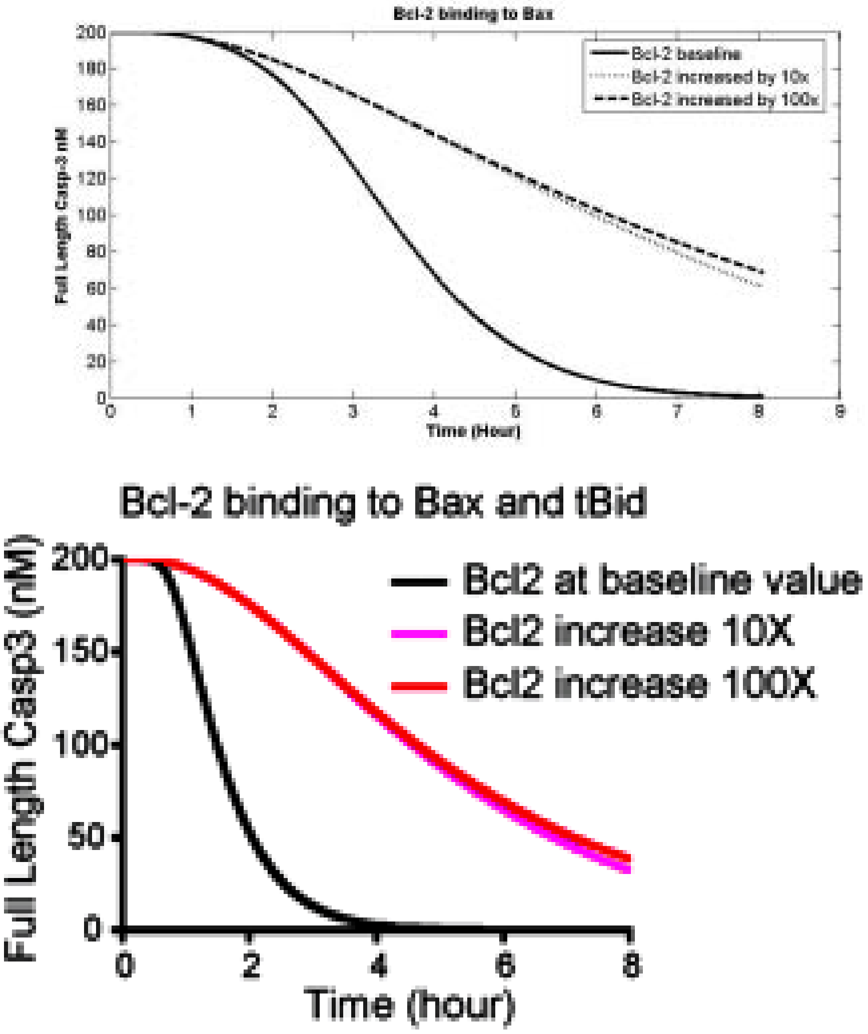

Figure 7.20.: Above the P system simulation, below the ODE simulation, from [55].

Finally, Figure 7.19 depicts a table presenting the sensitivity of caspase 3 activation to overexpression of Bcl2 in the functioning of the selected pathway.

# 8. Modelling Quorum Sensing

In this chapter we will study a communication mechanism in colonies of bacteria consisting of a gene regulation system which, depending on the number of individuals, allows the whole colony of bacteria to express certain and specific genes in a coordinated way. This cell density dependent gene regulation system is referred to as *quorum sensing*, and has been described as the most consequential molecular microbiology story of the last decade [51].

In the first section we present a brief description of the quorum sensing system in the marine bacterium Vibrio fischeri. In section 8.2 some principles for the development of specifications and models for colonies of bacteria within the P system modelling framework are presented in general and in particular for the quorum sensing system in Vibrio fischeri. In colonies of bacteria the size of the environment is such that it cannot be considered a well mixed volume (see the discussion in section 2.6) so the environment will be partitioned into a grid represented by a *multienviroment.* A variant of P systems called *population P systems* is introduced as a suitable framework for the development of specifications and models of colonies of bacteria. Finally, in section 8.3 various P system models describing colonies of different sizes will be analysed using the Multicompartmental Gillespie algorithm introduced in section 3.3.

## 8.1. Quorum Sensing System in Vibrio Fischeri

Bacteria are generally considered to be independent unicellular organisms. However it has been observed that certain bacteria, like the marine bacterium *Vibrio fischeri*, exhibit coordinated behaviour which allows an entire population of bacteria to regulate the expression of certain or specific genes in a coordinated way depending on the size of the population. This cell density dependent gene regulation system is referred to as *quorum sensing* [128].

This phenomenon was first investigated in the marine bacterium *Vibrio fischeri.* This bacterium exists naturally either in a free-living planktonic state or as a symbiont of certain luminescent squid. The bacteria colonise specialised light organs in the squid, which cause it to luminesce. Luminescence in the squid is thought to be involved in the attraction of prey, camouflage and communication between different individuals. The source of the luminescence is the bacteria themselves. The bacteria only luminesce when colonising the light organs and do not emit light when in the free-living state [110].

The quorum sensing system in Vibrio fischeri relies on the synthesis, accumulation and subsequent sensing of a signal molecule, 3-oxo-C6-HSL, an N-acyl homoserine lactone or AHL, we will refer to as OHHL. When only a small number of bacteria are present these molecules are produced by the bacteria at a low level. OHHL diffuses out of the bacterial cells and into the surrounding environment. At high cell density the signal accumulates in the area surrounding the bacteria and can also diffuse to the inside of the bacterial cells. The signal is able to interact with the LuxR protein to form the complex LuxR-OHHL. This complex acts as a transcription factor binding to a region of DNA called the Lux Box causing the transcription of the luminescence genes, a small cluster of 5 genes, luxCDABE. Adjacent to this cluster are two regulatory genes for the transcription of LuxR and OHHL. In this sense OHHL and LuxR are said to be autoinducers because they activate their own synthesis. A graphical representation of the system can be seen in Figure 8.1.

The bacteria are effectively communicating, as a single bacterium is able to detect and respond to signals produced by the surrounding bacteria. Bacteria sense their cell density by measuring the amount of signals. Quorum sensing therefore explains why the bacteria are dark when in the free living planktonic state at low cell density and light when colonising the light organ of the squid at high cell density. A large number of Gram negative bacteria have been found to have AHL-based quorum sensing systems similar to Vibrio fischeri. For a review on quorum sensing in general and the system in Vibrio fischeri see [128].

## 8.2. A P System Specification of the Quorum Sensing in Vibrio fischeri

In this section we present a P system specification of the quorum sensing system in Vibrio fischeri. Although this system consists of genes arranged in operons we will not represent them using strings as described in section 4.3.2. Instead we will follow section 4.3.1 and represent them as individual objects. We have chosen this approach because in this case we are focused on the analysis of the behaviour of the whole colony of bacteria rather than on the analysis of the behaviour of a single bacterium. In this

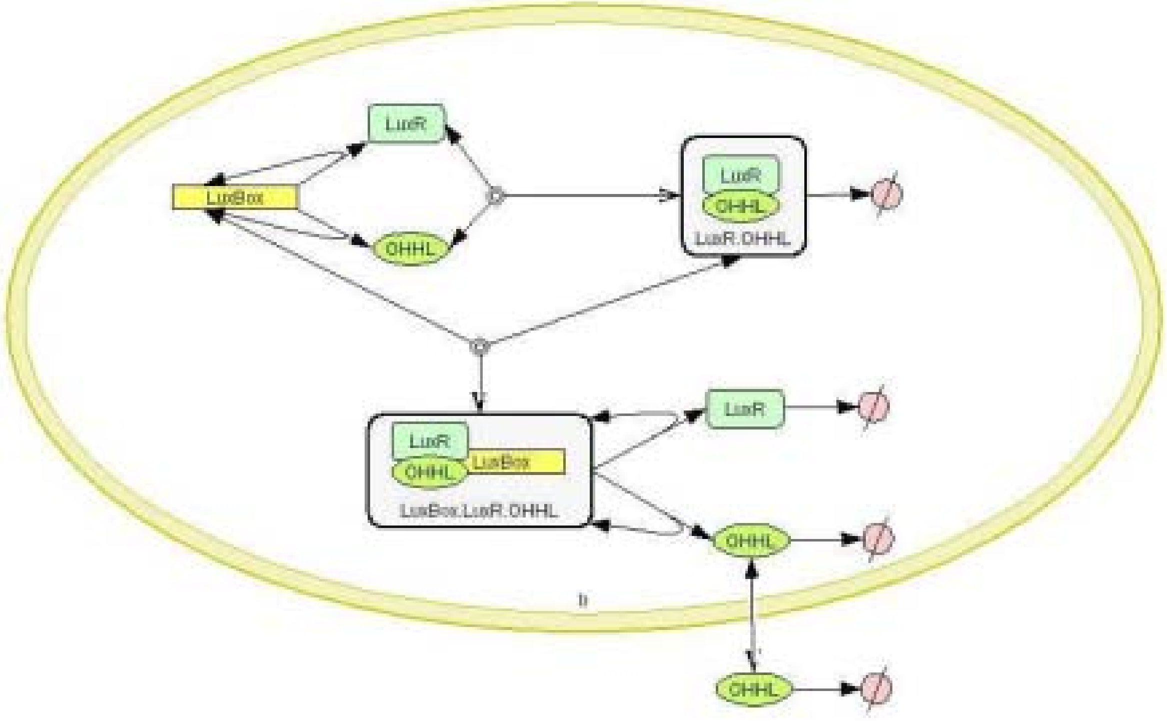

Figure 8.1.: Quorum sensing system in Vibrio fischeri

respect, the representation of genes and the processes involved in gene regulation as strings and rewriting rules on multisets of strings would have made the process of analysing the system intractable.

