

Process algebras in systems biology

Federica Ciocchetta and Jane Hillston

Laboratory for Foundations of Computer Science, The University of Edinburgh,
Edinburgh EH9 3JZ, Scotland
{fcioच्che, jeh}@inf.ed.ac.uk

Abstract. In this chapter we introduce process algebras, a class of formal modelling techniques developed in theoretical computer science, and discuss their use within systems biology. These formalisms have a number of attractive features which make them ideal candidates to be intermediate, formal, compositional representations of biological systems. As we will show, when modelling is carried out at a suitable level of abstraction, the constructed model can be amenable to analysis using a variety of different approaches, encompassing both individuals-based stochastic simulation and population-based ordinary differential equations. We focus particularly on Bio-PEPA, a recently defined extension of the PEPA stochastic process algebra, which has features to capture both stoichiometry and general kinetic laws. We present the definition of the language, some equivalence relations and the mappings to underlying mathematical models for analysis. We demonstrate the use of Bio-PEPA on two biological examples.

1 Introduction

In recent years there has been increasing interest in the application of process algebras in the modelling and analysis of biological systems [60, 26, 28, 58, 19, 49, 14]. Process algebras have some interesting properties that make them particularly useful in describing biological systems. First of all, they offer *compositionality*, i.e. the possibility of defining the whole system starting from the definition of its subcomponents. Secondly, process algebras give a formal representation of the system avoiding ambiguity. Thirdly, biological systems can be abstracted by concurrent systems described by process algebras: species may be seen as processes that can interact with each other and reactions may be modelled using actions. Finally, different kinds of analysis can be performed on a process algebra model. These analyses provide conceptual tools which are complementary to established techniques: it is possible to detect and correct potential inaccuracies, to validate the model and to predict its possible behaviours.

The original work on process algebra modelling of biochemical pathways by Regev *et al.* was based on the abstraction “*processes as molecules*” [60]. This abstraction has proven to be fruitful and highly influential with most of the subsequent work based on the same abstraction. However, it is not without its drawbacks. It takes an inherently individual-based view of the system (i.e. views the system at the level of individual molecules) which has the consequence that the state space, under all but the smallest examples, will be prohibitively large and amenable only to analysis via simulation.

In recent years Calder *et al.* [14, 15] have been experimenting with alternative abstractions using the stochastic process algebra, PEPA, which was originally defined for

performance analysis of computer systems [42]. Two different approaches have been proposed: one based on reagents (the so-called *reagent-centric view*) and another based on pathways (*pathway-centric view*). In both cases the species concentrations are discretized into levels, each level abstracting an interval of concentration values. In the reagent-centric view the PEPA sequential components represent the different concentration levels of the species. In this approach the abstraction is “*processes as species*” and not “*processes as molecules*”. In the pathway-centric approach adopts an alternative abstract view: the processes represent sub-pathways. Here multiple copies of components represent levels of concentration. The two views have been shown to be equivalent [15].

Even though PEPA and other stochastic process algebras have proved useful in studying signalling pathways, they do not readily allow us to represent all the features of biological networks. The main difficulties are the definition of *stoichiometric coefficients* (i.e. the coefficients used to show the quantitative relationships of the reactants and products in a biochemical reaction) and the representation of *kinetic laws*. Concerning stoichiometry, in the reagent-centric view of PEPA stoichiometry is not represented explicitly. Furthermore, in other process algebras it is not possible to render interactions with more than two reactants as biochemical interactions are abstracted by pairwise communications. This is justified by appeal to Gillespie’s stochastic simulation algorithm which, in the original version, assumes elementary (i.e. monomolecular and bimolecular) reactions. However, it is often convenient to model at a more abstract level, where reactions involving more than two species are common (e.g. Michaelis-Menten). In terms of kinetic laws, PEPA and other process algebras consider elementary reactions with constant rates (*mass-action* kinetic laws). The problem of extending to the domain of kinetic laws beyond basic mass-action (hereafter called *general kinetic laws*) is particularly relevant, as these kinds of reactions are frequently found in the literature as abstractions of complex situations whose details are unknown. Reducing all reactions to the elementary steps is complex and often impractical. In the case of process algebras such as π -calculus the assumption of elementary reactions is motivated by the fact that they rely on Gillespie’s stochastic simulation for analysis. Some recent works have extended the approach of Gillespie to deal with complex reactions [1, 17] but these extensions are yet to be reflected in the work using process algebras. Previous work concerning the use of general kinetic laws in process algebras and formal methods was presented in [9, 20]. These are discussed in Section 2.3.

In this chapter we give a tutorial introduction to Bio-PEPA, a new language for the modelling and analysis of biochemical networks. A preliminary version of the language was proposed in [22], with full details presented in [23]. Here, in addition to defining the language we illustrate its use with a number of examples.

Bio-PEPA is based on the reagent-centric view in PEPA, modified in order to represent explicitly some features of biochemical models, such as stoichiometry and the role of the different species in a given reaction. A major feature of Bio-PEPA is the introduction of functional rates to express general kinetic laws. Each action type represents a reaction in the model and is associated with a functional rate.

The idea underlying our work is represented schematically in the diagram in Fig. 1. The context of application is biochemical networks. Broadly speaking, biochemical net-

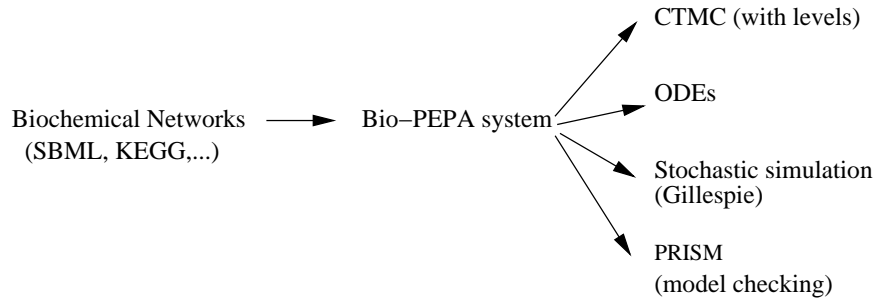


Fig. 1. Schema of the Bio-PEPA framework

works consist of some chemical species, which interact with each other through chemical reactions. The dynamics of reaction are described in terms of some kinetic laws. The biochemical networks can be obtained from databases such as *KEGG* [46, 45] and *BioModels Database* [8, 56]¹. From the biological model, we develop the Bio-PEPA specification of the system. This is an *intermediate, formal, compositional* representation of the biological model. At this point we can apply different kinds of analysis, including stochastic simulation [36], analysis based on ordinary differential equations, numerical solution of continuous time Markov chains (CTMC) and stochastic model checking using PRISM [61, 40]. It is worth noting that each of these analyses can help in understanding the system. The choice of one or more methods depends on the context of application [68]. There exist some relations between the different kinds of analysis. It is well-known that the results of stochastic simulations tend to the ODEs solution when the number of elements is relatively high. Similarly, it is shown in [35] that the numerical solution of the CTMC with levels (derived from the PEPA pathway-centric view) tends to the solution of the ODEs when the number of levels increases.

The rest of the chapter is organised as follows. In the next section we outline our modelling domain, biochemical networks, and discuss the development of process algebras and their use to model biological systems. The recent development of Bio-PEPA has been informed by earlier work using the PEPA process algebra so this work is also briefly presented. In Section 3 we give a detailed account of Bio-PEPA; its syntax, semantics, equivalence relations and analysis techniques. The use of the language is illustrated in Section 4, in which the translation of two biological models into Bio-PEPA and their subsequent analysis is described. Finally, in Section 5 we present some conclusions and future perspectives.

2 Background

In this section we outline the application domain of this tutorial before giving an introduction to process algebras. Subsequent sections will focus on one particular process

¹ The *BioModels Database* is a collection of SBML models. SBML is a widely used XML-based format for representing models of biochemical reaction networks.

algebra, Bio-PEPA, but here we aim to give a broad overview of the class of formalisms known as process algebras.

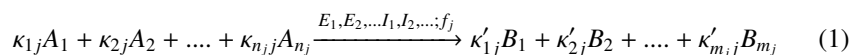
2.1 Application domain: biochemical networks

The application domain of this tutorial concerns biochemical networks, such as those collected in the Biomodels Database [56] and KEGG [46].

A biochemical system \mathcal{M} is composed of:

1. a set of *compartments* \mathbf{C} . These represent the locations of the various species;
2. a set of *chemical species* \mathbf{S} . These species may be genes, proteins, etc.;
3. a set of *reactions* \mathbf{R} . We consider *irreversible reactions*. Reversible reactions are decomposed as a pair of forward and inverse reactions.

The general form of an irreversible reaction j is given by:



where A_h , $h = 1, \dots, n_j$, are the reactants, B_l , $l = 1, \dots, m_j$, are the products, E_v are the enzymes and I_u , the inhibitors. Enzymes and inhibitors are represented differently from the reactants and products. Their role is to enhance or inhibit the reaction, respectively. We call species such as these, that are involved in a reaction without changing their concentration, *modifiers*. The parameters κ_{hj} and κ'_{lj} are the stoichiometric coefficients. These express the degree to which species participate in a reaction. The dynamics associated with the reaction is described by a kinetic law f_j , depending on some parameters and on the concentrations of some species.

The best known kinetic law is *mass-action*: the rate of the reaction is proportional to the product of the reactants' concentrations. In published models it is common to find *general kinetic laws*, which describe approximations of sequences of reactions. They are useful when it is difficult to derive certain information from the experiments, e.g. the reaction rates of elementary steps, or when there are different time-scales for the reactions. Generally these laws are valid under some conditions, such as the *quasi-steady-state assumption (QSSA)*. This describes the situation where one or more reaction steps may be considered faster than the others and so the intermediate elements can be considered to be constant. There is a long list of kinetic laws; for details see [65].

2.2 Overview of process algebras

Process algebras are calculi that were originally motivated by problems associated with concurrent computer systems [51, 43]. The objective was to specify and formally reason about such systems. In subsequent years process algebras have been used extensively to describe complex systems characterized by *concurrency*, *communication*, *synchronization* and *nondeterminism*. Process algebras offer several attractive features. The most important of these are *compositionality*, the ability to model a system as the interaction of its subsystems, *formality*, the ability to give a precise meaning to all terms in

the language, and *abstraction*, the ability to build up complex models from detailed components but disregarding internal behaviour when it is appropriate to do so.

The most widely known process algebras are Milner's Calculus of Communicating Systems (CCS) [51] and Hoare's Communicating Sequential Processes (CSP) [43]. Process algebras are typically defined by a simple syntax and semantics. The semantics may be given by axioms or inference rules expressed in an operational way [57]. A system is defined as a collection of agents which execute atomic actions. Some operators are introduced for combining the primitives. For instance in CCS the main operators are:

prefix	$a.P$, after action a the agent becomes a P
parallel composition	$P \mid Q$, agents P and Q proceed in parallel
choice	$P + Q$, the agent behaves as P or Q
restriction	$P \setminus M$, the set of actions M may not occur
relabelling	$P[a_1/a_0, \dots]$, in this agent label a_1 is renamed a_0
the null agent	0 , this agent cannot act (deadlock)

One of the main features of process algebras is the possibility to express communication between two processes. In some cases, such as CCS [51] and the π -calculus [52], a communication between two parallel processes is enabled when one process can perform an action a (receive) and the other process can perform the complementary action \bar{a} (send). So the actions must be complementary (input-output) and share the same name (a in the case considered). The resulting communication has the distinguished label τ , which indicates an internal (invisible) action. A distinguishing feature of π -calculus with respect to CCS is the possibility to represent *name-passing*: communicating processes can exchange names over channels and consequently they may change their interaction topology.

The communication mechanism in CSP is different from the one described above as there is no notion of complementary actions. In CSP, two agents communicate by simultaneously executing actions with the same label. Since during the communication the joint action remains visible to the environment, it can be reused by other concurrent processes so that more than two processes can be involved in the communication (*multiway synchronisation*).

The analysis of the behaviours of the model, represented in a formal language, is generally produced through a *Labelled Transition System (LTS)* derived from the operational semantics. This may be regarded as a *derivative tree* or *graph* in which language terms form the nodes and transitions are the arcs. This structure is a useful tool for reasoning about agents and the systems they represent: two agents are considered to be equivalent if they are observed to perform exactly the same actions and the resulting agents are also equivalent. Strong and weak forms of equivalence are defined depending on whether the internal actions of an agent are deemed to be observable.

In CCS and CSP, since the objective is qualitative analysis rather than quantitative, time and uncertainty are abstracted away. In the last two decades, various suggestions for incorporating time and probability into these formalisms have been investigated (see [54] for an overview of process algebras with time). For example, Temporal CCS (TCCS) [53] extends CCS with *fixed delays* and *wait-for synchronisation* (asynchronous waiting). Note that most of the timed extensions, including TCCS, retain the assumption that actions are instantaneous and regard time progression as orthogonal to the

activity of the system. Probabilistic extensions of process algebras, such as PCCS [44], allow uncertainty to be quantified using a probabilistic choice combinator. In this case a probability is associated with each possible outcome of a choice.

In the early 1990s several stochastic extensions of process algebra (*stochastic process algebras* or SPAs), were introduced. The motivation for SPA was performance modelling and quantification, in the form of random variables characterising the duration of actions, was added to models. In most cases, the random variables are assumed to be exponentially distributed and a rate is added to each prefix to represent the parameter of the exponential distribution that describes the dynamic behaviour of the associated action. A *race condition* is then assumed to resolve conflicts: all the activities that are enabled in a given state compete and the fastest one succeeds. The choice of the exponential distribution means that each process algebra model is associated with a continuous time Markov chain (CTMC) [42]. It is then possible to carry out performance analysis based on this underlying mathematical model. Some examples of SPA are TIPP [38], EMPA² [4, 5], PEPA [42], SPADE³ [67] and Stochastic π -calculus [59].

2.3 Process algebras in systems biology

Recently, as a response to the need to model the dynamics of complex biological systems, there have been several applications of process calculi in systems biology [60, 62, 28, 58, 27, 18, 19, 14]. These techniques are appropriate for formally describing and analysing a biological system as a whole and for reasoning about protein/gene interactions. Indeed, there is a strong correspondence between concurrent systems described by process algebras and biological ones: biological entities may be abstracted as processes that can interact with each other and reactions may be modelled as actions.

Process calculi have several properties that make them useful for studying biological systems:

- they allow the formal specification of the system;
- they offer different levels of abstraction for the same biological system;
- they can make interactions explicit, in particular biological elements may be seen as entities that interact and evolve;
- they support modularity and compositionality;
- they provide well-established techniques for reasoning about possible behaviours. They may be used not only for the simulation of the system, but also for the verification of formal properties and for behavioural comparison through equivalences.

Several process calculi have been proposed in biology. Each of them has different properties able to render different aspects of biological phenomena. They may be divided into two main categories:

- calculi defined originally in computer science and then applied in biology, such as the biochemical stochastic π -calculus [60], CCS-R [28] and PEPA [42];
- calculi defined specifically by observing biological structures and phenomena, such as BioAmbients [19], Brane Calculi [18], κ -calculus [27], and Beta-binders [58].

² Originally called simply MPA.

³ Originally called CCS+.

One of the first process algebras used in systems biology is the biochemical π -calculus [60], a variant of the π -calculus defined to model biological systems. The underlying idea of application of the π -calculus to biology is the *molecule-as-computation* abstraction [63, 60]: each biological entity and interaction is associated with an agent specification in the calculus. Specifically, molecules are modelled as processes, interaction capabilities as channels, interactions as communications between processes, modifications as state and channel changes and, finally, compartments and membranes as restrictions. Two stochastic simulation tools based on Gillespie [36] have been defined (BIOSPI [6] and SPIM [66]), various applications have been shown [50, 26, 49, 21] and some modified versions have been proposed (e.g. SPICO[48] and Sp@ [69]).

CCS-R [28] is a variant of CCS with new elements which allow the capture of reversibility. The interactions are described in terms of binary synchronised communications, similarly to π -calculus. It was motivated by modelling reversible reactions in biochemistry. The successor of CSS-R is the Reversible CCS (RCCS) [31]. This calculus allows processes to backtrack if this is in agreement with a notion of casual equivalence defined in the paper.