In the quorum sensing system in Vibrio fischeri there are two relevant regions, namely the environment and the bacteria. Each bacterium in the colony will be specified using a membrane. The representation of the environment is not straight forward in this case. As discussed in section 4.1 , a membrane in a P system specification is assumed to define a well mixed region. Formula (2.12) in section 2.6 states the condition a region must fulfill in order to be considered well mixed. Roughly speaking, this condition states that the region must be small enough compared to the average time a diffusing molecule needs to visit the whole region. The size of the environment in a colony of bacteria does not fulfill this condition and therefore it cannot be represented by a single membrane.

In order to solve this problem we introduce a variant of P systems where the environment is partitioned into a grid big enough so that each piece can be considered a well mixed region. Each of these regions is then represented using a membrane. This structure will be referred to as a *multienvironment*.

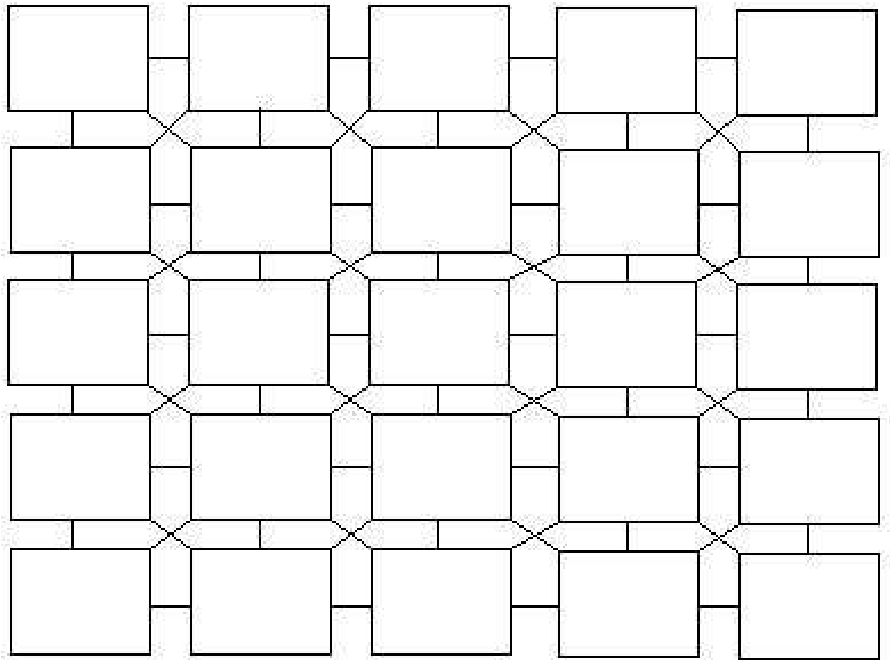

Figure 8.2.: A multienvironment with 25 different environments.

**Definition 8.1** (multienvironment)**.** A *multienvironment* is a collection of membranes called *environments* and communication links between them. Formally, a multienvironment is a graph, $G = (V, S)$, whose nodes $V$ are membranes representing environments. These environments are connected according to the edges $S$ that define the links between them.

Each environment will contain a multiset of objects representing molecular entities and a number of P system specifications representing the individual bacteria or cells located in the region of the system represented by this environment. Furthermore, each environment will be associated with a set of rules of one of the following forms:

- The molecular entities in the environments can interact yielding new molecules. These interactions are described by the following type of rules:

$$[\, obj_1 \,]_l \xrightarrow{c} [\, obj_2 \,]_l \tag{8.1}$$

These rules only change the content of one environment. In order to apply

Gillespie's theory of stochastic kinetics a propensity must be associated with these types of rules. In this case this propensity is computed by multiplying the stochastic constant $c$ associated specifically with the rule by the number of distinct possible combinations of the objects in the multiset $obj_1$.

- In the case of movement of different substances from one environment to one of the environments connected to it, the following type of rules will be used:

$$[\, obj \,]_l - [\ ]_{l'} \xrightarrow{c} [\ ]_l - [\, obj \,]_{l'} \tag{8.2}$$

  These rules are multiset rewriting rules that operate on two environments, one labelled $l$ which is linked to another environment labelled $l'$. A multiset $obj$ is removed from the first environment and placed in the second one. In this way, we are able to capture in a concise way the diffusion of signals from one region to another in a large environment. As in the previous case, a propensity is associated with these type of rules.

- Besides multisets of objects representing molecules, a certain number of copies of P system specifications are placed in the environments. These P system specifications represent individuals from a colony that can move from one environment to another by applying rules of the form:

$$[\, [\,]_{l''} \,]_l - [\ ]_{l'} \xrightarrow{c} [\ ]_l - [\, [\,]_{l''} \,]_{l'} \tag{8.3}$$

  When a rule of this type is applied, a membrane with label $l''$ and all its contents, objects and other membranes, is moved from an environment labelled $l$ to another connected to it that must be labelled $l'$.

  The propensity of this rule is computed by multiplying the stochastic constant $c$ by the number of membranes labelled by $l''$ located inside the environment with label $l$.

Finally in order to study a colony or population of individuals, each type of individual is represented by a P system specification. Then different copies of the P system specifications describing different individuals are randomly distributed in the environments of a *multienvironment.* This construct will be referred to as a *Population P system Specification.*

**Definition 8.2** (Population P system Specification)**.** A *population P system specification* is a construct:

$$P\Pi = (\Sigma, H, G, E_1, \ldots, E_m, R_{l_1}, \ldots, R_{l_t}, \Pi_1, \ldots, \Pi_p, k_1, \ldots, k_p) \tag{8.4}$$

where:

- $\Sigma$ is a finite alphabet of objects.
- $G = (V, S)$ is a graph whose nodes $V = 1, \ldots, m$ are membranes representing environments labelled with elements from $H = \{l_1, \ldots, l_t\}$ and whose edges, $S$, define how the environments are linked.
- $E_j = (l_j, w_j)$ for each $1 \leq j \leq m$, is the initial configuration of the environment $j$ with $l_j \in H$ and $w_j \in \Sigma^*$ a finite multiset of objects.
- $R_{l_1}, \ldots, R_{l_t}$ are finite sets of rules of the forms (8.1), (8.2) and (8.3) associated with environments of the type specified by the labels $l_1, \ldots, l_t$.
- $\Pi_1, \ldots, \Pi_p$ are P system specifications as in definition 3.1 describing the different individual types in the population.
- $k_1, \ldots, k_p \in \mathbb{N}$ is the number of copies of the P system specifications $\Pi_1, \ldots, \Pi_p$ that are distributed randomly in the environments in the initial configuration of the system.

**Definition 8.3** (Parameters of a Population P system Specification)**.** Given a Population P system specification

$$P\Pi = (\Sigma, H, G, E_1, \ldots, E_m, R_{l_1}, \ldots, R_{l_t}, \Pi_1, \ldots, \Pi_p, k_1, \ldots, k_p)$$

the set of parameters of $P\Pi$,

$$\mathcal{P}(P\Pi) = (E_1, \ldots, E_m, \mathbb{C}_{l_1}, \ldots, \mathbb{C}_{l_t}, \mathcal{P}(\Pi_1), \ldots, \mathcal{P}(\Pi_p), k_1, \ldots, k_p)$$

consists of:

- The initial multisets associated with the environments, $E_1, \ldots, E_m$.
- The stochastic constants, $\mathbb{C}_{l_1}, \ldots, \mathbb{C}_{l_t}$, associated with the rules assigned to the environments $R_{l_1}, \ldots, R_{l_t}$.
- The parameters associate with the P system specification of the individual types $\mathcal{P}(\Pi_1), \ldots, \mathcal{P}(\Pi_p)$.
- The number of copies, $k_1, \ldots, k_p$, of the P system specifications $\mathcal{P}(\Pi_1), \ldots, \mathcal{P}(\Pi_p)$.

This variant of P systems was introduced under the name of quorum sensing P systems in [17] and their computational power studied.

These definitions will be used in the specification of the Quorum sensing system in Vibrio fischeri. The specific population P system specification used in this work is presented in what follows.

$$P\Pi_{VF} = (\Sigma, \{e\}, G, E_1, \ldots, E_{25}, R_e, \mathbf{\Pi}_{VF}, N) \quad \text{where:} \tag{8.5}$$

1. The alphabet $\Sigma$ collects the objects which represent the molecular entities present in the environments. In this case, $\Sigma$ contains a single object, OHHL, representing the signal 3-oxo-C6-HSL. This is the only molecule that can be present in the environments.

2. In the specification of the quorum sensing system in Vibrio fischeri all environments have the same label $e$. That is the set of labels associated with the environments consists of a single label $H = \{e\}$.

3. The media where the colony of Vibrio fischeri bacteria is placed is represented using a multienvironment, $G$, which consists of 25 environments. The way in which the environments are connected in $G$ is depicted in Figure 8.2.

4. The initial multisets, $E_1, \ldots, E_{25}$ associated with the environments are part of the parameters of this population P system specification. They will describe the initial number of signal molecules in each environment.

5. The P system specification $\mathbf{\Pi}_{VF}$ represents a Vibrio fischeri bacterium.

$$\mathbf{\Pi}_{VF} = (O, \{b\}, [\ ], M_1, R_b) \quad \text{where:} \tag{8.6}$$

   a) In our specification of the quorum sensing system in Vibrio fischeri we will take into account a cluster of genes, a signal molecule and a protein acting as a transcription factor. Although the description of genes as single objects is questionable in certain cases, we will adopt this approach in our study as we are focused on the analysis of the whole colony of bacteria rather than on a single bacterium. The alphabet $O$ collects all the objects used to specified the signals, proteins, complexes and genes present inside a Vibrio fischeri bacterium. Figure 8.3 presents the objects in $O$ and the molecular entities they represent.

$$O = \{\text{OHHL, LuxR, LuxR-OHHL, LuxBox, LuxR-OHHL-LuxBox}\}$$

| **Object** | **Molecular Entity** |
|---|---|
| OHHL | Signal molecule 3-oxo-C6-HSL |
| LuxR | Transcription factor, intracellular recpetor |
| LuxR-OHHL | Active transcription factor |
| LuxBox | Binding site for LuxR-OHHL |
| LuxR-OHHL-LuxBox | Site occupied by LuxR-OHHL |

Figure 8.3.: Objects in the P system specification of a Vibrio fischeri bacterium

b) A single membrane is used to describe a bacterium. This membrane defines the bacterium itself separating it from the environment. This membrane will be labelled $b$.

c) The initial multiset $M_1$ represents the initial configuration of the membrane describing a bacterium. As mentioned above the label $b$ will be associated with this membrane. With regard to the multiset of objects, we are interested in examining how bacteria communicate to coordinate their behaviours and how the population moves from a downregulated state, where the protein and the signal are produced at basal rates, to an upregulated state where there is a massive production of signals and proteins. Therefore, in the initial multiset $M_1$ we will only have the genes involved in the quorum sensing system, represented by LuxBox, to start the production of the signal, OHHL, and protein, LuxR.

d) In the set of rules, $R_b$, we specify the molecular interactions forming the quorum sensing system inside the bacteria. In what follows we enumerate these rules presenting a brief description of the interactions they describe.