Beta-binders [58, 64] are an extension of the π -calculus inspired by biological phenomena. This calculus is based on the concept of *bio-process*, a box with some sites (*beta-binders*) to express the interaction capabilities of the element, in which π -calculus-like processes (*pi-processes*) are encapsulated. Beta-binders enrich the π -calculus with some constructs that allow us to represent biological features, such as the join between two bio-processes, the split of one bio-process into two, the change of the bio-process interface by hiding, unhiding and exposing a site. The Beta Workbench [64] is a collection of tools for the modelling, simulation and analysis of Beta-binders system. The BetaWB simulator is based on a variant of Gillespie's algorithm.

In most of the calculi considered it is not possible to represent all the features of biochemical networks. Generally the kinetic laws are assumed to be mass-action and reactions can have at most two reactants. Indeed these calculi refer to the standard Gillespie's algorithm for the analysis and this assumes elementary reactions (i.e. monomolecular or bimolecular) with constant rates. Furthermore, biological reactions are abstracted as communications/interactions between agents and in some process algebras such as π -calculus, CCS and Beta-binders, these actions are pairwise. Therefore multiple-reactant multiple-product reactions cannot be modelled in these calculi. In order to represent multiple-reactant multiple-product reactions, π -calculus and Beta-binders have been enriched with transactions [24, 25].

A first proposal to deal with general kinetic laws has been shown in [9]. The authors present a stochastic extension of Concurrent Constraint Programming (CCP) and show how to apply it to biological systems. Here each species is represented by a variable and the reactions are expressed by constraints on these variables. The domain of application is extended to any kind of reactions and the rate can be expressed by a generic function.

The possibility of representing general kinetic laws is also offered by BIOCHAM [20], a programming environment for modelling biochemical systems, which supports making simulations and querying the model in temporal logic. This language is not a process algebra, but it is based on a rule-based language for modelling biochemical systems, in which species are expressed by objects and reactions by reaction rules.

A similar approach is taken in the κ -calculus [27], based on the description of protein interactions. Processes describe proteins and their compounds. A set of processes model solutions and protein behaviour is given by a set of rewriting rules, driven by suitable side-conditions. The two main rules concern activation and complexation. A stochastic simulator for κ -calculus is described in [30]. A few applications are reported, as in [29].

Finally, some calculi have been defined to model compartments and membranes. Here we briefly describe Bio-ambients [19] and Brane calculi [18]. Bio-ambients are centered on *ambients*, bounded places where processes are contained and where communication may happen. Ambients can be nested and organised in a hierarchy. This hierarchy may be modified by suitable operations that have a biological interpretation. It is possible to have *enter* and *exit* primitives to move an ambient into or out of another ambient or a *merge* for merging two ambients together. Ambients contain compounds that interact via communication. Bio-ambients have been used to model compartments in *BIOSPI* [6]. A stochastic semantics for Bio-ambients has been formalized in [10]. There have been some applications, for instance [3].

In Brane calculi [18] a system consists of nested membranes, which are collections of actions. Membranes may shift, merge, break apart and may be replenished, leading to very expressive models, in which actions occur on membranes. Membranes may be seen as oriented objects that must obey some restrictions on orientation. In particular they must preserve *bitonality*, which requires nested membranes to have opposite orientations.

2.4 PEPA and biological systems

Performance Evaluation Process Algebra (PEPA) is a SPA originally defined for the performance modelling of systems with concurrent behaviour [42]. In PEPA each action is assumed to have a duration, represented by a random variable with a negative exponential distribution. We informally introduce the syntax of the language below. For more details see [42].

Prefix The basic term is the *prefix combinator* $(\alpha, r).S$. It denotes a component which has action of type α and an exponentially distributed duration with parameter r (mean duration $1/r$), and it subsequently behaves as S .

Choice The component $S + R$ represents a system which may behave either as S or as R . The activities of both S and R are enabled. The first activity to complete distinguishes one of them and the other is discarded.

Constant Constants are components whose meaning is given by a defining equation $C \stackrel{\text{def}}{=} S$. They allow us to assign names to patterns of behaviour associated with components.

Hiding In S/\mathcal{H} the set \mathcal{H} identifies those activities which can be considered internal or private to the component S .

Cooperation The term $P \bowtie_{\mathcal{L}} Q$ denotes cooperation between P and Q over the cooperation set \mathcal{L} , that determines those activities on which the cooperands are forced to synchronise. PEPA supports *multiway synchronisation* between components: the result of synchronising on an activity α is thus another α , available for further synchronisation. For action types not in \mathcal{L} , the components proceed independently and

concurrently with their enabled activities. In the context of performance evaluation the rate for the synchronised activities is the minimum of the rates of the synchronising activities.

Recently, PEPA has been applied to the modelling and analysis of signalling pathways. An initial study concerned the influence of the Raf Kinase Inhibitor Protein (RKIP) on the Extracellular signal Regulated Kinase (ERK) [14]. In [15] the PEPA representation of Schoeberl's model [32] involving the MAP kinase and EFG receptors is reported. The biological modelling in PEPA was motivated by a desire to experiment with more abstract modelling than that afforded by the *processes as molecules* mapping generally used in process algebra models. Indeed a *processes as species* mapping is applied instead. In [14] two different modelling styles were proposed, one based on the *reagent-centric view* and the other on the *pathway-centric view*. The former focuses on the variation in the concentrations of the reagents: the concentrations are discretized in levels, each level representing an interval of concentration values. The level l can assume values between 0 and N_{max} (maximum level). The pathway-centric style provides a different abstract view of the system and focuses on the subpathways. The two representations were shown to be equivalent [14]. In addition to the standard analysis offered by process algebras, in [13] a mapping from reagent-centric PEPA models to a system of ordinary differential equations (ODEs), has been proposed.

From the applications discussed above PEPA has been shown to be appropriate for the modelling of biological systems: it offers a high level of abstraction for the model and focuses on compositionality and on the interactions. Furthermore, by using PEPA as a modelling language it is possible to apply different kinds of analysis, not only stochastic simulation, but also differential equations and model checking.

However, not all the features of biochemical networks can be expressed using the present version of PEPA: general kinetic laws are not considered and stoichiometry is added by hand in the conversion of PEPA into ODEs. As observed above, with a few exceptions (e.g. [9]) and a few cases (dimerization), these features cannot be represented in other process algebras either. These and other problems motivated us to develop a new process algebra, Bio-PEPA, which is closely related to PEPA but better adapted to biological modelling.

3 Bio-PEPA: definition of the language

Our earlier experience using PEPA, and other stochastic process algebras, to model biochemical networks, developed insights which we then used in the definition of Bio-PEPA. We felt it was important to have a language which can represent all reactions in a straightforward way as well as handle stoichiometry and general kinetic laws. We retained the *reagent-centric view* previously used in PEPA models of biochemical pathways as this had been demonstrated to provide a flexible approach to modelling. For example, it is straightforward to capture reactions with any number of participants, something that is not readily captured in other process algebras such as the π -calculus. Moreover, once the model is constructed it is amenable to a variety of different analysis techniques.

We adopt a high level of abstraction similar to the one proposed in formalisms such as *SBML* [7], which have been widely adopted by biologists. Furthermore we make the following assumptions:

1. Compartments are *static*, i.e. compartments are not actively involved in the reactions — they are simply containers. The transport of a species from one compartment to another is modelled by introducing two distinct components for representing the species. The translocation is abstracted by a transformation of one species into another. Compartments are included in the definition of a Bio-PEPA system because the volume of the containing compartment can impact on reactions of a species.
2. Reactions are *irreversible reactions*. A reversible reaction is represented as a pair of irreversible reactions.

3.1 Discrete concentrations and granularity

Following the reagent-centric view, models are based not on individual molecules, but on discrete levels of concentration within a species: each component represents a species and it is parametric in terms of concentration level. Some advantages of this view are:

- It allows us to deal with uncertainty/incomplete information in the exact number of elements (semi-quantitative data);
- In a discrete state space representation the focus is on the concentration levels rather than the number of elements. This means that the state space is reduced as there are less states for each component.
- The population level view, in terms of continuously changing concentrations, and the individual level view, counting molecules, are both easily recovered from this abstract view.

This view was presented in [16]. The authors focused on the case of reactions with mass-action kinetics and stoichiometry equal to one for all the reactants and products. The granularity of the system has been expressed in terms of the number of levels, representing concentration intervals. Furthermore they considered the same step size h and the same maximum level N for all the species.

In Bio-PEPA we adapt this approach to general kinetic laws, stoichiometry greater than one and different numbers of levels for the species. The granularity of the system is defined in terms of the step size h of the concentration intervals instead of the number of levels. We define the same step size h for all the species⁴. This is motivated by the fact that, following the *law of conservation of mass*, there must be a “balance” between the concentrations consumed (reactants) and the ones created (products). In the case the stoichiometry is greater than one we need to consider concentration quantities proportional to stoichiometric coefficients. Given a species i , we can assume that it has

⁴ There can be some exceptions to this assumption: 1) since modifiers remain constant during reaction, we may define a different step size for each species which is only a modifier; 2) any species which is involved on in creation/degradation reactions may have a different step size.

a maximum finite concentration M_i . The number of levels for the species i is given by $N_i + 1$ where $N_i = \lceil \frac{M_i}{h} \rceil$ (the integer value greater than or equal to $\frac{M_i}{h}$). Each species can assume the discrete concentration levels from 0 (concentration null) to N_i (maximum concentration).

If l_i is the concentration level for the species i , the concentration is taken to be $x_i = l_i \times h$.

When a finite state space CTMC is to be generated, for numerical analysis or stochastic model checking, we must assume that there is a maximum concentration for each species. However, we can have a species without a limiting value: we use a maximum level to capture all values greater than a given (high) value.

3.2 The syntax

The syntax of Bio-PEPA is similar to that of PEPA but with some important differences. As in PEPA a model is made up of a number of sequential components; here there is one sequential component for each species. As we will see, the syntax of Bio-PEPA is designed in order to collect the biological information that we need. For example, instead of a single prefix combinator there are a number of different operators which capture the role that the species plays with respect to this reaction.

$$S ::= (\alpha, \kappa) \text{ op } S \mid S + S \mid C \quad P ::= P \boxtimes_{\ell} P \mid S(l)$$

where $\text{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$.

The component S is called *sequential component* (or *species component*) and represents the species. The component P , called a *model component*, describes the system and the interactions among components. We suppose a countable set of sequential components C and a countable set of action types \mathcal{A} . The parameter $l \in \mathbb{N}$ represents the discrete level of concentration. The prefix term, $(\alpha, \kappa) \text{ op } S$, contains information about the role of the species in the reaction associated with the action type α :

- (α, κ) is the *activity* or *reaction*, where $\alpha \in \mathcal{A}$ is the *action type* and κ is the *stoichiometric coefficient* of the species in that reaction; information about the rate of the reaction is defined elsewhere (in contrast to PEPA);
- the *prefix combinator* “op” represents the role of the element in the reaction. Specifically, \downarrow indicates a *reactant*, \uparrow a *product*, \oplus an *activator*, \ominus an *inhibitor* and \odot a generic *modifier*.

The choice operator, cooperation and definition of constant are unchanged. In contrast to PEPA the hiding operator is omitted.

In order to fully describe a biochemical network in Bio-PEPA we need to define structures that collect information about the compartments, the maximum concentrations, number of levels for all the species, the constant parameters and the functional rates which specify the rates of reactions. In the following the function *name* returns the names of the elements of a given Bio-PEPA component.

First of all we define the set of compartments.

Definition 1. Each compartment is described by “ $V: v$ unit”, where V is the compartment name, “ v ” is a positive real number expressing the compartment size and the (optional) “unit” denotes the unit associated with the compartment size. The set of compartments is denoted \mathcal{V} .

In Bio-PEPA compartments are static and they cannot change their structure/size. The set of compartments must contain at least one element. When we have no information about compartments we add a default compartment whose size is 1 and whose unit depends on the model.

Definition 2. For each species we define the element $C : H, N, M_0, M, V$, where:

- C is the species component name,
- $H \in \mathbb{N}$ is the step size,
- $N \in \mathbb{N}$ is the maximum level,
- $M_0 \in \mathbb{R}^+ \cup \{-\}$ is the initial concentration,
- $M \in \mathbb{R}^+ \cup \{-\}$ is the maximum concentration,
- $V \in \text{name}(\mathcal{V}) \cup \{-\}$ is the name of the enclosing compartment.

The set of all the elements $C : H, N, M_0, M, V$ is denoted \mathcal{N} .

In the definition the symbol “-” denotes the empty string, indicating that the last three components are optional. The initial concentration may added when we want to compare our model results with the results in the literature. The maximum concentration is used in the definition of the number of levels, but generally it can be derived from the step size and the maximum number of levels. Finally, if there is only one compartment for all the species in the model we can omit it in the definition of \mathcal{N} .

In order to specify the dynamics of the system we associate a functional rate f_{α_j} with each action α_j . This function represents the kinetic law of the associated reaction. For the definition of functional rates we consider mathematical expressions with simple operations and operators involving constant parameters and components. All the kinetic laws proposed in the book by Segel [65] can be defined in this way. In addition, for convenience, we include some predefined functions to express the most commonly used kinetic laws.

The predefined kinetic laws considered are mass-action (fMA), Michaelis-Menten (fMM) and Hill kinetics (fH). They depend only on some parameters; the components/species are derived from the context⁵. The functional rates are defined externally to the components and are evaluated when the system is derived. They are used to derive the transition rates of the system. In the functional rates some parameter constants can be used. These must be defined in the model by means of the set of parameter definitions \mathcal{K} .

⁵ In the case of mass-action, the function $fMA(r)$ is $r \times \prod_{i=1}^{n_j} (C_i)^{\kappa_i}$, where C_i $i = 1, \dots, n_j$ are the n_j distinct reactants involved in the reaction and κ_i is the associated stoichiometric coefficients. The information about the reactants are derived from the Bio-PEPA specifications of the system. In the case of Michaelis-Menten, the function $fMM(v_M, K_M)$ is $v_M \times E \times S / (K_M + S)$, where E is the concentration of the enzyme and S the concentration of the substrate. Also in this case E and S are derived from the Bio-PEPA specifications. In the case of Hill kinetics, the function $fH(v, K, n)$ is $v \times C^n / (K + C^n)$, where C is the concentration of the element involved in the reaction.

Definition 3. Each parameter is defined by “ $k_{name} = \text{value unit}$ ”, where “ k_{name} ” $\notin C$ is the parameter name, “value” denotes a positive real number and the (optional) “unit” denotes the unit associated with the parameter. The set of the parameters is denoted \mathcal{K} .

Finally, we have the following definition for the set of sequential components:

Definition 4. The set *Comp* of sequential components is defined as

$$\text{Comp} ::= \{C \stackrel{\text{def}}{=} S, \text{ where } S \text{ is a sequential component } \}$$

We can define a Bio-PEPA system in the following way:

Definition 5. A Bio-PEPA system \mathcal{P} is a 6-uple $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, \text{Comp}, P \rangle$, where:

- \mathcal{V} is the set of compartments;
- \mathcal{N} is the set of quantities describing each species;
- \mathcal{K} is the set of parameter definitions;
- \mathcal{F}_R is the set of functional rate definitions;
- *Comp* is the set of definitions of sequential components;
- P is the model component describing the system.

In a *well-defined* Bio-PEPA system each element has to satisfy some (reasonable) conditions. Details can be found in [23]. In the remainder of the chapter we consider only well-defined Bio-PEPA systems. The set of such systems is denoted $\tilde{\mathcal{P}}$.

3.3 The semantics

The semantics of Bio-PEPA is defined in terms of an operational semantics. We define two relations over the processes. The former, called the *capability relation*, supports the derivation of quantitative information and it is auxiliary to the latter which is called the *stochastic relation*. The stochastic relation gives us the rates associated with each action. The rates are obtained by evaluating the functional rate corresponding to the action, divided by the step size of the species involved, using quantitative information derived from the capability relation.