($*$) In an unstressed bacterium the production of the signal OHHL and the protein LuxR takes place at basal rates from the genes transcribed from the site LuxBox.

$r_1 : [\text{ LuxBox }]_b \xrightarrow{c_1} [\text{ LuxBox + OHHL }]_b$

$r_2 : [\text{ LuxBox }]_b \xrightarrow{c_2} [\text{ LuxBox + LuxR }]_b$

($*$) The protein LuxR acts as an intracellular receptor and OHHL as its ligand. Both together form the complex LuxR-OHHL which in turn can dissociate into OHHL and LuxR again.

$r_3 : [\text{ LuxR + OHHL }]_b \xrightarrow{c_3} [\text{ LuxR-OHHL }]_b$

$r_4 : [\text{ LuxR-OHHL }]_b \xrightarrow{c_4} [\text{ LuxR + OHHL }]_b$

($*$) The complex LuxR-OHHL acts as a transcription factor binding to the regulatory region of the bacterium DNA called LuxBox. The complex LuxR-OHHL can also dissociate from the LuxBox.

$r_5 : [\text{ LuxR-OHHL} + \text{LuxBox }]_b \overset{c_5}{\rightarrow} [\text{ LuxR-OHHL-LuxBox }]_b$

$r_6 : [\text{ LuxR-OHHL-LuxBox }]_b \overset{c_6}{\rightarrow} [\text{ LuxR-OHHL} + \text{LuxBox }]_b$

($*$) The binding of the complex LuxR-OHHL to the LuxBox produces a massive increase in the production of the signal OHHL and of the protein LuxR.

$r_7 : [\text{ LuxR-OHHL-LuxBox }]_b \overset{c_7}{\rightarrow} [\text{ LuxR-OHHL-LuxBox} + \text{OHHL }]_b$

$r_8 : [\text{ LuxR-OHHL-LuxBox }]_b \overset{c_8}{\rightarrow} [\text{ LuxR-OHHL-LuxBox} + \text{LuxR }]_b$

($*$) The signal OHHL can diffuse outside the bacterium and accumulate in the environment.

$r_9 : [\text{ OHHL }]_b \overset{c_9}{\rightarrow} \text{OHHL } [\ ]_b$

($*$) OHHL, LuxR and the complex LuxR-OHHL undergo a process of degradation in the bacterium

$r_{10} : [\text{ OHHL }]_b \overset{c_{10}}{\rightarrow} [\ ]_b$

$r_{11} : [\text{ LuxR }]_b \overset{c_{11}}{\rightarrow} [\ ]_b$

$r_{12} : [\text{ LuxR-OHHL }]_b \overset{c_{12}}{\rightarrow} [\ ]_b$

6. All the environments will have the same set of rules associated with them, $R_e$. This set of rules describes the processes taking place in the environments. Below we present an enumeration of the rules in $R_e$ and a brief description of the processes they specify.

($*$) Once the signal OHHL accumulates in the environment it can also diffuse inside the bacteria.

$r_{13} : \text{OHHL } [\ ]_b \overset{c_{13}}{\rightarrow} [\text{ OHHL }]_b$

($*$) The signal, OHHL, can also be degraded in the environments.

$r_{14} : [\text{ OHHL }]_e \overset{c_{14}}{\rightarrow} [\ ]_e$

($*$) The signal diffuses from one environment to another.

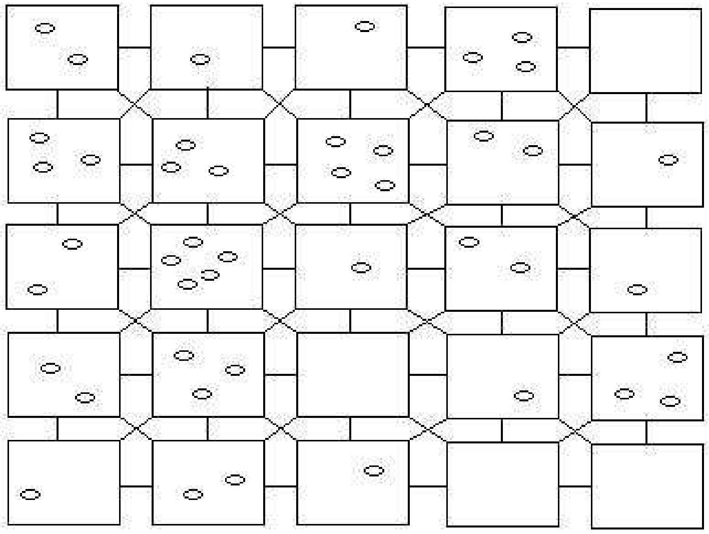

Figure 8.4.: A possible initial configuration of our Population P system specification

$r_{15}$ : $[\ \mathrm{OHHL}\ ]_e - [\ ]_e \overset{c_{15}}{\rightarrow} [\ ]_e - [\ \mathrm{OHHL}\ ]_e$

(∗) Bacteria can also move freely from one environment to another.

$r_{16}$ : $[\ [\ ]_b\ ]_e - [\ ]_e \overset{c_{16}}{\rightarrow} [\ ]_e - [\ [\ ]_b\ ]_e$

7. Finally in order to study the behaviour of a colony of bacteria represented by the P system specification $\mathbf{\Pi}_{VF}$ we have to distribute randomly $N$ copies of $\mathbf{\Pi}_{VF}$ in the multienvironment $G$. An example of a colony of bacteria consisting of 44 individuals is presented in Figure 8.4.

## 8.3. P Systems Models of the Quorum Sensing System in Vibrio fischeri

Models of a population P system specification are obtained in the same way as in the case of a simple P system specification. More precisely, in order to develop a population P system model one has to assign specific values to the parameters of the corresponding population P system specification.

As discussed previously the parameters associated with a population P system specification consist of:

- The initial multisets associated with the environments. In the models of the quorum sensing system in Vibrio fischeri developed in this section all the environments will be empty and they will be labelled with the same label $e$. That is, $E_1 = \cdots = E_{25} = \mathbb{E}_0 = (\mathrm{e}, \lambda)$.

- The stochastic rule constants, $\mathbb{C}_e = \{c_{13}, c_{14}, c_{15}, c_{16}\}$, of the set of rules $R_e$, associated with the environments. In our model these constants are assigned a fixed value, $c_{13} = 1, c_{14} = 5, c_{15} = 8$ and $c_{16} = 2$.

- The parameters corresponding to the P system specification of a Vibrio fischeri bacterium, $\mathbf{\Pi}_{VF}$:

  Recall that according to definition 3.2 the parameters of a simple P system specification are the initial multisets and the rule constants.

  Regarding the initial multiset associated with the membrane describing a bacterium, recall that we are interested in examining how bacteria communicate to coordinate their behaviours and how the population moves from a downregulated state, where the protein and the signal are produced at basal rates, to an upregulated state where there is a massive production of signals and proteins. Therefore, in the initial multiset $M_1$ we will only have the genome LuxBox to start the production of the signal OHHL and protein LuxR at basal rates, $\mathbb{M}_0 =$ LuxBox.

  Regarding the stochastic constants associated with the rules in $R_b$, $\mathbb{C}_b = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}\}$. Where possible we have used estimates derived from the literature [34, 47, 48, 118, 132]. For parameter values not available from these sources we used a "trial-and-error" approach, making an initial "guess" at the values of the missing constants and then comparing the resulting behaviour with known properties of the system. If they did not match the unknown constants were then adjusted systematically, one parameter at a time.

  By following this process the following set of parameters were chosen: $c_1 = 2, c_2 = 2, c_3 = 9, c_4 = 1, c_5 = 10, c_6 = 2, c_7 = 250, c_8 = 200, c_9 = 50, c_{10} = 30, c_{11} = 20, c_{12} = 20, c_{13} = 1, c_{14} = 5, c_{15} = 8, c_{16} = 2$. These values have been set such that the degradation rates compensate the basal production of the signal and the protein and such that the production rates when the regulatory region is occupied produce a massive increase in the transcription of the signal and the protein.

- Finally, the last parameter is $N$, the number of copies of the P system specification $\mathbf{\Pi}_{VF}$ distributed randomly in the environments. This parameter constitutes the only variable parameter of the system in contrast to the previous ones that are assigned fixed values. This parameter represents the size of the colony of Vibrio fischeri bacteria and will allow us to study the behaviour of colonies of different sizes.

Summing up the population P system models that will be used in this section consists in the following construct:

$$PPSM_{VF}(N) = (P\Pi_{VF}; (\mathbb{E}_0, \dots, \mathbb{E}_0), \mathbb{C}_e, (\mathbf{\Pi}_{VF}; \mathbb{M}_0, \mathbb{C}_b), N) \tag{8.7}$$

This model has been represented in the Systems Biology Markup Language using CellDesigner, a structured diagram editor for drawing gene-regulatory and biochemical networks [37].

The analysis of the behaviour of a colony of bacteria of different sizes will be through running simulations. We have run our simulations using a program written in C with an SBML input file specifying our model. This software tool is available from [134].

The emergent behaviour of the system has been studied for two colonies of different size to examine how bacteria can sense the number of bacteria in the colony and produce light only when the number of individuals is big enough.

- First we have considered a population of 100 bacteria represented in the Population P system model $PPSM_{VF}(100)$.

  In Figure 8.5 we show the evolution over time of the number of quorated bacteria [1] and the number of OHHL signals in the environments.

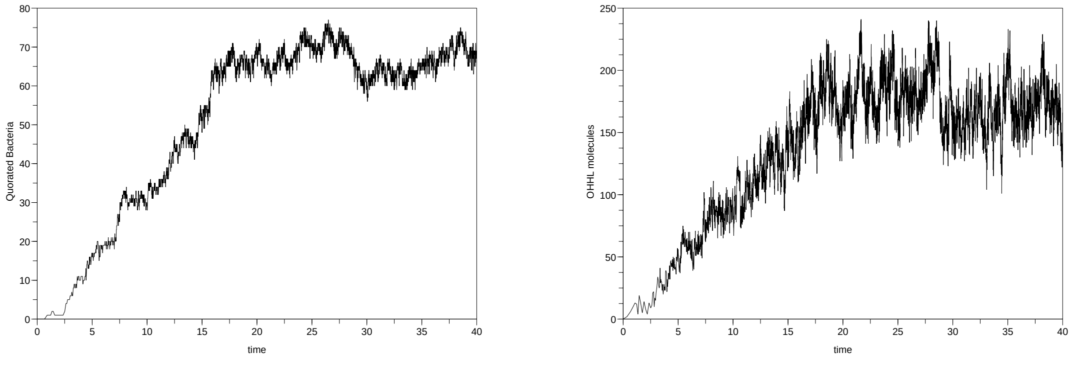

Figure 8.5.: Number of quorated bacteria (left) and signals in the environment (right)

[1] We will say that a bacterium is quorated if the LuxBox in this bacterium is occupied by the complex producing the transcription of the enzymes involved in the production of light.

Observe that the signal, OHHL, accumulates in the environments until saturation and then, when this threshold is reached, bacteria are able to detect that the size of the population is big enough. At the beginning, a few bacteria get quorated and then accelerate a process of recruitment that makes the whole population of bacteria behave in a coordinated way.

Note that there exists a correlation between the number of signals in the environments and the number of quorated bacteria such that, when the number of signals in the environment drops, so does the number of quorated bacteria and when the signal goes up it produces a recruitment of more bacteria.

In our approach the behaviour of each individual in the population can be tracked. In Figure 8.6 we have taken a sample of three bacteria and have studied the correlation between the number of signals inside each bacterium (first row) and the occupation of the LuxBox by the complex (second row) which represents that the bacterium has been quorated.

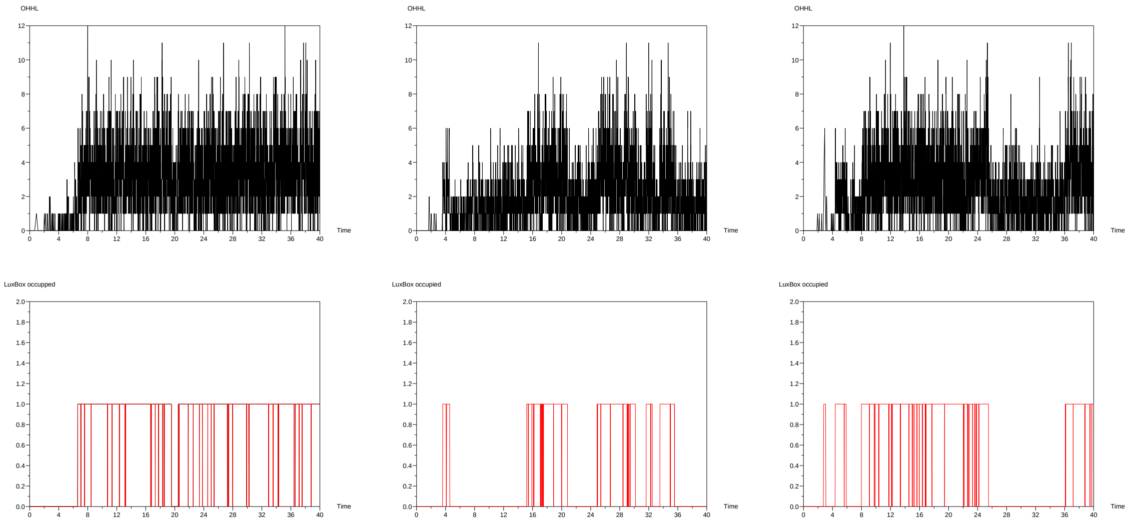

Figure 8.6.: A sample of three bacteria: Signal and Occupation of the LuxBox

Observe in Figure 8.6 that the number of signal molecules inside each bacterium has to exceed a threshold of approximately seven molecules in order to recruit the bacterium. On the one hand, when the number of molecules is greater than seven the LuxBox is occupied, that is, the bacterium is quorated or upregulated. On the other hand, when there are fewer than seven signals the bacterium switches off the system and it goes downregulated.

We can also study how rules are applied across the evolution of the system. Figure

8.7 shows the evolution of the number of applications of the rules representing the basal production (first two graphs) and the rules representing the production of the signal and protein induced by the binding of the complex to the LuxBox.

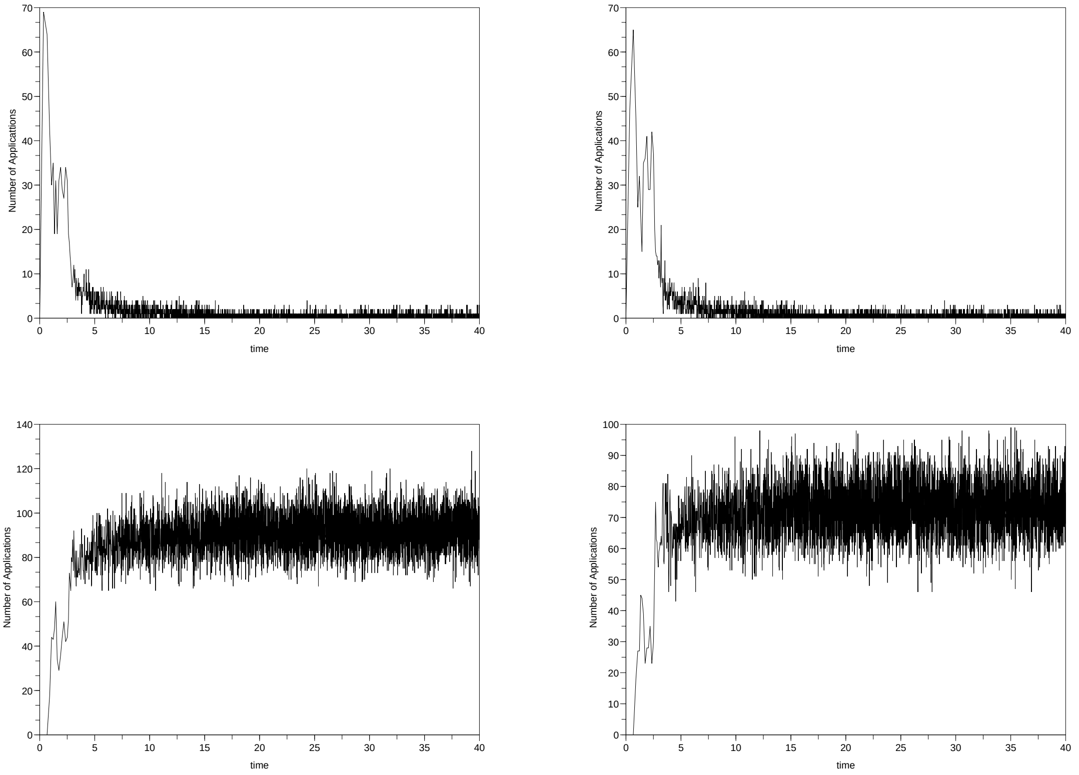

Figure 8.7.: Number of applications of rules $r_1$, $r_2$ (first row) and $r_7$,$r_8$ (second row)

Figure 8.7 shows how at the beginning the basal production rules are the most applied rules while the other two are seldomly applied. But then, as a result of the recruitment process the bacteria sense the size of the population and they behave in a coordinated way, with a massive application of the third and fourth rules. So the system moves from a downregulated state to an upregulated state where the bacteria are luminescent.

- In order to study how bacteria can sense the number of individuals in the colony and get quorated only when the size of the colony is big enough we have examined the behaviour of a colony of only 10 bacteria. This colony of bacteria is represented in the Population P system model $PPSM_{VF}(10)$.

  In this case no recruitment process takes place and the signal does not accumulate in the environment. Only one of the bacteria wrongly guessed the size of the population and entered an upregulated state, see Figure 8.8. But then, after

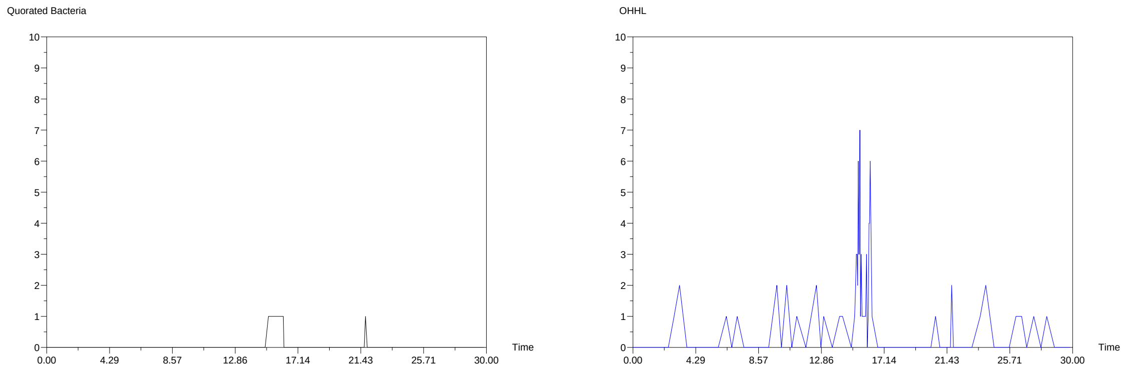

Figure 8.8.: Quorated bacteria and signals in the environment in a population of 10 bacteria.

sensing that the signal did not accumulate in the environment, it switched off its system.