The capability relation is $\rightarrow_c \subseteq C \times \Theta \times C$, where the label $\theta \in \Theta$ contains the quantitative information needed to evaluate the functional rate. We define the labels θ as:

$$\theta := (\alpha, w)$$

where w is defined as $w ::= [S : op(l, \kappa)] \mid w :: w$, with $S \in C$, l the level and κ the stoichiometric coefficient of the components. The order of the components is not important. The rules governing the behaviour of components are presented in the structured operational style [57] in Table 1. The rules should be read as follows: if the transition above the line can be inferred then the transition below the line can be deduced. The relation \rightarrow_c is defined as the minimum relation satisfying the rules reported in Table 1.

The first three axioms describe the behaviour of the three different prefix terms. In the case of a reactant, the level decreases; in the case of a product, the level increases; whereas in the case of modifiers, the level remains the same. For reactants and products,

prefixReac	$((\alpha, \kappa) \downarrow S)(l) \xrightarrow{(\alpha, [S: \downarrow(l, \kappa)])}_c S(l - \kappa) \quad \kappa \leq l \leq N$	
prefixProd	$((\alpha, \kappa) \uparrow S)(l) \xrightarrow{(\alpha, [S: \uparrow(l, \kappa)])}_c S(l + \kappa) \quad 0 \leq l \leq (N - \kappa)$	
prefixMod	$((\alpha, \kappa) op S)(l) \xrightarrow{(\alpha, [S: op(l, \kappa)])}_c S(l) \quad \text{with } op = \odot, \oplus, \ominus \text{ and } 0 \leq l \leq N$	
choice1	$\frac{S_1(l) \xrightarrow{(\alpha, w)}_c S'_1(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha, w)}_c S'_1(l')}$	choice2 $\frac{S_2(l) \xrightarrow{(\alpha, w)}_c S'_2(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha, w)}_c S'_2(l')}$
constant	$\frac{S(l) \xrightarrow{(\alpha, S': [op(l, \kappa)])}_c S'(l')}{C(l) \xrightarrow{(\alpha, C: [op(l, \kappa)])}_c S'(l')} \quad \text{with } C \stackrel{def}{=} S$	
coop1	$\frac{P_1 \xrightarrow{(\alpha, w)}_c P'_1}{P_1 \boxtimes_{\mathcal{L}} P_2 \xrightarrow{(\alpha, w)}_c P'_1 \boxtimes_{\mathcal{L}} P_2} \quad \text{with } \alpha \notin \mathcal{L}$	
coop2	$\frac{P_2 \xrightarrow{(\alpha, w)}_c P'_2}{P_1 \boxtimes_{\mathcal{L}} P_2 \xrightarrow{(\alpha, w)}_c P_1 \boxtimes_{\mathcal{L}} P'_2} \quad \text{with } \alpha \notin \mathcal{L}$	
coop3	$\frac{P_1 \xrightarrow{(\alpha, w_1)}_c P'_1 \quad P_2 \xrightarrow{(\alpha, w_2)}_c P'_2}{P_1 \boxtimes_{\mathcal{L}} P_2 \xrightarrow{(\alpha, w_1 \oplus w_2)}_c P'_1 \boxtimes_{\mathcal{L}} P'_2} \quad \text{with } \alpha \in \mathcal{L}$	

Table 1. Axioms and rules for Bio-PEPA.

the number of levels increment or decrement depends on the stoichiometric coefficient κ . This expresses the degree to which a species (reactant or product) participates in a reaction. Therefore some side conditions concerning the present concentration level must be added to the rules. Specifically, for the reactants the level has to be greater than or equal to κ , whereas for the products the level has to be less than or equal to $(N - \kappa)$, where N is the maximum level. The modifiers can have any possible value between 0 and N . In all three cases the label θ records the level and the stoichiometry of the associated component. The rules **choice1** and **choice2** have the usual meaning, but note that choices only occur within a species component so both alternatives are associated with the same level. The rule **constant** is used to define the behaviour of the constant term, defined by one or more prefix terms in summation. The label contains the information about the level and the stoichiometric coefficient related to the action α . The last three rules report the case of cooperation. The rules **coop1** and **coop2** concern the case when the action enabled does not belong to the cooperation set. In this case the label in the conclusion contains only the information about the component that fires the

action. The rule `coop3` describes the case in which the two components synchronize and the label reports the information from both the components. The concatenation operator of lists `@` is used for this purpose.

In order to associate the rates with the transitions we introduce the stochastic relation $\rightarrow_s \subseteq \tilde{\mathcal{P}} \times \Gamma \times \tilde{\mathcal{P}}$, where the label $\gamma \in \Gamma$ is defined as $\gamma := (\alpha, r_\alpha)$, with $r_\alpha \in \mathbb{R}^+$. In this definition r_α represents the parameter of a negative exponential distribution. The dynamic behaviour of processes is determined by a *race condition*: all activities enabled attempt to proceed but only the fastest succeeds.

The relation \rightarrow_s is defined as the minimal relation satisfying the rule

$$\text{Final} \quad \frac{P \xrightarrow{(\alpha_j, w)}_c P'}{\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, \text{Comp}, P \rangle \xrightarrow{(\alpha_j, r_\alpha[w, \mathcal{N}, \mathcal{K}])}_s \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, \text{Comp}, P' \rangle}$$

The second component in the label of the conclusion represents the rate associated with the transition. The rate is calculated from the functional rate f_α in the following way:

$$r_\alpha[w, \mathcal{N}, \mathcal{K}] = \frac{f_\alpha[w, \mathcal{N}, \mathcal{K}]}{h}$$

where h is the step size and $f_\alpha[w, \mathcal{N}, \mathcal{K}]$ denotes the function f_α is evaluated over w , \mathcal{N} and \mathcal{K} . Specifically, for each component C_i we derive the concentration as $l_i \times h$. Then we replace each free occurrence of C_i by $(l_i \times h)^{\kappa_{ij}}$, where κ_{ij} is the stoichiometric coefficient of the species i with respect to the reaction R_j . The derivation of rates is discussed in some more detail later.

A *Stochastic Labelled Transition System* can be defined for a Bio-PEPA system.

Definition 6. *The Stochastic Labelled Transition System (SLTS) for a Bio-PEPA system is $(\tilde{\mathcal{P}}, \Gamma, \rightarrow_s)$, where \rightarrow_s is the minimal relation satisfying the rule `Final`.*

The states of *SLTS* are defined in terms of the concentration levels of the species components and the transitions from one state to another represent reactions that cause changes in the concentration levels of some components.

Note that using the relation \rightarrow_c it is possible to define another labelled transition system (*LTS*) as $(\mathcal{C}, \Theta, \rightarrow_c)$ which differs only in the transition labels.

Derivation of rates In the *SLTS* the states represent *levels of concentration* and the transitions cause a change in these levels for one or more species. As we have seen the number of levels depends on the stoichiometric coefficients of the species involved.

Consider a reaction j described by a *kinetic law* f_j and with all stoichiometric coefficients equal to one. Following [16], we can define the transition rate as $(\Delta t)^{-1}$, where Δt is the time to have a variation in the concentration of one step for both the reactants and the products of the reaction. Let y be a variable describing one product of the reaction. We can consider the rate equation for that species with respect to the given reaction. This is $dy/dt = f_j(\bar{x}(t))$, where \bar{x} is the set (or a subset) of the reactants/modifiers of the reaction. We can apply the *Taylor expansion* up to the second term and we obtain

$$y_{n+1} \approx y_n + f(\bar{x}_n) \times (t_{n+1} - t_n)$$

Now we can fix $y_{n+1} - y_n = h$ and then derive the time interval $(t_{n+1} - t_n) = \Delta t$ as $\Delta t \approx h/f(\bar{x}_n)$. From this we obtain the transition rate as $f(\bar{x}_n)/h$.

When the reaction has stoichiometric coefficients different from one, we can consider an approach similar to the one above. However, in this case, we assume mass action kinetics as this is generally the case for stoichiometric coefficient greater than one. Let y be a product of the reaction. The approximation gives:

$$y_{n+1} \approx y_n + r \times \kappa \times \prod_{i=1}^{n_r} x_{i,n}^{\kappa_i} \times (t_{n+1} - t_n)$$

where r is the reaction constant rate, κ is stoichiometric coefficient of the product y , x_i $i = 1, \dots, n_r$ are the reactants of the reaction, κ_i $i = 1, \dots, n_r$ are the associated stoichiometric coefficients, n_r is the number of distinct reactants.

Now we can fix $y_{n+1} - y_n = \kappa \times h$ and then derive the respective $(t_{n+1} - t_n) = \Delta t$ as $\Delta t \approx h/(r \times \prod_{i=1}^{n_r} x_{i,n}^{\kappa_i})$. From this expression we can derive the rate as usual.

Note that this approach is based on an *approximation*, the accuracy of which will depend on the time/concentration steps used.

From biochemical networks to Bio-PEPA We define a translation, tr_BM_BP , from a biochemical network \mathcal{M} to a Bio-PEPA system $\mathcal{P} = \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P \rangle$, based on the following abstraction:

1. Each compartment is defined in the set \mathcal{V} in terms of a name and an associated volume. Recall that currently in Bio-PEPA, compartments are not involved actively in the reactions and therefore are not represented by processes.
2. Each species i in the network is described by a constant component $C_i \in Comp$. The constant component C_i is defined by the “sum” of *elementary components* describing the interaction capabilities of the species. We suppose that there is at most one term in each species component with an action of type α . A single definition can express the behaviour of the species at any level.
3. Each reaction j is associated with an action type α_j and its dynamics is described by a specific function $f_{\alpha_j} \in \mathcal{F}_R$. The constant parameters used in the function can be defined in \mathcal{K} .
4. The model P is defined as the cooperation of the different components C_i .

3.4 Some examples

Now we present some simple examples in order to show how Bio-PEPA can be used to capture some biological situations.

Example 1: Mass-action kinetics Consider the reaction $2X + Y \xrightarrow{f_M} 3Z$, described by the mass-action kinetic law $f_M = r \times X^2 \times Y$. The three species can be specified by the syntax:

$$X \stackrel{def}{=} (\alpha, 2)\downarrow X \quad Y \stackrel{def}{=} (\alpha, 1)\downarrow Y \quad Z \stackrel{def}{=} (\alpha, 3)\uparrow Z$$

The system is described by $(X(l_{X0}) \underset{[\alpha]}{\bowtie} Y(l_{Y0})) \underset{[\alpha]}{\bowtie} Z(l_{Z0})$, where l_{X0} , l_{Y0} and l_{Z0} denote the initial concentration level of the three components. The functional rate is $f_\alpha = fMA(r)$. The rate associated with a transition is given by:

$$r_\alpha = \frac{r \times (l_X \times h)^2 \times (l_Y \times h)}{h}$$

where l_X , l_Y are the concentration levels for the species X and Y in a given state and h is the step size of all the species. The reaction can happen only if we have at least 3 levels (0, 1, 2) for X , 2 levels for Y (0, 1) and 4 levels (0, 1, 2, 3) for Z .

Example 2: Michaelis-Menten kinetics One of the most commonly used kinetic laws is Michaelis-Menten. It describes a basic enzymatic reaction from the substrate S to the product P and is written as $S \xrightarrow{E;f_E} P$, where E is the enzyme involved in the reaction. This reaction is an approximation of a sequence of two reactions, under the quasi-steady state assumption (QSSA). The whole sequence of reactions is described by the kinetic law $f_E = \frac{v_M \times E \times S}{(K_M + S)}$. For more details about the derivation of this kinetic law and the meaning of parameters see [65].

The three species can be specified in Bio-PEPA by the following components:

$$S \stackrel{\text{def}}{=} (\alpha, 1) \downarrow S \quad P \stackrel{\text{def}}{=} (\alpha, 1) \uparrow P \quad E \stackrel{\text{def}}{=} (\alpha, 1) \oplus E$$

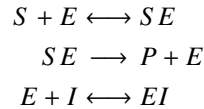
The system is described by $(S(l_{S0}) \underset{[\alpha]}{\bowtie} E(l_{E0})) \underset{[\alpha]}{\bowtie} P(l_{P0})$ and the functional rate is $f_\alpha = fMM(v_M, K_M)$.

The transition rate is given by:

$$r_\alpha = \frac{v_M \times (l_S \times h) \times (l_E \times h)}{(K_M + l_S \times h)} \times \frac{1}{h}$$

where l_S , l_E are the concentration levels for the species S and E in a given state and h is the step size of all the species. The reaction can happen only if we have at least 2 levels (0, 1) for all the species involved.

Example 3: competitive inhibition Competitive inhibition is a form of enzyme inhibition where binding of the inhibitor to the enzyme prevents binding of the substrate and vice versa. In classical competitive inhibition, the inhibitor binds to the same active site as the normal enzyme substrate, without undergoing a reaction. The substrate molecule cannot enter the active site while the inhibitor is there, and the inhibitor cannot enter the site when the substrate is there. This reaction is described as:



where S is the substrate, E the enzyme, I the inhibitor and P the product. Under QSSA the intermediate species SE and EI are constant and we can approximate the reactions above by a unique reaction $S \xrightarrow{E,I:f_I} P$, with rate $f_I = \frac{v_c \times S \times E}{S + K_M(1 + \frac{I}{K_I})}$, where v_c is the turnover number (catalytic constant), K_M is the Michaelis-constant and K_I is the inhibition constant.

The specification in Bio-PEPA is:

$$S \stackrel{\text{def}}{=} (\alpha, 1) \downarrow S \quad P \stackrel{\text{def}}{=} (\alpha, 1) \uparrow P \quad E \stackrel{\text{def}}{=} (\alpha, 1) \oplus E \quad I \stackrel{\text{def}}{=} (\alpha, 1) \ominus I$$

The system is described by $((S(l_{S0}) \bowtie_{(\alpha)} E(l_{E0})) \bowtie_{(\alpha)} I(l_{I0})) \bowtie_{(\alpha)} P(l_{P0})$ with functional rate

$$f_\alpha = f_{CI}((v_c, K_M, K_I), S, E, I) = \frac{v_c \times S \times E}{S + K_M(1 + \frac{I}{K_I})}$$

The transition rate is given by:

$$r_\alpha = \frac{v_c \times (l_S \times h) \times (l_E \times h)}{(l_S \times h + K_M(1 + \frac{l_I \times h}{K_I}))} \times \frac{1}{h}$$

where l_S, l_E, l_I are the concentration levels for the species S, E, I in a given state and h is the step size of all the species. The reaction can happen only if we have at least 2 levels (0, 1) for all the species involved.

Example 4: degradation and synthesis of a species Two particular reactions are those which describe the degradation and the creation of a species. In order to model these reactions we need to add two auxiliary species components to represent respectively the *residue* (Res) of the reaction and the *creation factor* (CF), i.e. genes or DNA.

Let us consider the degradation reaction $A \rightarrow \emptyset$. We describe this reaction in Bio-PEPA by introducing the component Res as the residue/product of the reaction. The two species A and Res are defined as:

$$A \stackrel{\text{def}}{=} (\alpha, 1) \downarrow A \quad Res \stackrel{\text{def}}{=} (\alpha, 1) \odot Res$$

The component Res is described by one or more sub-terms each of which describes a different degradation reaction.

In contrast the synthesis of a species $\emptyset \rightarrow A$ is described by a new component CF . The two species A and CF are described by:

$$A \stackrel{\text{def}}{=} (\alpha, 1) \uparrow A \quad CF \stackrel{\text{def}}{=} (\alpha, 1) \odot CF$$

In the definitions of the components Res and CF we use the symbol \odot to indicate that they do not change with the reaction.

3.5 Equivalences

It is sometimes useful to consider *equivalences* between models in order to determine whether the systems represented are in some sense the “same”. In this section we discuss some notions of equivalence for Bio-PEPA. We consider two styles of equivalence which are commonly considered for process algebras: *isomorphism*, a structural equivalence, and *bisimulation*, a behavioural equivalence. Some characteristics of the language impact on the definitions of equivalence and we start by highlighting those. Firstly, there is no hiding operator or τ actions. Therefore, in Bio-PEPA we do not have weaker forms of equivalence based on abstracting τ actions. Secondly, in well-defined systems we have at most one action of a given type in each sequential term and each component describes the behaviour of a single species. So we cannot have processes of the form “ $S + S$ ” and terms such as “ $A = a.C$ ” (where A and C differ). Thirdly, if we have two transitions between the processes P and P' , they involve different action types and they represent similar reactions that differ only in the kind/number of modifiers. Finally, we have defined two relations within the semantics. In one case the labels contain the information about the action type and about the elements involved. This is used as an auxiliary relation for the derivation of the second one, in which the labels contain the information about the action type and the rate (similarly to PEPA activity). Thus we have a choice of which relation on which to base each notion of equivalence.

Recall that in Bio-PEPA we make a distinction between systems and model components. However note that the only element that is modified by the transitions of a Bio-PEPA system is the model component. All the other components remain unchanged. Thus we define equivalences for the Bio-PEPA systems in terms of equivalences for the model components. Specifically, we say that two Bio-PEPA systems \mathcal{P}_1 and \mathcal{P}_2 are equivalent if their respective model components are equivalent.

Auxiliary definitions Before we proceed it will be useful to make some auxiliary definitions. Firstly we consider the derivative of a component, the derivative set and the derivative graph. We refer to the relation \rightarrow_s , the case of \rightarrow_c is analogous, the only differences are in the label and in the fact that the former relation refers to Bio-PEPA systems and the latter refers to model components.

Definition 7. If $\mathcal{P} \xrightarrow{(\alpha,r)}_s \mathcal{P}'$ then \mathcal{P}' is a one-step \rightarrow_s system derivative of \mathcal{P} .

If $\mathcal{P} \xrightarrow{(\alpha_1,r_1)}_s \mathcal{P}_1 \xrightarrow{(\alpha_2,r_2)}_s \dots \xrightarrow{(\alpha_n,r_n)}_s \mathcal{P}'$ then \mathcal{P}' is a system derivative of \mathcal{P} .

We can indicate the sequence $\xrightarrow{\gamma_1}_s \xrightarrow{\gamma_2}_s \dots \xrightarrow{\gamma_n}_s$ with $\xrightarrow{\mu}_s$, where μ denotes the sequence $\gamma_1\gamma_2, \dots, \gamma_n$ (possibly empty).

Definition 8. A system α -derivative of \mathcal{P} is a system \mathcal{P}' such that $\mathcal{P} \xrightarrow{(\alpha,r)}_s \mathcal{P}'$. For each $\alpha \in \mathcal{A}$ we have at most one system α -derivative of a system \mathcal{P} .

Definition 9. The system derivative set $ds(\mathcal{P})$ is the smallest set such that:

- $\mathcal{P} \in ds(\mathcal{P})$;
- if $\mathcal{P}' \in ds(\mathcal{P})$ and there exists $\alpha \in \mathcal{A}(\mathcal{P}')$ such that $\mathcal{P}' \xrightarrow{(\alpha,r)}_s \mathcal{P}''$ then $\mathcal{P}'' \in ds(\mathcal{P})$, where $\mathcal{A}(\mathcal{P}')$ is the set of action types currently enabled in the system derivative \mathcal{P}' .

Definition 10. The system derivative graph $\mathcal{D}(\mathcal{P})$ is the labelled directed multi-graph whose set of nodes is $ds(\mathcal{P})$ and whose multi-set of arcs are elements in $ds(\mathcal{P}) \times ds(\mathcal{P}) \times \Gamma$.

Note that in well-defined Bio-PEPA components the multiplicity of $\langle \mathcal{P}_i, \mathcal{P}_j, \gamma \rangle$ is always one.

The definitions above refer to Bio-PEPA systems. The only components of the system $\mathcal{P} = \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, \text{Comp}, P \rangle$ that evolves is the model component P . The other components collect information about the compartments, the species, the rates and report the definition of the species components. They remain unchanged in the evolution of the system. In some cases it can be useful (and simpler) to focus on the model component instead of considering the whole system and use the other components for the derivation of the rates. We define a function $\pi_P(\mathcal{P}) = P$, that, given a Bio-PEPA system returns the model component. Then we define a (*component*) *derivative* of P by considering the model component P' of the system derivative of \mathcal{P} . Similarly, we define a (*component*) α -*derivative* of P , (*component*) *derivative set* $ds(P)$ and the (*component*) *derivative graph* $\mathcal{D}(P)$ starting from the definitions for the associated system \mathcal{P} .

In the derivation of the CTMC (see Section 3.6) we need to identify the actions describing the interactions from one state to another.

Definition 11. Let \mathcal{P} be a Bio-PEPA system and let $P = \pi_P(\mathcal{P})$. Let P_u, P_v be two derivatives of a model component P with P_v a one-step derivative of P_u . The set of action types associated with the transitions from the process P_u to the process P_v is denoted $\mathcal{A}(P_u | P_v)$.

The next definition concerns the *complete action type set* of a system \mathcal{P} and a component P .

Definition 12. The complete action type set of a system \mathcal{P} is defined as:

$$\bar{\mathcal{A}} = \cup_{\mathcal{P}_i \in ds(\mathcal{P})} \mathcal{A}(\mathcal{P}_i)$$

The complete action type set of a component P is defined similarly.

Other useful definitions are the ones concerning the exit rate and transition rates. In the following we report the definition for the model components, but a similar definition can be used for Bio-PEPA systems.

Definition 13. Let us consider a Bio-PEPA system $\mathcal{P} = \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, \text{Comp}, P \rangle$ and let $P_1, P_2 \in ds(P)$. The exit rate of a process P_1 is defined as:

$$\text{rate}(P_1) = \sum_{\substack{(\alpha, r_\alpha[w, \mathcal{N}, \mathcal{K}]) \\ \{\alpha | \exists \mathcal{P}_2, \mathcal{P}_1 \xrightarrow{(\alpha, r_\alpha[w, \mathcal{N}, \mathcal{K}])} \mathcal{P}_2, P_1 = \pi_P(\mathcal{P}_1)\}}} r_\alpha[w, \mathcal{N}, \mathcal{K}]$$

Similarly, the transition rate is defined as:

$$\text{rate}(P_1 | P_2) = \sum_{\substack{(\alpha, r_\alpha[w, \mathcal{N}, \mathcal{K}]) \\ \{\alpha | \mathcal{P}_1 \xrightarrow{(\alpha, r_\alpha[w, \mathcal{N}, \mathcal{K}])} \mathcal{P}_2, P_1 = \pi_P(\mathcal{P}_1), P_2 = \pi_P(\mathcal{P}_2)\}}} r_\alpha[w, \mathcal{N}, \mathcal{K}]$$

For the label γ in the stochastic relation, the function $action(\gamma) = \alpha$ extracts the first component of the pair (i.e. the action type) and the function $rate(\gamma) = r \in \mathbb{R}$ returns the second component (i.e. the rate).

In the following we use the same symbol to denote equivalences for both the system and the corresponding model component. In this section we present definitions of isomorphism and strong bisimulation which are similar to the relations defined for PEPA in [42]. Furthermore we show some relationships between the defined equivalences.

Isomorphism *Isomorphism* is a strong notion of equivalence based on the derivation graph of the components (systems). Broadly speaking, two components (systems) are isomorphic if they generate derivation graphs with the same structure and capable of carrying out exactly the same activities.

We have the following definition of isomorphism based on the capability relation:

Definition 14. *Let $\mathcal{P}_1, \mathcal{P}_2$ be two Bio-PEPA systems whose model components are P and Q , respectively. A function $\mathcal{F} : ds(P) \rightarrow ds(Q)$ is a component isomorphism between P and Q , with respect to \rightarrow_c , if \mathcal{F} is an injective function and for any component $P' \in ds(P)$, $\mathcal{A}(P') = \mathcal{A}(\mathcal{F}(P'))$, with $r_\alpha[w, \mathcal{N}, \mathcal{K}] = r'_\alpha[\mathcal{F}(w), \mathcal{N}', \mathcal{K}']$ for each $\alpha \in \mathcal{A}(P)$, where $\mathcal{F}(w)$ is defined component-wise over the list w , and for all $\alpha \in \mathcal{A}$ the set of α -derivatives of $\mathcal{F}(P')$ is the same as the set of \mathcal{F} -images of the α -derivatives of P' , with respect to \rightarrow_c .*

This is a very strong relation because the labels associated with the capability relation contain a lot of information, all of which must be matched. Formally, we can define isomorphic components in the following way:

Definition 15. *Let $\mathcal{P}_1, \mathcal{P}_2$ be two Bio-PEPA systems whose model components are P and Q . P and Q are isomorphic with respect to \rightarrow_c (denoted $P =_c Q$), if there exists a component isomorphism \mathcal{F} between them such that $\mathcal{D}(\mathcal{F}(P)) = \mathcal{D}(Q)$, where \mathcal{D} denotes the derivative graph.*

We can now define when two Bio-PEPA systems are isomorphic.

Definition 16. *Let $\mathcal{P}_1, \mathcal{P}_2$ be two Bio-PEPA systems whose model components are P and Q . \mathcal{P}_1 and \mathcal{P}_2 are isomorphic with respect to \rightarrow_c (denoted $\mathcal{P}_1 =_c \mathcal{P}_2$), if $P =_c Q$.*

A similar structural relation based on the stochastic relation can also be defined and used to characterise another form of isomorphism between systems (components) $=_s$ (see [23] for details). Both isomorphisms, $=_c$ and $=_s$ are equivalence relations, and congruences with respect to the combinators of Bio-PEPA. In both cases they retain enough information about the structure and behaviour of the isomorphic components to ensure that they give rise to identical underlying Markov processes. However, $=_c$ is more strict than $=_s$, i.e. there will be pairs of systems (components) which satisfy $=_s$ but do not satisfy $=_c$.

Equational laws Once an equivalence relation has been defined it can be used to establish equational laws which may be used to manipulate models and recognise equivalent terms. In the following the symbol “=” denotes either $=_c$ or $=_s$. The proof of the laws follow from the definition of isomorphism and the semantic rules.

Choice

1. $(P + Q) \boxtimes_{\mathcal{L}} S = (Q + P) \boxtimes_{\mathcal{L}} S$
2. $(P + (Q + R)) \boxtimes_{\mathcal{L}} S = ((P + Q) + R) \boxtimes_{\mathcal{L}} S$

Cooperation

1. $P \boxtimes_{\mathcal{L}} Q = Q \boxtimes_{\mathcal{L}} P$
2. $P \boxtimes_{\mathcal{L}} (Q \boxtimes_{\mathcal{L}} R) = (P \boxtimes_{\mathcal{L}} Q) \boxtimes_{\mathcal{L}} R$
3. $P \boxtimes_{\mathcal{K}} Q = P \boxtimes_{\mathcal{L}} Q$ if $\mathcal{K} \cap (\bar{\mathcal{A}}(P) \cup \bar{\mathcal{A}}(Q)) = \mathcal{L}$
4. $(P \boxtimes_{\mathcal{L}} Q) \boxtimes_{\mathcal{K}} R = \begin{cases} P \boxtimes_{\mathcal{L}} (Q \boxtimes_{\mathcal{K}} R) & \text{if } \bar{\mathcal{A}}(R) \cap (\mathcal{L} \setminus \mathcal{K}) = \emptyset \wedge \bar{\mathcal{A}}(P) \cap (\mathcal{K} \setminus \mathcal{L}) = \emptyset \\ Q \boxtimes_{\mathcal{L}} (P \boxtimes_{\mathcal{K}} R) & \text{if } \bar{\mathcal{A}}(R) \cap (\mathcal{L} \setminus \mathcal{K}) = \emptyset \wedge \bar{\mathcal{A}}(Q) \cap (\mathcal{K} \setminus \mathcal{L}) = \emptyset \end{cases}$

Constant If $A \stackrel{\text{def}}{=} P$ then $A = P$

Bio-PEPA systems

Let \mathcal{P}_1 and \mathcal{P}_2 be two Bio-PEPA systems, with $P = \pi_P(\mathcal{P}_1)$ and $Q = \pi_P(\mathcal{P}_2)$.
If $P = Q$ then $\mathcal{P}_1 = \mathcal{P}_2$.

Strong bisimulation The definition of bisimulation is based on the *labelled transition system*. Strong bisimulation captures the idea that bisimilar components (systems) are able to perform the same actions with same rates resulting in derivatives that are themselves bisimilar. This makes the components (systems) indistinguishable to an external observer. As with isomorphism we can develop two definitions based on the two semantic relations. This time for illustration we present the definitions based on the stochastic relation. The strong capability bisimulation, \sim_c , is defined similarly (see [23] for details).

Definition 17. A binary relation $\mathcal{R} \subseteq \tilde{\mathcal{P}} \times \tilde{\mathcal{P}}$ is a strong stochastic bisimulation, if $(\mathcal{P}_1, \mathcal{P}_2) \in \mathcal{R}$ implies for all $\alpha \in \mathcal{A}$:

- if $\mathcal{P}_1 \xrightarrow{\alpha}_s \mathcal{P}'_1$ then there exists \mathcal{P}'_2 such that $\mathcal{P}_2 \xrightarrow{\alpha}_s \mathcal{P}'_2$ with $(\mathcal{P}'_1, \mathcal{P}'_2) \in \mathcal{R}$.
- if $\mathcal{P}_2 \xrightarrow{\alpha}_s \mathcal{P}'_2$ then there exists \mathcal{P}'_1 such that $\mathcal{P}_1 \xrightarrow{\alpha}_s \mathcal{P}'_1$ with $(\mathcal{P}'_1, \mathcal{P}'_2) \in \mathcal{R}$.

Definition 18. Let $\mathcal{P}_1, \mathcal{P}_2$ be two Bio-PEPA systems whose model components are P and Q , respectively. P and Q are strong stochastic bisimilar, written $P \sim_s Q$, if $(\mathcal{P}_1, \mathcal{P}_2) \in \mathcal{R}$ for some strong stochastic bisimulation \mathcal{R} .

Definition 19. Let $\mathcal{P}_1, \mathcal{P}_2$ be two Bio-PEPA systems whose model components are P and Q , respectively. $\mathcal{P}_1, \mathcal{P}_2$ are strong stochastic bisimilar, written $\mathcal{P}_1 \sim_s \mathcal{P}_2$, if $P \sim_s Q$.

Both \sim_c and \sim_s are equivalence relations and congruences with respect to the combinators of Bio-PEPA. Moreover it is straightforward to see that isomorphism implies strong bisimulation in both cases.

Example Consider the following systems representing two biological systems. The former system \mathcal{P}_1 represents a system described by an enzymatic reaction with kinetic law $\frac{v_1 \times E \times S}{K_1 + S}$, where S is the substrate and E the enzyme. We have that the set \mathcal{N} is defined as $\{S : h, N_S; P : h, N_P; E : 1, 1\}$ for some step size h and maximum levels N_S and N_P . The component and the model components are defined as:

$$S \stackrel{\text{def}}{=} (\alpha, 1)\downarrow S \quad E \stackrel{\text{def}}{=} (\alpha, 1) \oplus E \quad P \stackrel{\text{def}}{=} (\alpha, 1)\uparrow P$$

The model component P_1 is $(S(l_{S0}) \bowtie_{[\alpha]} E(1)) \bowtie_{[\alpha]} P(l_{P0})$. The functional rate is $f_\alpha = fMM(v_1, K_1)$.

The second system \mathcal{P}_2 describes an enzymatic reaction where the enzyme is left implicit (it is constant). The rate is given by $\frac{v_1 \times S'}{K_1 + S'}$, where S' is the substrate.

We have that the set \mathcal{N} is defined as $\{S' : h, N_{S'}; P' : h, N_{P'}\}$.

The components are defined as $S' \stackrel{\text{def}}{=} (\alpha, 1)\downarrow S'$ and $P' \stackrel{\text{def}}{=} (\alpha, 1)\uparrow P'$ and the model component P_2 is $S'(l_{S0}) \bowtie_{[\alpha]} P'(l_{P0})$. In this case $f_\alpha = fMM'((v_1, K_1), S') = \frac{v_1 \times S'}{K_1 + S'}$ and the component S' and P' have the same number of levels/maximum concentration of S and P .

We have that $P_1 \sim_s P_2$, but $P_1 \not\sim_c P_2$, because the number of enzymes is different. The same relations are valid if the systems rather than the model components are considered.

3.6 Analysis

A Bio-PEPA system is an *intermediate, formal, compositional* representation of the biological model. Based on this representation we can perform different kinds of analysis. In this section we discuss briefly how to use a Bio-PEPA system to derive a *CTMC with levels*, a set of *Ordinary Differential Equations (ODEs)*, a *Gillespie simulation* and a *PRISM* model.