In Figure 8.9 the behaviour of the bacterium that got quorated is depicted. Observe that this bacterium got quorated because the number of signals inside it exceeded the threshold of seven signals. Then it started to produce signals massively, these signals diffused to the environment where there was not enough bacteria to sense them and they were degraded. When the bacterium sensed that the signal did not accumulate in the environment it switched off its system.

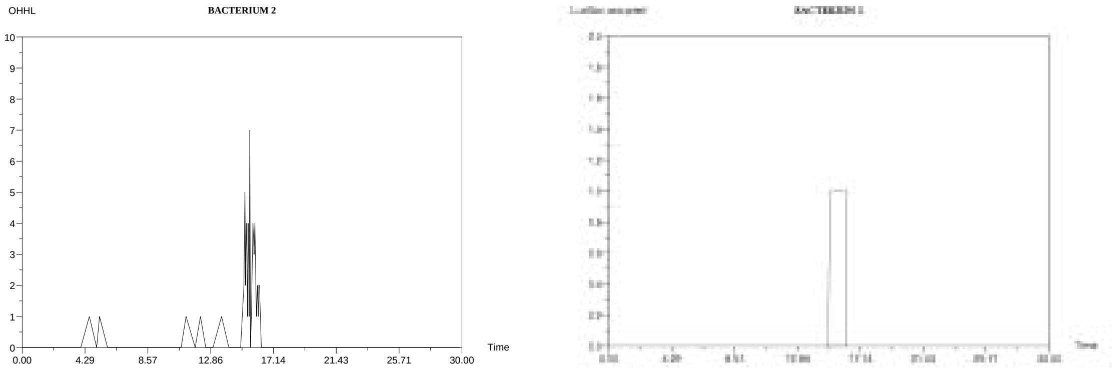

Figure 8.9.: Behaviour of a bacterium in a population of 10 bacteria.

Finally, observe in Figure 8.10 that for only ten bacteria the system remains in an downregulated state only applying the rules representing the basal productions while the rules associated with the production of light are seldomly applied.

Summing up, our simulations show that Vibrio fischeri has a quorum sensing sys-

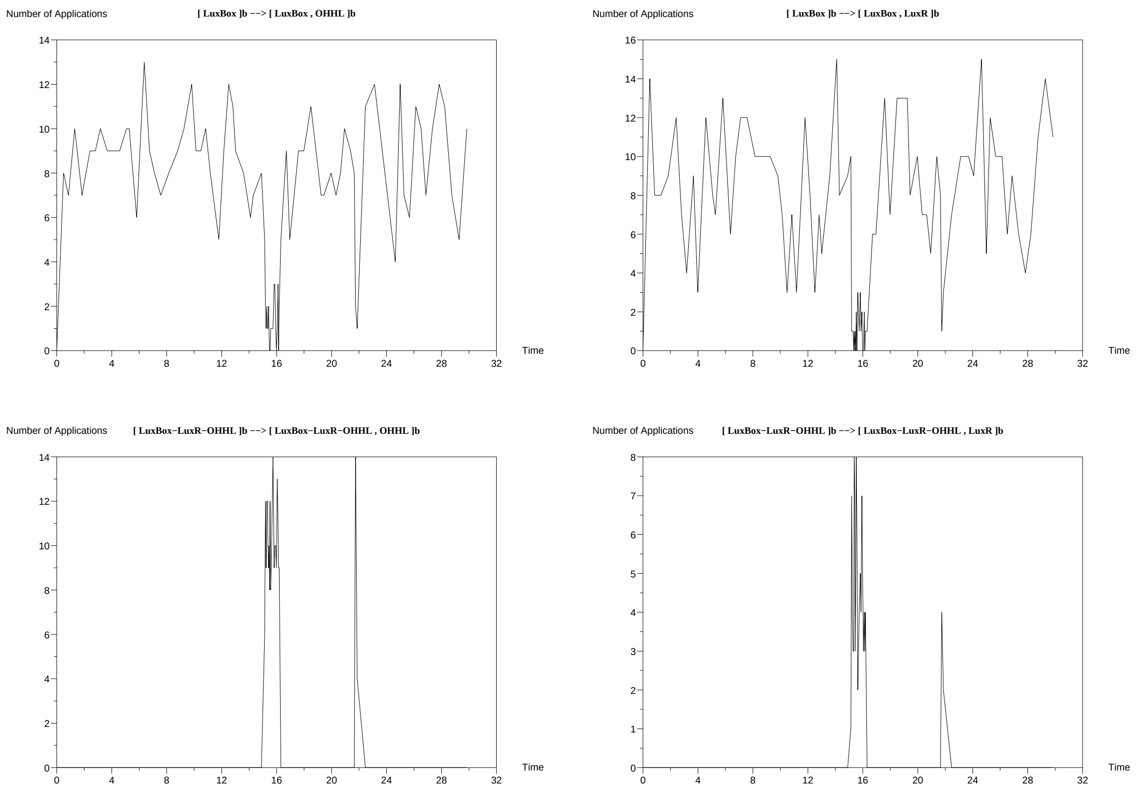

Figure 8.10.: Number of applications of rules $r_1$, $r_2$, $r_7$ and $r_8$ in 10 bacteria

tem where a single bacterium can guess that the size of the population is big enough and start to produce light. This bacterium starts to massively produce signals, but if the signal does not accumulate in the environment it means that the guess was wrong and it switches off the system. In contrast, if the signal does accumulate in the environment meaning that the number of bacteria in the colony is big enough, a recruitment process takes place that causes the entire population of bacteria to become luminescent. Observe that this emergent behaviour is a result of local interactions in the environments between different simple agents, the bacteria, which only able to produce and receive molecular signals. In this respect, our approach, using population P systems, confirms the ability to reveal emergent behaviour, which has already been proved within the more abstract framework of general self-assembly [39] or for the specific case of graph self-assembly [16, 18]. The key difference between our approach and the previous models is given by the strategy used in selecting the rules that are applied.

# Part IV.

# Conclusions

*8.*

# 9. Conclusions

According to the original motivation [89] P systems were not intended to provide a comprehensive and accurate model of the living cell. Rather, they were introduced to explore the computational nature of various features of biological membranes. Indeed, most research in P systems concentrates on the study of simplified variants of the original membrane systems trying to show whether or not they are computationally complete and/or computationally efficient. Nevertheless, at the beginning of the development of the research described in this thesis P systems started to be used to model biological phenomena.

The first attempts presented a semi-quantitative framework taking into account the discrete character of the molecular interactions taking place inside a single compartment [6, 20]. Although these preliminary attempts were proved to achieve some success, they failed to model quantitative aspects that are key to the functioning of many cell systems. Specifically they kept the classical strategy for the evolution of P system models based on non determinism and maximal parallelism. This approach produces two inaccuracies:

- Rewriting rules are not applied at the correct rate.
- All time steps are equal and do not represent the time evolution of the real cellular system.

These two problems are interdependent and have been solved in this thesis by applying Gillespie's theory of stochastic kinetics to P systems.

Our modelling framework was not restricted to the simple generation of simulations of our models as it was the case in the previous works. In this thesis we have taken the first steps towards the development of techniques to analyse P system models based on probabilistic and symbolic model checking.

The starting point of the research discussed here was the study of various case studies involving signalling and transcription networks. Our experience gained during these investigations showed that the classical rewriting rules used in P systems lack a systematic analysis of what molecular interactions could be specified and model using them. In this thesis we have developed a description of how to specify and model the most important molecular interactions taking place in living cells. Furthermore, our investigations uncovered the fact that individual objects were not adequate to specify certain molecular entities, like operons and clusters of genes, with a relevant internal structure. In order to solve this problem the use of strings to specify such molecules were introduced together with specific rewriting rules on multisets of objects and strings.

At this point it worth recalling the features generally required in a *good* modelling framework for cellular systems, namely relevance, understandability, extensibility and computability. In what follows we present some benefits and limitations of the use of P systems as a modelling framework for systems biology with regard to these four desirable properties.

- **Relevance:** A relevant abstraction of cellular systems should capture two essential properties of these systems: their structural organisation and their dynamic behaviour. In this respect we believe that P systems are indeed highly relevant.

  On the one hand, the compartmentalisation of cellular systems is explicitly specified using membrane structures. Furthermore, strings are used to describe molecular entities with an important internal structure. Nevertheless, when the internal structure of the molecular entities are not crucial individual objects are used.

  On the other hand, the dynamic behaviour of cellular systems is very closely mimic using rewriting rules which are very similar to the classical equations written by biologists to describe biochemical interactions. Two alternative approaches are presented in this work, rewriting rules on multisets of objects and rewriting rules on multisets of objects and strings. These two abstractions for molecular interactions present a tradeoff between relevance and utility. Rewriting rules on multisets of objects are simpler but suffer from a relevance problem as for many molecular entities their internal structure is crucial being questionable to represent them as individual objects. Rewriting rules on multisets of strings and objects take into account the internal structure of molecules but we appreciate that they can be computationally very expensive depending on the size of the cellular system under study. For instance, this approach to study colonies of millions of bacteria is intractable.

  A key benefit of our proposed modelling framework is its ability to handle compartmentalisation and the key role played by membranes in the functioning of living cells. This is demonstrated by our collection of case studies presented in

Part III. Our approach has been shown very suitable to model selective uptake of molecules from the environment, signalling at the cell surface and colonies of interacting bacteria which communicate by sending and receiving diffusing signals. Nevertheless, we have not investigated other important processes where membranes are crucial like cell division, cell adhesion, biofilm formation, etc. The specification and simulation of this type of processes remain an open problem and a future direction to explore.