From Bio-PEPA to a CTMC As for the reagent-centric view of PEPA, the CTMC associated with the system refers to the concentration levels of the species components. Specifically, the states of the CTMC are defined in terms of concentration levels and the transitions from one state to the other capture some variations in these levels. Hereafter we call the CTMC derived from a Bio-PEPA system (or from a PEPA reagent-centric view system) *CTMC with levels*.

Theorem 1. For any finite Bio-PEPA system $\mathcal{P} = \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, \text{Comp}, P \rangle$, if we define the stochastic process $X(t)$ such that $X(t) = P_i$ indicates that the system behaves as the component P_i at time t , then $X(t)$ is a Markov Process.

The proof is not reproduced here but it is analogous the one presented for PEPA [42]. Instead of the PEPA activity we consider the label γ and the rate is obtained by evaluating the functional rate in the system. We consider finite models to ensure that a solution for the CTMC is at least theoretically feasible (in practice the size of the state space may make the model intractable). The finiteness assumption is equivalent to supposing that each species in the model has a maximum level of concentration.

Theorem 2. Given $(\tilde{\mathcal{P}}, \Gamma, \rightarrow_s)$, let \mathcal{P} be a Bio-PEPA system, with model component P . Let $n_c = |ds(P)|$, where $ds(P)$ is the derivative set of P . Then the infinitesimal generator matrix of the CTMC for \mathcal{P} is a square matrix Q ($n_c \times n_c$) whose elements $q_{u,v}$ are defined as

$$q_{u,v} = \sum_{\alpha_j \in \mathcal{A}(P_u|P_v)} r_{\alpha_j} \quad \text{if } u \neq v \quad \quad q_{u,u} = - \sum_{u \neq v} q_{u,v} \quad \text{otherwise.}$$

where P_u, P_v are two derivatives of P .

It is worth noting that the states of the CTMC are defined in terms of the derivatives of the model component. These derivatives are uniquely identified by the levels of species components in the system, so we can give the following definition of the CTMC states:

Definition 20. The CTMC states derived from a Bio-PEPA system can be defined as vectors of levels $\sigma = (l_1, l_2, \dots, l_n)$, where l_i , for $i = 1, 2, \dots, n$ is the level of the species i and n is the total number of species.

Note that we can avoid consideration of the two levels for *Res* and *CF* as they are always constant.

From this we can deduce that if two transitions are possible between a pair of states, the actions involved are different and they represent reactions that differ only in the modifiers and/or the number of enzymes used. The former point follows from the definition of well-defined Bio-PEPA system. The second point follows because the only possibility of having two transitions between two given states is that the associated reactions have the same reactants and products. We can see this by observing that the states depend on the levels and the reactions cause some changes in these levels. The only elements that do not change during a reaction are the modifiers.

The objective in forming the CTMC with levels is to generate a discrete state space model for which the state space is not prohibitively large. Such a model can then be subjected to numerical analysis, deriving the transient or steady state probability distribution over the states of the model. This form of analysis simultaneously considers all possible behaviours of the model. This is quite distinct from stochastic simulation, also based on a CTMC, which only considers a single trajectory over the state space of the model in each run, i.e. each run captures only one possible behaviour of the model.

From Bio-PEPA to ODEs The translation into ODEs is similar to the method proposed for PEPA (reagent-centric view) [13]. It is based on the syntactic presentation of the model and on the derivation of the stoichiometry matrix $D = \{d_{ij}\}$ from the definition of the components. The entries of the matrix are the stoichiometric coefficients of the reactions and are obtained in the following way: for each component C_i consider the prefix subterms C_{ij} representing the contribution of the species i to the reaction j . If the term represents a reactant we write the corresponding stoichiometry κ_{ij} as $-\kappa_{ij}$ in the entry d_{ij} . For a product we write $+\kappa_{ij}$ in the entry d_{ij} . All other cases are null.

Let t_{ODE} denote the mapping of a Bio-PEPA system \mathcal{P} into a set of ODEs. This mapping is based on the following steps:

1. Definition of the stoichiometry ($n \times m$) matrix D , where n is the number of species and m is the number of molecules;
2. Definition of the *kinetic law vector* ($m \times 1$) v_{KL} containing the kinetic laws of each reaction;
3. Association of the variable x_i with each component C_i and definition of the vector ($n \times 1$) \bar{x} .

The ODE system is then obtained as:

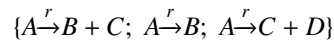
$$\frac{d\bar{x}}{dt} = D \times v_{KL}$$

with initial concentrations $x_{i0} = I_{i0} \times h$, for $i = 1, \dots, n$.

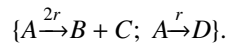
The following property holds:

Property 1. For a biochemical network \mathcal{M} and a Bio-PEPA system $\mathcal{P} = tr_BM_BP(\mathcal{M})$, we have that $t_{ODE}(\mathcal{P}) = t_{BODE}(\mathcal{M})$, where t_{ODE} and t_{BODE} are the translation functions from Bio-PEPA and the biological system into ODEs, respectively.

The ODE system derived from a Bio-PEPA system \mathcal{P} is “equal” to the one obtained directly from the biological network itself. This means that in the translation into Bio-PEPA no information for the derivation of ODEs is lost. This result is unsurprising since in both cases the construction of the ODEs is based on stoichiometric matrix. However the Bio-PEPA model can generally collect more information than the respective ODEs. This can be seen by considering examples which give rise to the same set of ODEs but which differ in their Bio-PEPA representation. For example consider the Bio-PEPA models corresponding to the following sets of reactions:



and



The two Bio-PEPA models are different, but the ODE systems that we derive from them coincide.

From Bio-PEPA to stochastic simulation Gillespie’s stochastic simulation algorithm [36] is a widely-used method for the simulation of biochemical reactions. It applies to homogeneous, well-stirred systems in thermal equilibrium and constant volume, composed of n different species that interact through m reactions. Broadly speaking, the goal is to describe the evolution of the system $\mathbf{X}(t)$, described in terms of the number of elements of each species, starting from an initial state. Every reaction is characterised by a stochastic rate constant c_j , termed the *basal rate* (derived from the constant rate r by means of some simple relations proposed in [36, 68]). Using this it is possible to calculate the *actual rate* $a_j(\mathbf{X}(t))$ of the reaction, that is the probability of the reaction R_j occurring in time $(t, t + \Delta t)$ given that the system is in a specific state.

The algorithm is based on the following two steps:

- Calculation of the next reaction to occur in the system;
- Calculation of the time at which that reaction occurs.

The calculations are based on two conditional density functions:

$$p(j | \mathbf{X}(t)) = a_j(\mathbf{X}(t))/a_0,$$

that is, the probability that the next reaction is R_j and

$$p(\tau | \mathbf{X}(t)) = a_0 e^{-a_0 \mathbf{X}(t) \tau},$$

the probability that the next reaction occurs in $[t + \tau, t + \tau + d\tau]$, where $a_0 = \sum_{v=1}^m a_v(\mathbf{X}(t))$.

The translation of a Bio-PEPA model to a simulation model amenable to Gillespie’s algorithm is similar to the approach proposed for ODEs. The main drawbacks are the definition of the rates and the correctness of the approach in the case of general kinetic laws. Indeed Gillespie’s stochastic simulation algorithm supposes elementary reactions and constant rates (mass-action kinetics). If the model contains only this kind of reactions the translation is straightforward. If there are non-elementary reactions and general kinetic laws, a widely-used approach is to consider them translated directly into a stochastic context. This is not always valid and some counterexamples have been demonstrated [11]. The authors of [11] showed that when Gillespie’s algorithm is applied to Hill kinetics in the context of the transcription initiation of autoregulated genes, the magnitude of fluctuations is overestimated. The application of Gillespie’s algorithm in the case of general kinetic laws is discussed by several authors [1, 17]. Rao and Arkin [1] show that this approach is valid in the case of some specific kinetic laws, such as Michaelis-Menten and inhibition. However, it is important to remember that these laws are approximations, based on some assumptions that specific conditions (such as “ $S \gg E$ ” in the case of Michaelis-Menten) hold. The approach we advocate is as in [47]: we apply Gillespie’s algorithm, but particular attention must be paid to the interpretation of the simulation results and to their validity.

The definition of a Gillespie model is based on:

- Definition of the state vector \vec{X} . It is composed of n components X_i , representing the number of elements for each species i .

- Definition of the initial condition \bar{X}_0 . The values are given by:

$$X_{i0} = l_{i0} \times h \times N_A \times v_i \text{ molecules}$$

where N_A is the Avogadro number, i.e. the number of molecules in a mole of a substance, and v_i is the volume of the containing compartment V_i .

- Definition of the actual rate for each reaction. We have two cases:
 1. Reactions described by the mass-action law and with constant rate r_j . The actual rate for the reaction is:

$$a_j(\bar{X}_j) = c_j \times f_h(\bar{X}_j)$$

where c_j is the stochastic rate constant, f_h is a function that gives the number of distinct combinations of reactant molecules and \bar{X}_j are the species involved in the reaction j . The stochastic rate constant is defined in [68] as:

$$c_j = \frac{r_j}{(N_A \times v)^{n_{tot}-1}} \times \prod_{u=1}^{n_j} \kappa_{uj}!$$

where n_j is the number of distinct reactants in the reaction j , r_j is the rate of the reaction and $n_{tot} = \sum_{u=1}^{n_j} \kappa_{uj}$ is the total number of reactants⁶.

Finally, the number of possible combinations of reactants is defined as

$$f_h(\bar{X}_j) = \prod_{u=1}^{n_j} \binom{X_{p(u,j)}}{\kappa_{uj}} \sim \frac{\prod_{u=1}^{n_j} (X_{p(u,j)})^{\kappa_{uj}}}{\prod_{u=1}^{n_j} \kappa_{uj}!}$$

2. Reactions with general kinetic laws $f_{\alpha_j}(\bar{k}, \bar{C})$. The actual rate is:

$$a_j(\bar{X}_j) = f_{\alpha_j}(\bar{k}, \bar{X}_j)$$

From Bio-PEPA to PRISM PRISM [61] is a probabilistic model checker, a tool for the formal modelling and analysis of systems which exhibit random or probabilistic behaviour. PRISM has been used to analyse systems from a wide range of application domains. Models are described using the PRISM language, a simple state-based language and it is possible to specify quantitative properties of the system using a temporal logic, called *CSL* [2] (Continuous Stochastic Logic). For our purposes the underlying mathematical model of a PRISM model is a CTMC and the PRISM models we generate from Bio-PEPA correspond to the CTMCs with levels. However we present the translation separately as the models are specified in the PRISM language.

⁶ We assume that all the species that are involved in the reaction as reactants are inside the same compartment with volume v .

The PRISM language is composed of *modules* and *variables*. A model is composed of a number of modules which can interact with each other. A module contains a number of local variables. The values of these variables at any given time constitute the state of the module. 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. Each update describes a transition which the module can make if the guard is true. A transition is specified by giving the new values of the variables in the module, possibly as a function of other variables. Each update is also assigned a probability (or in some cases a rate) which will be assigned to the corresponding transition. It is straightforward to translate a Bio-PEPA system into a PRISM model. We have the following correspondences:

- The model is defined as **stochastic** (this term is used in PRISM for CTMC).
- Each element in the set of parameters \mathcal{K} is defined as a *global constant*.
- The maximum levels, the concentration steps and the volume sizes are defined as *global constants*.
- Each species component is represented by a *PRISM module*. The species component concentration is represented by a *local variable* and it can (generally) assume values between 0 and N_i . For each sub-term (i.e. reaction where the species is involved) we have a definition of a *command*. The name of the command is related to the action α (and then to the associated reaction). The guards and the change in levels are defined according to whether the element is a reactant, a product or a modifier of the reactions.
- The functional rates are defined inside an auxiliary module.
- In PRISM the rate associated with an action is the *product* of the rates of the commands in the different modules that cooperate. For each reaction, we give the value “1” to the rate of each command involved in the reaction, with the exception of the command in the module containing the functional rates. In this case the rate is the functional rate f , expressing the kinetic law. The rate associated with a reaction is given by $1 \times 1 \times \dots \times f = f$, as desired.

4 Examples

This section reports the translation of two biological models into Bio-PEPA and some analysis results. The first example is taken from [37] and describes a minimal model for the cascade of post-translational modifications that modulate the activity of cdc2 kinase during the cell cycle. The second example is the repressilator [33], a synthetic genetic network with an oscillating behaviour.

In the present work the stochastic and deterministic simulations are obtained exporting the Bio-PEPA system by means of the derivations described in Section 3.6. An automatic translation is under implementation.

4.1 The Goldbeter’s model

In the following we show the translation of the Goldbeter’s model presented in [37] into Bio-PEPA and we discuss the kinds of analysis that are possible for it. Broadly speaking, the model describes the activity of the *protein cyclin* in the cell cycle. The cyclin

promotes the activation of a *cdk* (*cdc2*) which in turn activates a *cyclin protease*. This protease promotes cyclin degradation, and thus a negative feedback loop is obtained.

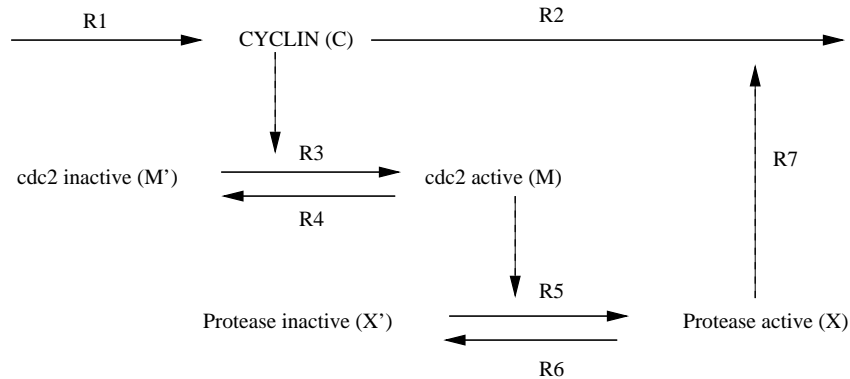


Fig. 2. Goldbeter's model.

The biological model A schema of the model is shown in Fig. 2. There are three distinct species involved:

- *cyclin*, the protein protagonist of the cycle, represented by variable C ;
- *cdc2 kinase*, in both active (i.e. dephosphorylated) and inactive form (i.e. phosphorylated). The variables used to represent them are M and M' , respectively;
- *cyclin protease*, in both active (i.e. phosphorylated) and inactive form (i.e. dephosphorylated). The variables are X and X' .

A detailed list of reactions is reported in Table 2. The first two reactions are the creation of cyclin and its degradation. The reactions $R3$ - $R6$ are enzymatic reactions describing the activation/deactivation of the biological species *cdc2* and protease. These reactions are activated through phosphorylation/dephosphorylation. The last reaction is the degradation of the cyclin triggered by the protease.

Concerning the kinetic laws, the first two reactions have mass-action kinetics, whereas the others all have Michaelis-Menten kinetics. We have some kinetic laws in which the enzyme is explicit (reactions 3, 5), other ones in which it is implicit (reactions 4, 6) as it is constant and abstracted within the Michaelis-Menten parameter V_i .

The Bio-PEPA system The translation of the Goldbeter's model into Bio-PEPA is achieved in the following steps.

- *Definition of the list \mathcal{V} .* In the model compartments are not considered. Here we add the default compartment:

$$cell : 1.0 \times 10^{-14} \text{ litre}$$

id	name	react.	prod.	mod.	kinetic laws
R1	creation of cyclin	-	C	-	v_i
R2	degradation of cyclin	C	-	-	$k_d \times C$
R3	activation of cdc2 kinase	M'	M	C	$\frac{C \times V_1}{(K_c + C)} \frac{M'}{(K_1 + M')}$
R4	deactivation of cdc2 kinase	M	M'	-	$\frac{M \times V_2}{(K_2 + M)}$
R5	activation of cyclin protease	X'	X	M	$\frac{X' \times M \times V_3}{(K_3 + X')}$
R6	deactivation of cyclin protease	X	X'	-	$\frac{X \times V_4}{K_4 + X}$
R7	degradation of cyclin triggered by protease	C	-	X	$\frac{C \times V_d \times X}{C + K_d}$

Table 2. Goldbeter model. The list of reactions.

– *Definition of the set \mathcal{N} .* This is defined as:

$$\begin{aligned}
 C &: h, N_C, 0.01, 0.6, \text{cell}; & M' &: h, N_{M'}, 0.99, 1, \text{cell}; & M &: h, N_M, 0.01, 0.7, \text{cell}; \\
 X' &: h, N_{X'}, 0.99, 1, \text{cell}; & X &: h, N_X, 0.01, 0.65, \text{cell}; \\
 Res &: 1, 1, \rightarrow, \rightarrow, \text{cell}; & CF &: 1, 1, \rightarrow, \rightarrow, \text{cell};
 \end{aligned}$$

The components Res and CF are added to represent degradation reactions and the synthesis of the cyclin, respectively. The information about the initial and the maximum concentrations are derived from Goldbeter's paper. We can fix the step size as 0.05. In this case the maximum levels are: $N_C = 12$, $N_M = 14$, $N_X = 13$, $N_{M'} = N_{X'} = 20$. If we wanted to consider the finer granularity $h = 0.01$ (corresponding to the initial concentration of some of the species) we would have $N_C = 60$, $N_M = 70$, $N_X = 65$, $N_{M'} = N_{X'} = 100$.

– *Definition of functional rates (\mathcal{F}_R) and parameters (\mathcal{K}).* The functional rates are:

$$\begin{aligned}
 f_{a_1} &= fMA(v_i); & f_{a_2} &= fMA(k_d); & f_{a_4} &= fMM(V_2, K_2); \\
 f_{a_5} &= fMM(V_3, K_3); & f_{a_6} &= fMM(V_4, K_4); & f_{a_7} &= fMM(V_d, K_d);
 \end{aligned}$$

$$f_{a_3} = fMM'((V_1, K_c, K_1), M', C) = \frac{V_1 \times C}{K_c + C} \frac{M'}{K_1 + M'};$$

The parameters are those reported in the original paper and we have:

$$\begin{aligned}
 v_i &= 0.025 \mu M \cdot \text{min}^{-1}; & k_d &= 0.01 \text{ min}^{-1}; & V_1 &= 12 \mu M \cdot \text{min}^{-1}; & K_1 &= 0.02 \mu M; \\
 V_2 &= 1.5 \mu M \cdot \text{min}^{-1}; & K_2 &= 0.02 \mu M; & V_3 &= 12 \text{ min}^{-1}; & K_3 &= 0.02 \mu M; \\
 V_d &= 0.0625 \mu M \cdot \text{min}^{-1}; & V_4 &= 2 \mu M \cdot \text{min}^{-1}; & K_4 &= 0.02 \mu M; & K_d &= 0.02 \mu M; \\
 K_c &= 0.5 \mu M
 \end{aligned}$$

– Definition of species components (*Comp*) and of the model component (*P*).

$$\begin{aligned}
C &\stackrel{\text{def}}{=} (\alpha_1, 1)\uparrow C + (\alpha_2, 1)\downarrow C + (\alpha_7, 1)\downarrow C + (\alpha_3, 1) \oplus C; \\
M' &\stackrel{\text{def}}{=} (\alpha_4, 1)\uparrow M' + (\alpha_3, 1)\downarrow M'; \\
M &\stackrel{\text{def}}{=} (\alpha_3, 1)\uparrow M + (\alpha_4, 1)\downarrow M + (\alpha_5, 1) \oplus M; \\
X' &\stackrel{\text{def}}{=} (\alpha_6, 1)\uparrow X' + (\alpha_5, 1)\downarrow X'; \\
X &\stackrel{\text{def}}{=} (\alpha_5, 1)\uparrow X + (\alpha_6, 1)\downarrow X + (\alpha_7, 1) \oplus X; \\
Res &\stackrel{\text{def}}{=} (\alpha_2, 1) \odot Res; \\
CF &\stackrel{\text{def}}{=} (\alpha_1, 1) \odot CF;
\end{aligned}$$

$$C(l_{0C}) \underset{[\alpha_3]}{\boxtimes} M(l_{0M}) \underset{[\alpha_3, \alpha_4]}{\boxtimes} M'(l_{0M'}) \underset{[\alpha_5, \alpha_7]}{\boxtimes} X(l_{0X}) \underset{[\alpha_5, \alpha_6]}{\boxtimes} X'(l_{0X'}) \underset{[\alpha_2]}{\boxtimes} Res(0) \underset{[\alpha_1]}{\boxtimes} CF(1)$$

The levels are chosen to reflect the initial values of the species and are set to $l_{0C} = l_{0M} = l_{0X} = 0$ and $l_{0M'} = l_{0X'} = 20$.

Analysis In the following we report some observations about the analysis of the Bio-PEPA system.

SLTS and CTMC By considering the step size $h = 0.05$ and the number of levels given in the Bio-PEPA system we obtain a CTMC with 52 states and 185 transitions. The states are described by the vector:

$$(C(l_C), M'(l_{M'}), M(l_M), X'(l_{X'}), X(l_X))$$

where the different components can assume different values according to the possible number of levels for each species. This CTMC is not depicted.

Instead, for illustrative purposes, we present a simpler CTMC for our model, obtained assuming $h = 1$ and considering only two levels for each species. The vector \mathcal{N} is modified accordingly. We show how to define the states and the transition rates of this CTMC starting from the Bio-PEPA system and the associated transition system. The initial situation is with C , M and X absent (0) and the other elements present (1). The initial state is $(C(0), M'(1), M(0), X'(1), X(0))$. Figure 3 reports the stochastic transition system in this simplified case.

The numbers indicate the different transitions. Each transition is characterized by a label γ_i containing the information about the action type and the rate. We have:

$$\begin{array}{llll}
\gamma_1 = (\alpha_1, r_1) & \gamma_2 = (\alpha_2, r_2) & \gamma_3 = (\alpha_3, r_3) & \gamma_4 = (\alpha_4, r_4) \\
\gamma_5 = (\alpha_5, r_5) & \gamma_6 = (\alpha_6, r_6) & \gamma_7 = (\alpha_4, r_7) & \gamma_8 = (\alpha_3, r_8) \\
\gamma_9 = (\alpha_1, r_9) & \gamma_{10} = (\alpha_2, r_{10}) & \gamma_{11} = (\alpha_7, r_{11}) & \gamma_{12} = (\alpha_4, r_{12}) \\
\gamma_{13} = (\alpha_3, r_{13}) & \gamma_{14} = (\alpha_5, r_{14}) & \gamma_{15} = (\alpha_6, r_{15}) & \gamma_{16} = (\alpha_4, r_{16}) \\
\gamma_{17} = (\alpha_2, r_{17}) & \gamma_{18} = (\alpha_6, r_{18}) & \gamma_{19} = (\alpha_1, r_{19}) & \gamma_{20} = (\alpha_2, r_{20}) \\
\gamma_{21} = (\alpha_7, r_{21}) & \gamma_{22} = (\alpha_1, r_{22}) & \gamma_{23} = (\alpha_6, r_{23}) &
\end{array}$$

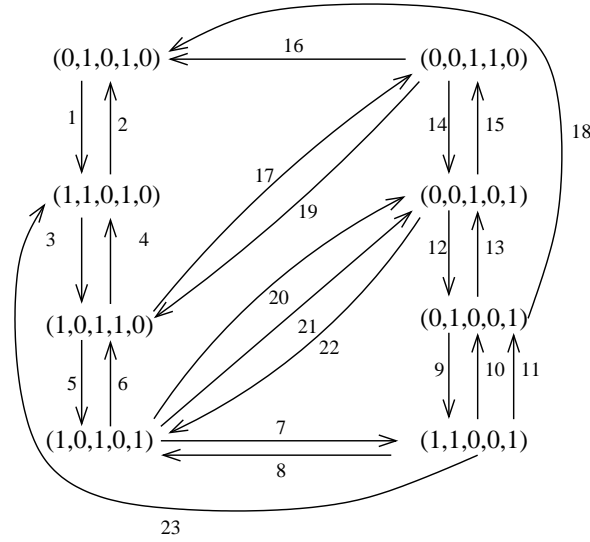


Fig. 3. The transition system for the Goldbeter's model in the case of two levels.

where

$$\begin{aligned}
 r_1 = r_9 = r_{19} = r_{22} = v_i &= 0.025, & r_2 = r_{10} = r_{20} = r_{17} &= k_d \times C = 0.0001, \\
 r_3 = r_{13} &= \frac{V_1 * C}{K_c + C} \frac{M'}{K_1 + M'} = 0.23, & r_4 = r_7 = r_{12} = r_{16} &= \frac{V_2 \times M}{K_2 + M} = 2.66, \\
 r_5 = r_{14} &= \frac{V_3 \times M \times X'}{K_3 + X'} = 0.117, & r_6 = r_{15} = r_{23} = r_{18} &= \frac{V_4 \times X}{K_4 + X} = 2.66, \\
 r_{11} = r_{21} &= \frac{V_d \times C \times X}{(K_d + C)} = 0.00086
 \end{aligned}$$

The states and transitions of the CTMC correspond directly to those of the SLTS with the exception of the case when there are multiple transitions between the same two states. In this example we have only two transitions in the CTMC whose rate is the sum of the rates of two single transitions in the SLTS. In the graph above these cases correspond to the degradation of cyclin, that can happen both with and without the protease. In the CTMC the rate associated with the transition between the states (1, 0, 1, 0, 1) and (0, 0, 1, 0, 1) and between (1, 1, 0, 0, 1) and (0, 1, 0, 0, 1) is given by the sum of the rates of the two degradation reactions $k_d \times M_C + \frac{V_d \times C \times X}{(K_d + C)} = 0.00096 \mu M.min^{-1}$. The rates associated with the other transitions are the ones contained in the labels γ_i above.

ODEs The stoichiometry matrix D associated with the Bio-PEPA system above is

	R1	R2	R3	R4	R5	R6	R7	
C	+1	-1	0	0	0	0	-1	x_C
M'	0	0	-1	+1	0	0	0	$x_{M'}$
M	0	0	+1	-1	0	0	0	x_M
X'	0	0	0	0	-1	+1	0	$x_{X'}$
X	0	0	0	0	+1	-1	0	x_X

The vector that contains the kinetic laws is:

$$v_{KL}^T = \left(v_i \times 1, k_d \times x_C, \frac{V_1 \times x_C}{K_c + x_C} \frac{x_{M'}}{(K_1 + x_{M'})}, \frac{V_2 \times x_M}{(K_2 + x_M)}, \frac{V_3 \times x_M \times x_{X'}}{(K_3 + x_{X'})}, \right. \\ \left. \frac{V_4 \times x_X}{(K_4 + x_X)}, \frac{V_d \times x_C \times x_X}{(K_d + x_C)} \right)$$

where “T” indicates the transpose of the vector. The system of ODEs is obtained as $\frac{d\bar{x}}{dt} = D \times v_{KL}$, with $\bar{x}^T := (x_C, x_{M'}, x_M, x_{X'}, x_X)$, the vector of the species variables:

$$\begin{aligned} \frac{dx_C}{dt} &= v_i \times 1 - k_d \times x_C - \frac{V_d \times x_C \times x_X}{(K_d + x_C)}; \\ \frac{dx_{M'}}{dt} &= -\frac{V_1 \times x_C}{K_c + x_C} \times \frac{x_{M'}}{(K_1 + x_{M'})} + \frac{V_2 \times x_M}{(K_2 + x_M)}; \\ \frac{dx_M}{dt} &= \frac{V_1 \times x_C}{K_c + x_C} \times \frac{x_{M'}}{(K_1 + x_{M'})} - \frac{V_2 \times x_M}{(K_2 + x_M)}; \\ \frac{dx_{X'}}{dt} &= -\frac{V_3 \times x_M \times x_{X'}}{(K_3 + x_{X'})} + \frac{V_4 \times x_X}{(K_4 + x_X)}; \\ \frac{dx_X}{dt} &= \frac{V_3 \times x_M \times x_{X'}}{(K_3 + x_{X'})} - \frac{V_4 \times x_X}{(K_4 + x_X)}; \end{aligned}$$

The initial conditions are the ones reported in the set \mathcal{N} . It is worth noting that the system is equivalent, after some arithmetic manipulations, to the ODE model presented in [37]. The analysis of the model using ODEs is reported in Figure 8. The graphs coincide with results in the original paper.

PRISM The full translation of the model into PRISM is reported in the Appendix A. The number of levels, the maximum concentrations and the parameters used in the kinetic laws are expressed using global constants. For each species a module is constructed. The module representing the cyclin is:

```
module cyclin
cyclin : [0..Nc] init 0;
[creationC] cyclin < Nc → (cyclin' = cyclin + 1);
[degradationC] cyclin > 0 → (cyclin' = cyclin - 1);
[activationM] cyclin > 0 → (cyclin' = cyclin);
[degradationCX] cyclin > 0 → (cyclin' = cyclin - 1);
endmodule
```

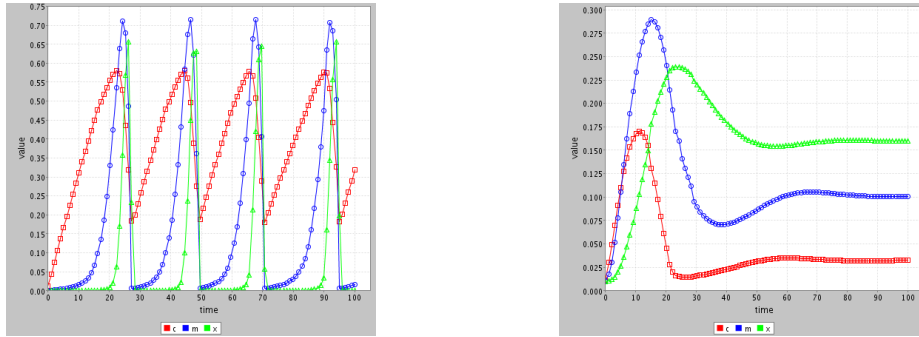


Fig. 4. ODE simulation results for two different instantiations of the model. The two instantiations differ only in the values of the Michaelis-Menten constants. For K_i $i = 1, 2, 3, 4$ we have that $K_i = 0.02 \mu M$ for the graph on the left and $K_i = 40 \mu M$ for the graph on the right. The initial concentrations are those reported in the Goldbeter’s original paper: $0.01 \mu M$ for C , X and M . The simulation time is 100 minutes. In first instantiation of the model, depicted in the figure on the left, we have sustained oscillations whereas in the second, depicted in the figure on the right, we have no oscillations.

The variable cyclin is *local* and represents the species “cyclin”. The possible values are $[0..N_c]$ (where N_c is the maximum level for cyclin) and the initial value is set to 0. Cyclin is involved in four different reactions represented by four commands. The name in the square brackets denotes the reaction. The *guards* are defined according to whether cyclin is a reactant, product or modifier of the reaction (this can be derived from the Bio-PEPA specification of the model). The rate associated with each command is “1” with the exception of the command in the module describing the functional rates. The functional rates are defined in a specific module.

Extension of the model with a control mechanism based on inhibition The authors of [34] proposed an extension of Goldbeter’s model in order to represent a control mechanism for the cell division cycle (CDC). Their approach is based on the introduction of a protein that binds to and inhibits one of the proteins involved in the CDC. This influences the initiation and the conclusion of cell division and modulates the frequency of oscillations. Their approach is based on the basic biochemical network of the CDC oscillations and not on the details of the model so that it may work for other models of this kind. One possible extension for Goldbeter’s model is reported in Figure 5.