- **Understandability:** Perhaps the most important property in a modelling framework is understandability. The abstractions used in a good modelling framework for cellular systems should correspond well to the informal concepts and ideas from molecular cell biology. A model should provide a better and integrated understanding of the real cellular system instead of producing a complicated and hard to decipher formalism.

  In this respect, although P systems were not introduced to provide an understandable modelling framework for cellular systems but rather as an abstraction of the computational aspects of membranes the concepts and terminology they use are very close to those used in molecular cell biology. This makes the abstractions developed within this framework comprehensible to cell biologist. Since P systems were inspired by the structure and functioning of living cells there exists a direct correspondence between P system abstractions and the real components of cellular systems. For instance, compartments in cellular systems correspond to the regions define by the membrane structure of a P systems; the multisets associated with each membrane represent the molecular entities present in the compartment represented by the corresponding membrane and finally the rewriting rules describe the molecular interactions taking place in cellular systems.

  Although very easy to understand P systems present a current limitation to the transparency and utility of the specifications and models designed within their framework. The P system abstractions are purely textual and so far lack of a graphical formal representation for the visualization of the modelled systems. In this text we have very occasionally used a graphical representation based on [71] but a formal correspondence between P system models and this graphical representation remains an open problem.

- **Computability:** A good modelling framework for cellular systems should be computable in order to allow both the simulation of the dynamic behaviour of the systems and the reasoning on the systems properties. In this respect P systems constitute a suitable modelling framework due to their easy computability and implementation. These properties are mostly a consequence of the fact that P systems consists of rule based systems distributed in specific structures. In particular P system specifications and models are easily implemented using object oriented languages. Each membrane can be thought as an object whose attributes are the multiset associated with it and the label representing its type. The set of rules associated with each membrane can be considered as the functions or methods associated with the object representing the membrane. Furthermore, the algorithms used in this work to simulate P system models are local in the sense that almost all computations only consider a single compartment. The only remaining global computation is the location of the index of the rule to be applied and the compartment where it is applied. The advantage of having local computations is that our approach is easily implemented in an event-driven object-oriented programming style, such an implementation could be multithreaded on a hyper-threading machine and would also lend itself to full message-passing implementation on a parallel computing cluster. In spite of this no such implementation has been addressed yet.

- **Extensibility:** Another interesting property of P system models and of the P system framework in general is the easy extensibility. The P system modelling framework presented in this work in easy extensible in two aspects.

  On the one hand, specific P system models can be easily extended and modified to incorporate new knowledge. In P system models molecular interactions are represented separately in rewriting rules. In this respect, when newly discovered molecular interactions need to be integrated in an old model all the modifications are limited to add new rewriting rules describing the found interactions. When an hypothetical interaction already incorporated in a P system model is proved to be wrong the P system model is changed by simply removing the rewriting rule used to describe this interaction.

  On the other hand, the whole P system modelling framework itself is easily extensible. This thesis itself present an example of this. The application of Gillespie's theory of stochastic kinetics to the compartmentalised structure of P systems were achieved easily. Furthermore, the use of strings to represent some molecular entities and the use of rewriting rules on multisets of objects and strings were incorporated to the P system modelling framework without producing much disturbance.

  In fact the knowledge obtained during the development of this thesis has made obvious the necessity of extending the P system modelling framework to incorporate more quantitative information of the components of cellular systems. For instance, a future direction consists of incorporating some attributes to the objects representing molecules and to the membranes describing compartments. Some of the proposed attributes so far are interaction capabilities, coordinates, shape, temperature, etc.

Finally, the use of modules introduced in this work will provide a more easier extensibility in P system specifications which can be obtained by combining simple building blocks represented using P system modules. In this respect, the study of more complex P systems modules representing modular patterns in cellular systems remains a promising and challenging future work open in this thesis.

Summing up the work developed in this thesis was intended to adapt the original P system framework to the modelling of cellular systems by providing them with stochastic dynamics and with the necessary elements to specify the most relevant interactions in cellular systems. This approach has been proved to be a *good* modelling framework fulfilling to a large extent the four generally required properties in modelling frameworks of cellular systems, relevance, understandability, computability and extensibility.

Nevertheless, this thesis has also uncovered several open problems opening various future directions to extend the proposed formalism to be able to specify and model in a more accurate manner cellular systems.

# Bibliography

[1] Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J. D. (1994) *Molecular Biology of the Cell*, Garland Publishing, New York, USA.

[2] Alimonti, J.B., Ball, T.B., Fowke, K.R. Mechanisms of CD4+ T lymphocyte cell death in human immunodeficiency virus infection and AIDS. *Journal of General Virology*, **84** (2003), 1649–1661.

[3] Alon, U. (2006). *An Introduction to Systems Biology.* Mathematical and Computational Biology Series. Chapman & Hall/Crc.

[4] Alon, U. (2007) Network Motifs: Theory and Experimental Approaches, *Nature Reviews Genetics*, **8**, 450 – 461.

[5] Alur, R., Henzinger, T.A. (1999) Reactive Modules, *Formal Methods in System Design*, **15** 7–48.

[6] Ardelean, I.I., Cavaliere, M. (2003) Modelling Biological Processes by Using a Probabilistic P system Software, *Natural Computing*, **2** (2), 173 – 197.

[7] Andrei, O., Ciobanu, G., Lucanu, D. (2005) Executable Specifications of P Systems, *LNCS*, **3365** 126 – 145.

[8] Arkin, A., Ross, J., McAdams, H. H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells, *Genetics*, **149**, 1633 – 1648.

[9] Arnold, L., Theodosopulu, M. (1980) Deterministic limit of the stochastic model of chemical reactions with diffusion, *Adv. Applied Prob.*, **12**, 367 – 379.

[10] Aziz, A., Sanwal, K., Singhal, V., Brayton, R. (1996) Verifying continuous time Markov chains, *LNCS*, **1102**, 269–276.

[11] Baier, C., Kwiatkowska (1998) Model checking for a probabilistic branching time logic with fairness, *Distributed Computing*, **11** (3), 125 – 155.

[12] Barabási, A.L., Oltvai, Z. (2004) Network biology: Understanding the cells functional organization. *Nat. Rev. Genetics*, **5**, 101 – 113.

[13] Batt, G., Casey, R., de Jong, H., Geiselmann, J., Gouzé, J.L., Page, M., Ropers, D., Sari, T., Schneider, D. (2005) Qualitative analysis of the dynamics of genetic regulatory networks using picewise-linear models. In Pecou, E., Martinez, S., Maass, A., editors, *Mathematical and Computational Methods in Biology.* Paris: Editions Hermann.

[14] Batt, G., Ropers, D., de Jong, H., Geiselmann, J., Mateescu, R., Page, M., Schneider, D. (2005) Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in Escherichia coli, *Bioinformatics*, **21**, i19–i28.

[15] Benner, S.A., Sismour, A.M. (2005) Synthetic biology, *Nature Review, Genetics*, **6**, 533–543.

[16] Bernardini, F., Gheorghe, M., Krasnogor, N., Giavitto, J.L. (2005) On Self-Assembly in Population P Systems, *Lecture Notes in Computer Science*, **3699**, 46 – 57.

[17] Bernardini, F., Gheorghe, M., Krasnogor, N. (2006) Quorum sensing P systems, *Theoretical Computer Science*, **371**, (1–2), 20 – 33.

[18] Bernardini, F., Brijder, R., Rozenberg, G., Zandron, C. (2007) Multiset Based Self-Assembly of Graphs, *Fundamenta Informaticae*, **75**(1–4),49 – 75.

[19] Bernardini, F., Manca, V. (2003) P Systems with Boundary Rules, *Lecture Notes in Computer Science*, **2597**,107–118.

[20] Besozzi, D., Ciobanu, G. (2005) A P Systems Description of the Sodium-Potassium Pump, *LNCS*,**3365**, 210 –223.

[21] Bhalla, U.S., Iyengar, R., Emergent properties of networks of biological signaling pathways. *Science*, **283** (1999), 381–387. Not cited

[22] Bianco, A., Alfaro, L. (1995) Model checking of probabilistic and nondeterministic systems, *LNCS*, **1026**, 499 – 513.

[23] Bianco, L., Fontana, F., Manca, V. (2006) P systems with Reactions Maps, *Int. J. Found. Comput. Sci.*, **17** (1), 27 – 48.

[24] Blossey, R., Cardelli, L., Phillips, A. (2006) A Compositional Approach to the Stochastic Dynamics of Gene Networks, *Transactions on Computational Systems Biology IV, LNBI* **3939**, 99-122.

[25] Bower, J., Bolouri, H. (2001) *Computationl Modelling of Genetic and Biochemical Networks.* MIT Press.

[26] Cardelli, L. (2005). Brane calculi: interactions of biological membranes, *Lecture Notes in Bioinformatics*, **3082** 257 - 278.

[27] Cheruku, S., A. Păun, F.J. Romero, M.J. Pérez–Jiménez, O.H. Ibarra (2006) Simulating FAS–induced apoptosis by using P systems. *Proceedings of the First International Conference on Bio–Inspired Computing: Theory and Applications*, Wuhan, China, September, 18–22.

[28] Ciobanu, G., Pan, L., Păun, Gh., Pérez-Jiménez, M.J. (2007) P Systems with Minimal Parallelism, *Theoretical Computer Science*, **378** (1), 117–130.

[29] Dang, Z., Ibarra, O.H. (2005) On One-membrane P systems Operating in Sequential Mode, *Int. J. Found. Comput. Sci.*, **16** (5) 867–881.

[30] Danos V, Laneve C. (2004) Formal molecular biology. *Theor. Comput. Sci.*, **325**(1), 69 – 110.

[31] Elf, J., Ehrenberg, M. (2004) Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases, *Systems Bio.*, **2**, 230 – 236.