Generally speaking, given a general CDC model with l proteins U_1, U_2, \dots, U_l , Gardner *et al.* show that the ODE model is modified in the following way (see [34] for

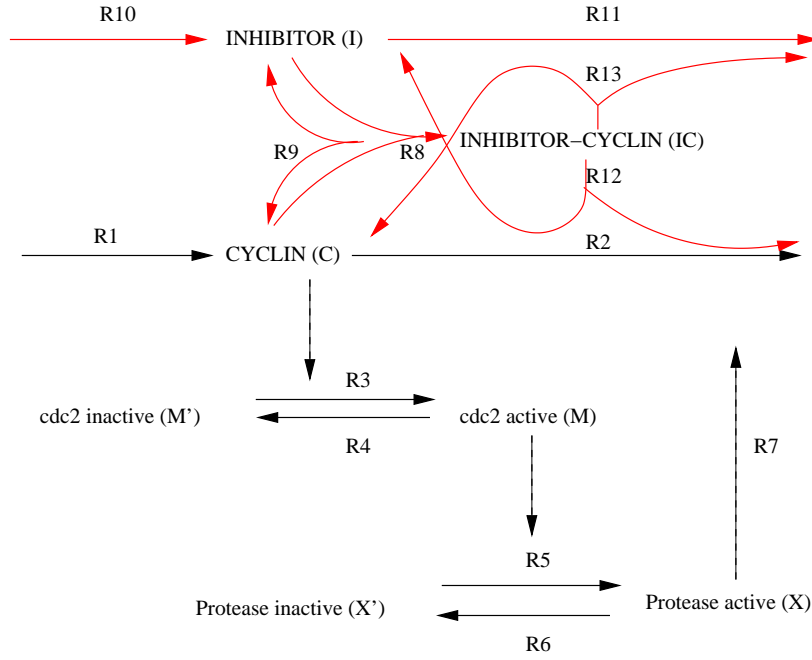


Fig. 5. Extension of the Goldbeter's model. An inhibitor is added.

details):

$$\begin{aligned} \frac{dU_1}{dt} &= f_1(U_1, U_2, \dots, U_l) - a_1 \times U_1 \times Y + (a_2 + \theta \times d_1); \\ \frac{dU_2}{dt} &= f_2(U_1, U_2, \dots, U_l); \\ &\vdots \\ \frac{dU_l}{dt} &= f_l(U_1, U_2, \dots, U_l); \\ \frac{dY}{dt} &= v_s - d_1 \times Y - a_1 \times U_1 \times Y + (a_2 + \theta \times k_d) \times Z; \\ \frac{dZ}{dt} &= a_1 \times U_1 \times Y - (a_2 + \theta \times d_1 + \theta \times k_d) \times Z; \end{aligned}$$

where:

- $f_i(U_1, U_2, \dots)$ with $i = 1, 2, \dots, l$ are the functions of the standard model;
- U_1 is the concentration of the target protein of the inhibitor, Y is the inhibitor and Z denotes the concentration of the inhibition-target complex. U_2, \dots, U_l are the other proteins involved in the cycle;
- a_1 and a_2 are the constant rates for the binding and for the release;
- v_s and d_1 are the rate for the inhibitor synthesis and degradation;

– $\theta < 1$ is the fraction of the degradation rates for the complex Z .

In the following we show how to modify the Bio-PEPA system in order to capture the new reactions and species. Bio-PEPA offers a *compositional approach*: it is possible to compose the whole system by defining the simple subcomponents that compose it. As observed in Section 1, compositionality is one of the main properties of process algebras, that makes them particularly useful for capturing of complex models. In our example, the new reactions and species can indeed be added in a straightforward way, with minor modifications of the system specification. Broadly speaking, we need to define components for the new species, some new terms to describe the new reactions and new functional rates. Finally, the new components are added to the system component.

Here we consider $l = 3$, $U_1 = C$, $U_2 = M$ and $U_3 = X$. The inhibition-target complex Z is thus IC in this case. This complex may dissociate in three distinct ways ($R9$, $R12$ and $R13$ in Figure 5) since each of I and C may be degraded during the dissociation. Note that we could obtain modulation of CDC frequency by using an inhibitor of any of the proteins, so alternative models could be formed with $U_1 = M$ or $U_1 = X$.

We need to extend the Bio-PEPA model in the following way:

$$\begin{aligned}
C &\stackrel{def}{=} \dots + (\alpha_8, 1)\downarrow C + (\alpha_9, 1)\uparrow C + (\alpha_{12}, 1)\uparrow C; \\
&\quad \vdots \\
Res &\stackrel{def}{=} \dots + (\alpha_{11}, 1) \odot Res; \\
CF &\stackrel{def}{=} \dots + (\alpha_{10}, 1) \odot CF; \\
I &\stackrel{def}{=} (\alpha_8, 1)\downarrow I + (\alpha_9, 1)\uparrow I + (\alpha_{10}, 1)\uparrow I + (\alpha_{11}, 1)\downarrow I + (\alpha_{13}, 1)\uparrow I; \\
IC &\stackrel{def}{=} (\alpha_8, 1)\uparrow IC + (\alpha_9, 1)\downarrow IC + (\alpha_{12}, 1)\downarrow IC + (\alpha_{13}, 1)\downarrow IC;
\end{aligned}$$

where I stands for the inhibitor and IC for the inhibitor-cyclin complex in Figure 5. The new functional rates, all described by mass-action kinetics:

$$\begin{aligned}
f_{\alpha_8} &= v_s; & f_{\alpha_9} &= fMA(d_1); & f_{\alpha_{10}} &= fMA(a_1); \\
f_{\alpha_{11}} &= fMA(a_2); & f_{\alpha_{12}} &= fMA(\theta \times d_1); & f_{\alpha_{13}} &= fMA(\theta \times k_d)
\end{aligned}$$

The list of parameters must also be extended to reflect the new elements. Finally the Bio-PEPA model is:

$$\begin{aligned}
C(l_{0C}) &\boxtimes_{[\alpha_3]} M(l_{0M}) \boxtimes_{[\alpha_3, \alpha_4]} M'(l_{0M'}) \boxtimes_{[\alpha_5, \alpha_7]} X(l_{0X}) \boxtimes_{[\alpha_5, \alpha_6]} X'(l_{0X'}) \\
&\quad \boxtimes_{[\alpha_2]} Res(0) \boxtimes_{[\alpha_1]} CF(I) \boxtimes_{[\alpha_8, \alpha_9, \alpha_{10}, \alpha_{11}]} I(l_{0I}) \boxtimes_{[\alpha_8, \alpha_9, \alpha_{12}, \alpha_{13}]} IC(l_{0IC})
\end{aligned}$$

The results of the ODE simulations corresponding to the new model are reported in Fig. 6.

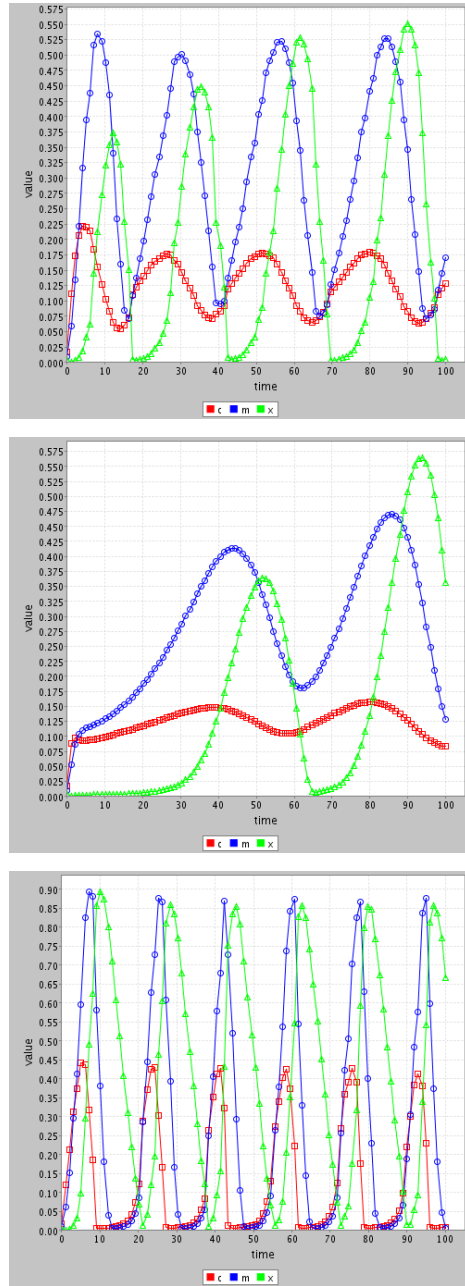


Fig. 6. ODE simulation results for the extended model. The parameters of Goldbeter's model are as before. For the new parameters, in all the graphs $d_1 = 0.05$, $\theta = 0.1$ and $K_{diss} = \frac{a_1}{a_2} = 1$. The top graph corresponds to $a_1 = a_2 = 0.3$ and $v_s = 0.6$, the middle graph to $a_1 = a_2 = 0.7$ and $v_s = 1.4$ and the lower graph to $a_1 = a_2 = 0.05$ and $v_s = 0.1$. The initial values of C, X, M and I are $0.01\mu M$. Simulation time is 100 minutes.

4.2 The repressilator

The *repressilator* is a synthetic genetic regulatory network with oscillating behaviour reported in [33]. The repressilator consists of three genes connected in a *feedback loop*, such that the transcription of a gene is inhibited by one of the other proteins. In the following we present the translation of the original model into Bio-PEPA and we report some analysis results.

The biological model A schema of network is reported in Figure 7.

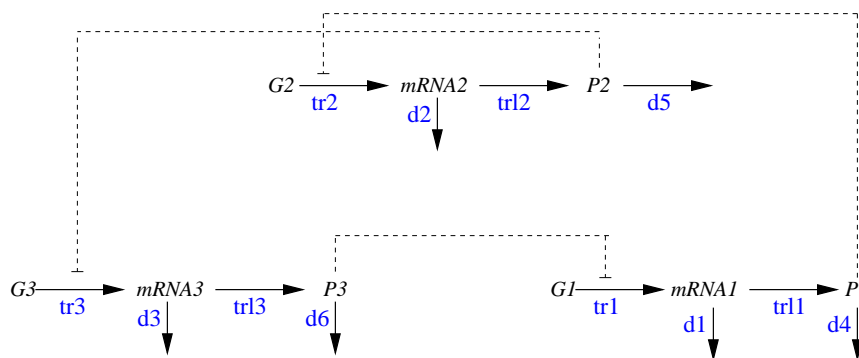


Fig. 7. Repressilator model.

The species involved are:

- Three kinds of genes, hereafter denoted $G1$, $G2$, $G3$. These represent the genes *lacI*, *tetR* and *cI*, respectively.
- The mRNAs transcribed from the three genes, hereafter denoted $mRNA1$, $mRNA2$, $mRNA3$, respectively.
- The proteins corresponding to the three genes, denoted $P1$, $P2$, $P3$, respectively. These represent the proteins associated with the previous genes i.e. *LacI*, *TetR*, *Ci*.

The reactions are:

- The *transcription* of the three mRNAs with inhibition by one of the proteins. These reactions are indicated as $tr1$, $tr2$, $tr3$. The genes are constant and are kept implicit;
- The *translation* of mRNAs into the proteins, indicated as $trl1$, $trl2$, $trl3$;
- *Degradation* of both mRNAs and proteins, indicated as d_i with $i = 1, \dots, 6$.

The transcription reactions are described by Hill kinetics, while the other reactions have mass-action kinetic laws.

The Bio-PEPA system The definition of the Bio-PEPA corresponding to the repressor model is reported below. The parameters and the initial concentrations are one of the possibilities defined in the paper [33].

- *Definition of compartments.* There are no compartments defined explicitly in the model. We consider the default compartment:

$$v_{Cell} : 1;$$

- *Definition of the set \mathcal{N}* It is defined as:

$$\begin{aligned} mRNA1 : 1, 1, 0, \rightarrow, v_{Cell}; & \quad mRNA2 : 1, 1, 0, \rightarrow, v_{Cell}; & \quad mRNA3 : 1, 1, 0, \rightarrow, v_{Cell}; \\ P1 : 1, 1, 5, \rightarrow, v_{Cell}; & \quad P2 : 1, 1, 0, \rightarrow, v_{Cell}; & \quad P3 : 1, 1, 15, \rightarrow, v_{Cell}; \\ Res : 1, 1, \rightarrow, \rightarrow, v_{Cell}; & \quad CF : 1, 1, \rightarrow, \rightarrow, v_{Cell}; \end{aligned}$$

It is worth noting that in the original model the genes are not represented explicitly. In Bio-PEPA we introduce CF to define the transcription. For all the species we consider two levels (*high* and *low*) and step $h = 1$. The initial values (third elements) are those reported in the paper.

- *Definition of the set \mathcal{F}_R and of the set of parameters.* The set of functional rates is:

$$\begin{aligned} f_{ir1} &= fI((\alpha, \alpha 0), P3, 2) = \frac{\alpha}{1 + P3^2} + \alpha 0; \\ f_{ir2} &= fI((\alpha, \alpha 0), P1, 2) = \frac{\alpha}{1 + P1^2} + \alpha 0; \\ f_{ir3} &= fI((\alpha, \alpha 0), P2, 2) = \frac{\alpha}{1 + P2^2} + \alpha 0; \\ f_{ir11} &= fMA(\beta); \\ f_{ir12} &= fMA(\beta); \\ f_{ir13} &= fMA(\beta); \\ f_{di} &= fMA(1) \quad i = 1, 2, 3, 4, 5, 6; \end{aligned}$$

All the three repressors have same behaviour except for their DNA-binding specificities. We assume that all the degradation reactions have rate 1. The other parameters are: $\alpha = 250$; $\alpha 0 = 0$; $\beta = 5$.

These parameters have the following meaning:

- $\alpha 0$ is the number of protein copies per cell produced from a given promoter type during growth in the presence of saturating amounts of the repressor. In the case of the absence of the repressor this number is $\alpha 0 + \alpha$;
- β is the ratio of the protein decay rate to the mRNA decay rate.

- *Definition of the species components.* The species components are:

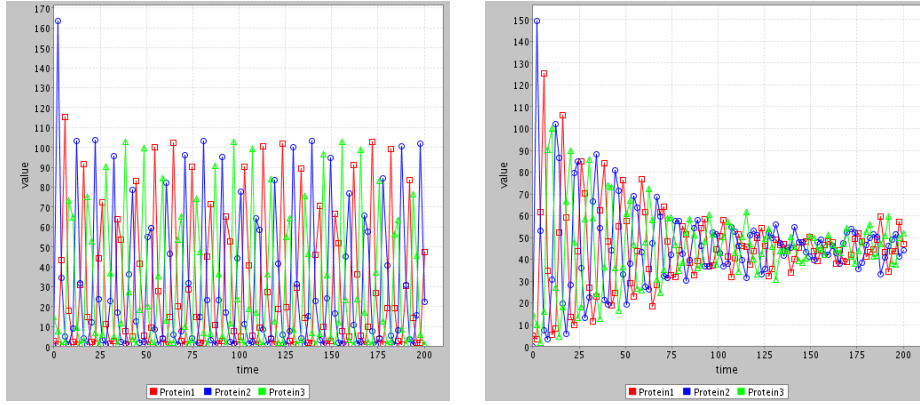


Fig. 8. Analysis of the model: ODE solution is depicted on the left and stochastic simulation results (averaged over 100 runs) are depicted on the right. The parameters are as reported in the text.

$$\begin{aligned}
mRNA1 &\stackrel{def}{=} (d1, 1)\downarrow mRNA1 + (tr1, 1)\uparrow mRNA1 + (trl1, 1)\oplus mRNA1; \\
mRNA2 &\stackrel{def}{=} (d2, 1)\downarrow mRNA2 + (tr2, 1)\uparrow mRNA2 + (trl2, 1)\oplus mRNA2; \\
mRNA3 &\stackrel{def}{=} (d3, 1)\downarrow mRNA3 + (tr3, 1)\uparrow mRNA3 + (trl3, 1)\oplus mRNA3; \\
P1 &\stackrel{def}{=} (d4, 1)\downarrow P1 + (trl1, 1)\uparrow P1 + (tr3, 1)\ominus P1; \\
P2 &\stackrel{def}{=} (d5, 1)\downarrow P2 + (trl2, 1)\uparrow P2 + (tr1, 1)\ominus P2; \\
P3 &\stackrel{def}{=} (d6, 1)\downarrow P3 + (trl3, 1)\uparrow P3 + (tr2, 1)\ominus P3; \\
CF &\stackrel{def}{=} (tr1, 1)\odot CF + (tr2, 1)\odot CF + (tr3, 1)\odot CF; \\
Res &\stackrel{def}{=} (d1, 1)\odot Res + (d2, 1)\odot Res + (d3, 1)\odot Res \\
&\quad + (d4, 1)\odot Res + (d5, 1)\odot Res + (d6, 1)\odot Res;
\end{aligned}$$

– *Definition of the model component.* The model is defined as:

$$\begin{aligned}
&\left(\left(\left(\left(M1(l_{M10}) \right) \right) \right) \right) \bowtie_{\{tr1, tr3\}} M3(l_{M30}) \bowtie_{\{trl1, tr3\}} P1(l_{P10}) \bowtie_{\{tr2, tr1\}} P2(l_{P20}) \bowtie_{\{tr3, tr2\}} \\
&\quad P3(l_{P30}) \bowtie_{\{tr1, tr2, tr3\}} CF(I) \bowtie_{\{d1, d2, d3, d4, d5, d6\}} Res(0)
\end{aligned}$$

The initial levels are defined according to the initial values of the model.