[32] El-Masri, H.A., Portier, C.J. (1999) Replication potential of cells via the protein kinase C-MAPK pathway: Application of a mathematical model, *Bulletin of Mathematical Biology*, **61**, 379–398.

[33] Elowitz, M.B., Levine, A.J., Siggia, E.D., Swain, P.S. (2002) Stochastic Gene Expression in a Single Cell, *Science*, **297**, 1183 – 1186.

[34] Fargerströn, T., James, G., James, S., Kjelleberg, S., Nilsson, P. (2000) Luminescence Control in the Marine Bacterium Vibrio fischeri: An Analysis of the Dynamics of lux Regulation. *J. Mol. Biol.* **296**, 1127–1137.

[35] Fontana, F., Bianco, L., Manca, V. (2005) P Systems and the Modelling of Biochemical Oscillations. In Freund, R., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) *Membrane Computing, Sixth International Workshop, WMC6, Vienna, Austria,* volume 3850 of *Lecture Notes in Computer Science*, pages 199 – 208. Springer Verlag.

[36] Freund, R. (2005) P systems Working in the Sequential Mode on Arrays and Strings, *Int. J. Found. Comput. Sci.*, **16** (4), 663 – 682.

[37] Funahashi, A., Morohashi, M., Kitano, H. (2003) Cell Designer: a Process Diagram Editor for Gene-regulatory and Biochemical Networks, *Biosilico*, **1**, 159 – 162.

[38] Gardiner, C., Steyn-Ross, M. (1984) Adiabatic elimination in stochastic systems. II. Application to reaction diffusion and hydrodynamic-like systems, *Phys. Rev. A.*, **29**, 2823 – 2833.

[39] Gheorghe,M., Păun, Gh. (2007) Computing by self-assembly: DNA molecules, polynominoes, cells. In N. Krasnogor, S. Gustafson, D. Pelta, and J.L. Verdegay, editors, Systems Self-Assembly: Multidisciplinary Snapshots, Studies in Multidisciplinarity, Elsevier.

[40] Gillespie, D.T. (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, *J. Comp. Phys.*, **22**, 403 – 434.

[41] Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.*, **81**, 2340 – 2361.

[42] Gillespie, D.T. (1992) A rigorous derivation of the chemical master equation, *Physics A*, **188**, 404–425.

[43] Gillespie, D.T. (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, **115**, 1716 – 1733.

[44] Gillespie, D.T., Petzold, L. (2003) Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.*, **119**, 8229 – 8234.

[45] Girault, C., Valk., R. (2003) *Petri Nets for Systems Engineering.* Springer Verlag.

[46] Goss, P.J.E., Peccoud, J. (1998) Quantitative modelling of stochastic system in molecular biology by using stochastic Petri nets, *Proc. Natl. Acad. Sci. USA*, **95**, 6750 – 6755.

[47] Greenberg, E.P., Kaplan, H.B. (1985). Diffusion of Autoinducer Is Involved in Regulation of the Vibrio fischeri Luminescence System. *Journal of Bacteriology* **163**(3), 1210–1214

[48] Greenberg, E.P., Kaplan, H.B. (1997). Overproduction and Purification of the luxR Gene Product: Transcriptional Activator of the Vibrio fischeri Luminescence System. *Proc. Natl. Acad. Sci.* **84**, 6639–6643

[49] Greenberg, E.P., Lupp, C.; Ruby, E.G., Urbanowski, M. (2003). The Vibrio fischeri Quorum-sensing System ain and lux Sequentially Induce Luminescence Gene Expression and Are Important for Persistence in the Squid Host. *Molecular Microbiology*, **50**(1) 319–331

[50] Hansson, H., Jonsson, B. (1994) A logic for reasoning about time and reliability. *Formal Aspects of Computing*, **6** (5), 512 – 535.

[51] Hardie, K.R., Williams, P., Winzer, K. (2002). Bacterial cell-to-cell communication: sorry, can't talk now, gone to lunch. *Current Opinion in Microbiology*, **5**, 216–222

[52] Harel, D. (1987). Statecharts: A visual formalism for Complex Systems. *Science of Computer Programming*, **8** (3), 231274.

[53] Hartwell, L.H., Hopfield, J.J., Leibler, S. Murray, A.W. (1999) From Molecular to Modular Cell Biology, *Nature*, **402**, C47 – C52.

[54] Holcombe, M., Gheorghe, M., Talbot, N. A hybrid machine model of rice blast fungus, Magnaphorte Grisea. *BioSystems*, **68**, 2–3 (2003), 223–228.

[55] Hua, F., Cornejo, M., Cardone, M., Stokes, C., Lauffenburger, D. Effects of Bcl-2 Levels on FAS Signaling-Induced Caspase-3 Activation: Molecular Genetic Tests of Computational Model Predictions. *The Journal of Immunology*, **175**, 2 (2005), 985–995 and correction **175**, 9 (2005), 6235–6237.

[56] Hucka, M., A. Finney, H. Sauro, H. Bolouri, J. Doyle, H. Kitano, A. Arkin, B. Bornstein, D. Bray, A. Cornish-Bowden, A. Cuellar, S. Dronov, E. Gilles, M. Ginkel, V. Gor, I. Goryanin, W. Hedley, T. Hodgman, J. Hofmeyr, P. Hunter, N. Juty, J. Kasberger, A. Kremling, U. Kummer, N. Le Novere, L. Loew, D. Lucio, P. Mendes, E. Minch, E. Mjolsness, Y. Nakayama, M. Nelson, P. Nielsen, T. Sakurada, J. Schaff, B. Shapiro, T. Shimizu, H. S. andJ Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524531.

[57] Huang, K.C., Meir, Y., Wingreen, N.S. (2003) Dynamic structures in Escherichia coli: Spontaneous formation of MinE rings and MinD polar zones. *Proc. Natl. Acad. Sci. USA*, **100**, 12724 – 12728.

[58] Huber, R.E., Wallenfels, K., Kurz, G. (1975). Action of $\beta-$galactosidase on allolactose. *Canad. J. Biochem.*, **53**, 1035–1038.

[59] Ideker, T., T. Galitski, and L. Hood (2001). A new approach to decoding life: systems biology. *Annual Review of Genomics and Human Genetics*, **2**, 34372.

[60] Ionescu, M., Sburlan, D. (2004) On P systems with Promoters/Inhibitors, *J. UCS*, **10** (5) 581 – 599.

[61] Ionescu, M., Păun, Gh., Yokomori, T. (2006) Spinking Neural P systems, *Fudam. Inform.* **71** (2-3) 279 – 308.

[62] Jaatela, M. (2004) Multiple cell death pathways as regulators of tumour initiation and progrssion. *Oncogene*, **23**, 2746–2756.

[63] Jackson, D., Holcombe, M., Ratnieks, F. (2004) Trail geometry gives polarity to ant foraiging networks. *Nature* **432**, 907–909.

[64] Kell, D.B. (2002) Genotype-phenotype mapping: genes as computer programs, *Trends in Genetics*, **18**, 11, 555 – 559.

[65] Kell, D.B. (2005) Metabolomics, machine learning and modelling: towards an understanding of the language of cells, *Biochemical Society Transactions*, **33**, 520 – 524.

[66] Kennell, D., Riezman, H. (1977). Transcription and Translation Initiation Frequencies of the Escherichia coli lac Operon. *J. Mol. Biol.*, **114**, 1–21.

[67] Kerr, J.F., Wyllie, A.H., Currie, A.R. (1972) Apoptosis: a basic biological phenomenon with wide-ranging implications in tissue kinetics. *British Journal Cancer*, **26**, 239.

[68] Kholodenko, B.G, Demin, O.V., Moehren, G., Hoek, J.B. (1999) Quantification of short term signalling by the Epidermal Growth Factor Receptor, *J. Biol. Chem.* **274**, 30169 – 30181.

[69] Kierzek, A.M., Zaim, J., Zielenkiewicz, P. (2001). The Effect of Transcription and Translation Initiation Frequencies on the Stochastic Fluctuations in Prokayotic Gene Expression. *The Journal of Biological Chemistry*, **276**, 11, 8165–8172.

[70] Kitano, H. (Ed.) (2001). *Foundations of Systems Biology.* MIT Press.

[71] Kitano, H., Matsuoka, Y., Funahashi, A., Oda, K. (2005) Using process diagrams for the graphical representation of biological networks, *Nature Biotechnology*, **23**(8), 961 – 966.

[72] Koch, A.J., Minhardt, H. (1994) Biological pattern formation: from basic mechanism to complex structures, *Rev. Mod. Phys.*, **66**, 1481 – 1507.

[73] Kiatkoska, M., Norman, G., Parker, D. (2005) Probabilistic Model Checking in Practice: Case Studies with PRISM, *ACM SIGMETRICS Performance Evaluation Review*, **32** (4), 16 – 21.

[74] Kuttler, C. (2006) Simulating Bacterial Transcription and Translation in a Stochastic Pi-calculus, *Transactions on Computational Systems Biology VI, LNBI*, **4220**, 113 – 149.

[75] Lipton, P. (2005) Testing hypotheses: Prediction and predjudice, *Science*, **307**, 219 – 221.

[76] Lodish, H., Berk, A., Zipursky, S.L., Matsudaira, P., Baltimore, D., Darnell, J. (2003) Molecular Cell Biology. Fith Edition, Freeman.

[77] Lolkema, J.S., Carrasco, N., Kaback, R. (1991). Kinetic analysis of lactose exchange in proteoliposomes reconstituted with purified lac permease. *Biochemistry*, **30**, 1284 – 1290.

[78] McAdams, H., Arkin, A. (1997) Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*, **94**, 814 – 819.

[79] McAdams, H., Arkin, A. (1999) It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Gnet.*, **15**, 65 – 69.

[80] Milner, R. (1999) Communication and Mobile Systems: The $\pi$-calculus. *Cambridge University Press.*

[81] Moehren G., Markevich, N., Demin, O., Kiyatkin, A., Goryanin, I., Hoek, J.B., Kholodenko, B.N. (2002) Temperature Dependence of the Epidermal Growth Factor Receptor Signaling Network Can Be Accounted for by a Kinetic Model, *Biochemistry* **41**, 306–320.

[82] Moghal, N., Sternberg, P.W. (1999) Multiple positive and negative regulators of signaling by the EGFR receptor. *Curr. Opin. Cell Biology*, **11**, 190–196. not cited

[83] Newman, M. (2003) The structure and function of complex networks. *SIAM Rev.*, **45**, 167 – 256.

[84] Nijhawan, D., Honarpour, N., Wang, X. (2000) Apotosis in neural development and disease. *Annual Reviews Neuroscience*, **23**, 73–87.

[85] Oltavi, Z.N., Milliman, C.L., Korsmeyer, S.J. (1993) Bcl-2 heterodimerizes in vivo with a conserved homolog, Bax, that accelerates programmed cell death. *Cell*, **74**, 4, 609–619.

[86] Orton, R.J., Sturm, O.E. et al. (2005) Computational modelling of the receptor-tyrosine-activated MAPK pathway, *Biochem. J.*, **392**, 249 – 261.

[87] Palsson, B. O. (2006). *Systems Biology: Properties of Reconstructed Networks.* Cambridge University Press.

[88] G. Păun (2002) *Membrane Computing: An Introduction.* Springer-Verlag.

[89] Păun, Gh. (2000) Computing with Membranes, *Journal of Computer and System Sciences*, **61**(1) 108 – 143.

[90] Păun, A., Păun, Gh. (2002) The Power of Communication: P systems with Symport/Antiport, *New Generation Comput.*, **20** (3) 295 – 306.

[91] G. Păun (2002) *Membrane Computing: An Introduction.* Springer-Verlag.

[92] Pérez-Jiménez, M.J., Romero-Campero, F.J. (2006) P Systems, a new computationl modelling tool for systems biology, *Transactions on Computational Systems Biology VI, LNBI*, **4220**, 176 – 197.

[93] Pescini, D., Besozzi, D., Mauri, G., Zandron, C. (2006) Dynamical probabilistic P systems, *International Journal of Foundations of Computer Science*, **17** (1) 183–195.

[94] Pogson, M., Smallwood, R., Qwarnstrom, E., Holcombe, M. (2006) Formal agent–based of intracellular chemical interactions. *BioSystems*, **85**, 1 37–45. not cited

[95] Priami, C., Regev, A., Shapiro, E., Silvermann, W. (2001) Application of a stochastic name-passing calculus to representation and simulation of molecular processes, *Information Processing Letters*, **80**, 25 – 31.

[96] Ptashne, M. (2004). *A Genetic Switch, Third Edition Phage Lambda Revisited.* Cold Spring Harbor Laboratory Press.

[97] Ptashne, M., Gann, A. (2002). *Genes and Signals.* Cold Spring Harbor Laboratory Press.

[98] Ray, T. S. (1994) An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, **1**(1/2): 195226. MIT Press.

[99] Reddy, V.N., Liebman, M.N., Maverovouniotis, M.L. (1996) Qualitative Analysis of Biochemical Reaction Systems, *Comput. Biol. Med.*, **26**, 1, 9 – 24.

[100] Regev, A., Panina, E.M., Silvermann, W., Cardelli, L., Shapiro, E. (2004) BioAmbients: an abstraction for biological compartments, *Theoretical Computer Science* **325** 141–167.

[101] Regev, A., Shapiro, E. (2004) The $\pi$-calculus as an abstraction for biomolecular systems, In Gabriel Ciobanu and Grzegorz Rozenberg, editors, *Modelling in Molecular Biology.* Springer Berlin.

[102] Regev, A., Shapiro, E. (2002) Cellular abstractions: Cells as computation, *Nature* **419** 343.

[103] Reisig, W. (1998) *Elements of Distributed Algorithms. Modelling and Analysis with Petri Nets.* Springer Verlag.

[104] Reisig, W., Rozenberg, G. (eds.) (1998) *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science.* Springer Verlag.

[105] Reisig, W., Rozenberg, G. (eds.) (1998) *Lectures on Petri Nets II: Applications*, volume 1492 of *Lecture Notes in Computer Science.* Springer Verlag.

[106] Rohwer, J.M., Meadow, N.D., Roseman, S., Westerhoff, H. (2000). Understanding Glucose Transport by the Bacterial Phosphoenolpyruvate: Glycose Phosphotransferase System on the Basis of Kinetic Measurements in Vitro. *The Journal of Biological Chemistry*, **275**, 45, 34909–34921.

[107] Romero-Campero, F.J., Pérez-Jiménez, M.J. Modelling Gene Expression Control Using P Systems: The Lac Operon, A Case Study, submitted.

[108] Romero-Campero, F.J., Pérez-Jiménez, M.J. A Model of the Quorum Sensing System in Vibrio Fischeri Using P Systems, submitted.

[109] Rosenfeld, N., Elowitz, M.B., Alon, U. (2002) Negative Autoregulation Speeds the Response Times of Transcription Networks, *J. Mol. Biol.*, **323**, 785 – 793.

[110] Ruby, E.G., McFall-Ngai, M.J. (1999) Oxygen-utilizing reactions and symbiotic colonisation of the squid light organ by vibrio fischeri, *Trends in Microbiology*, **7**, 414 – 420.

[111] Rutten, J., Kwiatkowska, M., Norman, G., Parker, D. (2004) *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems,* volume 23 of *CRM Monograph Series*, AMS.

[112] Savageau, M.A. (1976) *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*, Adison-Wesley, Reading, MA.

[113] Savageau, M.A. (1974) Comparison of classical and auto-genous systems of regulation in inducible operons, *Nature*, **252**, 546 – 549.

[114] Scaffidi, C., Fulda. S., Srinivasan, A., Friesen, C., Li, F., Tomaselli, K.J., Debatin, K.M., Krammer, P.H., Peter, M.E. (1998) Two CD95 (APO-1/Fas) signaling pathways. *The Embo Journal*, **17**, 1675–1687.

[115] Schoeberl, B., Eichler–Jonsson, C., Gilles, E.D., Muller, G. (2002) Computational Modeling of the Dynamics of the MAP Kinase Cascade Activated by Surface and Internalized EGF Receptors, *Nature Biotechnology*, **20**, 4 370–375.

[116] Shmulevich, I., Dougherty, E., Kim, S., Zhang, W. (2002) Probabilistic Boolean Networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**, 261–274.

[117] Starbuck, C., Lauffenburger, D.A. (1992) Mathematical model of the effects of Epidermal Growth Factor Receptor trafficking dynamics on Fibroblast proliferation responses, *Biotehcnol. Prog*, **8**, 132 – 143.

[118] Stevens, A.M., Greenberg, E.P. (1997). Quorum Sensing in Vibrio fischeri: Essential Elements for Activation of the Luminescence Genes. *Journal of Bacteriology* **179**(2) 557–562.

[119] Sydor, J.R. (1998) Transient kinetic study on the interactions of Ras-binding domain of c-Raf-1 reveals rapid equilibrium of the complex, *Biochemistry* **37**, 14292 – 14299.

[120] Szallasi, Z., J. Stelling, and V. Periwal (Eds.) (2006). *System modeling in cellular biology.* MIT press.

[121] Terrazas, G., Krasnogor, N., Gheorghe, M., Bernardini, F., Diggle, S., Camara, M. (2005) An Environment Aware P-System Model of Quorum Sensing. *Lecture Notes in Computer Science*, **3526**, 479 – 485.

[122] Thieffry, D., Huerta, A.M., Perez-Rueda, E., Collado-Vides, J. (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli, *Bioessays*, **20**, 433 – 440.

[123] Turing, A.M. (1952) The chemical basis of morphogenesis. *Phil. Trans. Roy. Soc.*, **237**, 37 – 72.

[124] Vayttaden, S., Ajay, S., Bhalla, U. (2004) A spectrum of models of signalling pathways. *ChemBioChem,* **5**, 1365 – 1374.

[125] Van Kampen, N.G. (1992) *Stochastics processes in Physics and Chemistry*, Elsevier Science B.V., Amsterdam, The Netherlands.

[126] Walker, D.C., Southgate, J., Hill, G., Holcombe, M., Hose, D.R., Wood S.M., MacNeil, S., Smallwood, R.H. (2004) The epitheliome: modelling the social behaviour of cells. *BioSystems*, **76**, 1–3, 89–100.

[127] Wang, K., Yin, X.M., Chao, D.T., Milliman, C.L., Korsmeyer, S.J. (1996) BID: a novel BH3 domain-only death agonist. *Genes & Development*, **10**, 2859–2869.

[128] Waters, C.M., Bassler, B.L. (2005) Quorum sensing: Cell-to-cell communication in bacteria. *Annual Review of cell and Developmental Biology*, **21**, 319 – 346.

[129] Wells, A. (1999) EGFR–receptor. *Int. Journal Biochem. Cell Biology*, **31**, 637–643.

[130] Wiley, H.S., Shvartsman, S.Y., Lauffenburger, D.A. (2003) Computational modeling of the EGFR–receptor system: A paradigm for systems biology. *Trends in Cell Biology*, **13**, 1, 43–50.

[131] Wong, P., Gladney, S., Keasling, J.D. (1997). Mathematical Model of the lac Operon: Inducer Exclusion, Catabolite Repression, and Diauxic Growth on Glucose and Lactose. *Biotechnol. Prog.*, **13**, 132–143

[132] Nottingham Quorum Sensing Web Site `http://www.nottingham.ac.uk/quorum/`

[133] PRISM Web Site: `www.cs.bham.ac.uk/~dxp/prism/`

[134] P Systems Modelling Framework Web Site:
`www.dcs.shef.ac.uk/~marian/PSimulatorWeb/P_Systems_applications.html`

[135] SBML Site: `http://sbml.org`