Analysis We consider analysis based on both ODEs and stochastic simulation. The analysis results are reported in Figures 8 and 9. In Figure 8 we have used the parameters

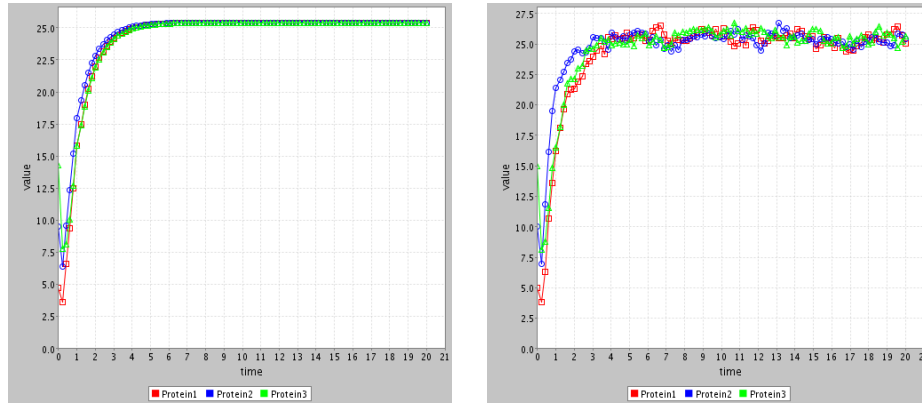


Fig. 9. Analysis of the model: as previously the ODE solution is depicted on the left and the stochastic simulation results (averaged over 100 runs) are depicted on the right. In this case the parameters differ from those reported in the text: α_0 is 25 and the initial values are $P_1 = 5$, $P_2 = 10$ and $P_3 = 15$.

reported in the original paper. On the left, the ODE simulation results are reported. An oscillating behaviour is shown by all three proteins. On the right the results of stochastic simulation, averaged over 100 runs, are reported. In this case the average oscillating behaviour becomes weaker after some time. This is probably due to the slight difference in phase in the 100 runs so that the averaged behaviour no longer shows oscillatory behaviour after some time. Note that varying the values of α and β for the different elements we obtain different amplitudes for the oscillations. In the case of Figure 9, the three proteins reach a steady state, with both ODE and stochastic simulation.

5 Conclusions and future perspectives

In this chapter we have introduced systems biology modelling based on process algebra, particularly focussing on Bio-PEPA. This new formalism is a modification of the process algebra PEPA and is designed specifically for the modelling and the analysis of biochemical networks. Bio-PEPA allows explicit representation of some features of biochemical networks, such as stoichiometry and general kinetic laws which are not readily captured in other process algebra based formalisms. Thus not only elementary reactions with constant rates, but also complex reactions described by general kinetic laws can be considered. Each reaction in the model is associated with an action type and a functional rate. The potential to consider various kinds of kinetic laws permits us to model a vast number of biochemical networks. Indeed complex reactions are frequently found in biologists' models as abstractions of sequences of elementary steps and reducing these to elementary reactions is often undesirable, or even impossible.

Some notions of equivalence have been developed for Bio-PEPA and we have shown how these may be used to compare models. In particular we presented definitions of isomorphism and strong bisimulation which are similar to the relations defined for PEPA

in [42]. These equivalences are motivated by the semantics of the language, in that they are standard notions of equivalence in process algebra, and turn out to be quite strict with respect to the biological systems. In future work we intend to investigate alternative forms of equivalence, motivated more by the biological domain with the aim of finding equivalences which may be used to manipulate and simplify models.

A principal feature of Bio-PEPA is the possibility of mapping the system to different kinds of analysis. A single system description may be interpreted in a number of ways, giving rise to ODEs for a population view of the system, a stochastic simulation for an individuals-based view of the system, or a CTMC with levels for a more abstract view of the system. Of course stochastic simulation is also a CTMC-based representation of the system but the focus on individual molecules means that such a CTMC is only amenable to analysis by simulation. The advantage of the CTMC with levels is that it may have a state space small enough to be tackled by numerical analysis, meaning that a different set of analysis techniques may be used. In particular the numerical solution of a CTMC captures all possible behaviours of the model, whereas each run of a simulation only captures one trajectory through the state space, i.e. one possible behaviour.

Currently, a tool for the analysis of biochemical networks using Bio-PEPA is under implementation and a translation from SBML into Bio-PEPA is planned.

A Appendix A: PRISM specification of the Goldbeter's model

```
//Kind of model
stochastic

//Volume
const double cell = 1;

// Levels
const int Nc = 1;
const int Nm = 1;
const int Nx = 1;
const int Nxi = 1;
const int Nmi = 1;

//Steps
const double Hc = 0.01;
const double Hm = 0.01;
const double Hx = 0.01;
const double Hxi = 0.01;
const double Hmi = 0.01;

//Parameters
const double vi = 0.05;
const double vd = 0.025;
const double kd = 0.01;
const double Kc = 0.5;
const double V1 = 3;
```

```

const double V3 = 1;
const double Kd = 0.2;
const double V2 = 1.5;
const double V4 = 0.5;
const double K1 = 0.005;
const double K2 = 0.005;
const double K3 = 0.005;
const double K4 = 0.005;

//Modules

//module Cyclin
module cyclin
cyclin : [0..Nc] init 0;
[creationC] cyclin<Nc --> 1: (cyclin' = cyclin+1);
[degradationC] cyclin>0 --> 1: (cyclin' = cyclin-1);
[activationM] cyclin>0 --> 1: (cyclin' = cyclin);
[degradationCX] cyclin>0 --> 1: (cyclin' = cyclin-1);
endmodule

//module kinase inactive
module kinasei
kinasei : [0..Nmi] init 1;
[activationM] kinasei>0 --> 1: (kinasei' = kinasei-1);
[deactivationM] kinasei<Nmi --> 1: (kinasei' = kinasei+1);
endmodule

//module kinase active
module kinasea
kinase : [0..Nm] init 0;
[activationM] kinase<Nm --> 1: (kinase' = kinase+1);
[deactivationM] kinase>0 --> 1: (kinase' = kinase-1);
[activationX] kinase>0 --> 1: (kinase' = kinase);
endmodule

//module protease inactive
module proteasei
proteasei : [0..Nxi] init 1;
[activationX] proteasei>0 --> 1: (proteasei' = proteasei-1);
[deactivationX] proteasei<Nxi --> 1: (proteasei' = proteasei+1);
endmodule

//module protease active
module proteasea
protease : [0..Nx] init 0;
[activationX] protease<Nx --> 1: (protease' = protease+1);
[deactivationX] protease>0 --> 1: (protease' = protease-1);
[degradationCX] protease>0 --> 1: (protease' = protease);
endmodule

module Functional_rates

```

```

dummy: bool init true;
[creationC] cyclin<Nc --> vi/Hc: (dummy'=dummy);
[degradationC] cyclin<Nc --> (kd*cyclin*Hc)/Hc: (dummy'=dummy);
[activationM] cyclin>0 & kinasei>0 -->
  ((cyclin*Hc*V1)/(Kc + cyclin*Hc))*((kinasei*Hmi)/(K1+kinasei*Hmi))
  *(1/Hmi): (dummy'=dummy);
[activationX] kinase>0 & proteasei>0 -->
  (kinase*Hm*proteasei*Hxi*V3/(K3+proteasei*Hxi))*(1/Hxi):
  (dummy'=dummy);
[deactivationM] kinase>0 --> ((kinase*Hm*V2)/(K2 +kinase*Hm))*(1/Hm):
  (dummy'=dummy);
[deactivationX] protease>0 --> (protease*Hx*V4/(K4 + protease*Hx))
  *(1/Hx): (dummy'=dummy);
[degradationCX] cyclin>0 & protease>0 -->
  ((cyclin*Hc*vd *protease*Hx)/(cyclin*Hc + Kd))*(1/Hc): (dummy'=dummy);
endmodule

```

References

1. A.P. Arkin and C.V. Rao. Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm. *Journal of Chemical Physics*, volume 11, pages 4999–5010, 2003.
2. A. Aziz, K. Kanwal, V. Singhal and V. Brayton. Verifying continuous time Markov chains. Proc. *8th International Conference on Computer Aided Verification (CAV'96)*, volume 1102 of LNCS, pages 269–276, Springer, 1996.
3. S. van Bakel, I. Kahn, M. Vigliotti and J. Heath. Modelling intracellular fate of FGF receptors with Bio-Ambients. Sixth Workshop on Quantitative Aspects of Programming Languages (QAPL 2008) to appear in *Electronic Notes in Theoretical Computer Science*, 2008.
4. M. Bernardo, R. Gorrieri and L. Donatiello. MPA: A Stochastic Process Algebra. Technical report UBLCS-94-10, Laboratory of Computer Science, University of Bologna, 1994.
5. M. Bernardo and R. Gorrieri. A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, volume 202, pages 1–54, 1998.
6. *The BIOSPI Project*. Available at <http://www.wisdom.weizmann.ac.il/biospi/>.
7. B.J. Bornstein, J.C. Doyle, A. Finney, A. Funahashi, M. Hucka, S.M. Keating, H. Kitano, B.L. Kovitz, J. Matthews, B.E. Shapiro and M.J. Schilstra. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. *Systems Biology*, volume 1, pages 41–53, 2004.
8. *BioModels Database*. <http://www.ebi.ac.uk/biomodels/>
9. L. Bortolussi and A. Policriti. Modeling Biological Systems in Stochastic Concurrent Constraint Programming, Proc. of *WCB 2006*, 2006.
10. L. Brodo, P. Degano and C. Priami. A Stochastic Semantics for Bio-Ambients. In Proc. of *International Conference on Parallel Computing Technologies (PaCT)*, LNCS, volume 4671, pages 22–34, Springer-Verlag, 2007.
11. R. Bundschuh, F. Hayot and C. Jayaprakash. Fluctuations and Slow Variables in Genetic Networks, *Biophys. J.*, volume 84, pages 1606–1615, 2003.
12. N. Busi and C. Zandron. Modeling and analysis of biological processes by membrane calculi and systems. Proc. of the *Winter Simulation Conference (WSC 2006)*.
13. M. Calder, S. Gilmore and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways. Proc. of *CMSB'05*, pages 204–215, 2005.

14. M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Trans. on Computational Systems Biology*, VII, volume 4230 of LNCS, pages 1–23, Springer, 2006.
15. M. Calder, A. Duguid, S. Gilmore and J. Hillston. Stronger computational modelling of signalling pathways using both continuous and discrete-space methods. Proc. of *CMSB'06*, volume 4210 of LNCS, pages 63–77, 2006.
16. M. Calder, V. Vyshemirsky, D. Gilbert and R. Orton. Analysis of Signalling Pathways using Continuous Time Markov Chains. *Trans. on Computational Systems Biology*, VI, volume 4220 of LNCS, pages 44–67, Springer, 2006.
17. Y. Cao, D.T. Gillespie and L. Petzold. Accelerated Stochastic Simulation of the Stiff Enzyme-Substrate Reaction. *J. Chem. Phys.*, volume 123, number 14, pages 144917–144929, 2005.
18. L. Cardelli. Brane Calculi - Interactions of Biological Membranes. *Proc. of Workshop on Computational Methods in Systems Biology (CMSB04)*, LNCS, volume 3082, pages 257–278, 2005.
19. L. Cardelli, E. M. Panina, A. Regev, E. Shapiro and W. Silverman. BioAmbients: An Abstraction for Biological Compartments. *Theoretical Computer Science*, volume 325, number 1, pages 141–167, Elsevier, 2004.
20. N. Chabrier-Rivier, F. Fages and S. Soliman. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, volume 4, pages 64–73, 2004.
21. D. Chiarugi, P. Degano and R. Marangoni. A Computational Approach to the Functional Screening of Genomes. *PLoS Comput. Biol.*, volume 9, pages 1801–1806, 2007.
22. F. Ciocchetta, and J. Hillston. Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. Proc. of *FBTC 2007, Electronic Notes in Theoretical Computer Science*, volume 194, number 3, pages 103–117, 2008.
23. F. Ciocchetta and J. Hillston. Bio-PEPA: a framework for the modelling and analysis of biological systems. Technical Report of the School of Informatics, University of Edinburgh, EDI-INF-RR-1231, 2008.
24. F. Ciocchetta and C. Priami. Biological transactions for quantitative models. Proc. of *MeCBIC 2006*, Electronic Notes in Theoretical Computer Science, volume 171, number 2, pages 55–67, 2007.
25. F. Ciocchetta and C. Priami. Beta-binders with Biological Transactions. Technical report TR-10-2006, The Microsoft Research-University of Trento Centre for Computational and Systems Biology, 2006.
26. G. Costantin, C. Laudanna, P. Lecca, C. Priami, P. Quaglia and B. Rossi. Language modeling and simulation of autoreactive lymphocytes recruitment in inflamed brain vessels. *SIMULATION: Transactions of the Society for Modeling and Simulation International*, volume 80, pages 273–288, 2003.
27. V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, volume 325, number 1, pages 69–110, 2004.
28. V. Danos and J. Krivine. Formal molecular biology done in CCS-R. Proc. of *Workshop on Concurrent Models in Molecular Biology (BioConcur'03)*, 2003.
29. V. Danos, J. Feret, W. Fontana, R. Harmer and J. Krivine. Ruled-based modelling of cellular signalling. Proc. of *CONCUR'07*, LNCS, volume 4703, 2007.
30. V. Danos, J. Feret, W. Fontana and J. Krivine. Scalable simulation of cellular signalling networks. Proc. of *APLAS'07* 2007.
31. V. Danos and J. Krivine. Reversible Communicating Systems. Proc. of *Concur'04*, LNCS, volume 3170, pages 292–307, 2004.

32. C. Eichler-Jonsson, E.D. Gilles, G. Muller and B. Schoeberl. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology*, volume 20, pages 370–375, 2002.
33. M.B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, volume 403, number 6767, pages 335–338, 2000.
34. T.S. Gardner, M. Dolnik, and J.J. Collins. A theory for controlling cell cycle dynamics using a reversibly binding inhibitor. *Proc. Nat. Acad. Sci. USA*, volume 95, pages 14190–14195, 1998.
35. N. Geisweiller, J. Hillston and M. Stenico. Relating continuous and discrete PEPA models of signalling pathways. To appear in *Theoretical Computer Science*, 2007.
36. D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, volume 81, pages 2340–2361, 1977.
37. A. Goldbeter. A Minimal Cascade Model for the Mitotic Oscillator Involving Cyclin and Cdc2 kinase. *Proc. Nat. Acad. Sci.*, volume 8, pages 9107–9111, 1991.
38. N. Götz, U. Herzog and M. Rettelbach. TIPP—a language for timed processes and performance evaluation. Technical report 4/92, IMMD7, University of Erlangen-Nürnberg, Germany, 1992.
39. E.L. Haseltine and J.B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.*, volume 117, pages 6959–6969, 2006.
40. J. Heath, M. Kwiatkowska, G. Norman, D. Parker and O. Tymchyshyn. Probabilistic Model Checking of Complex Biological Pathways. *Theoretical Computer Science (Special Issue on Converging Sciences: Informatics and Biology)*, 2007.
41. H. Hermanns. Interactive Markov Chains: The Quest for Quantified Quality. Volume 2428 of LNCS, Springer, 2002.
42. J. Hillston. *A Compositional Approach to Performance Modelling*, Cambridge University Press, 1996.
43. C.A.R. Hoare. *Communicating sequential processes*, Prentice Hall, International Series in Computer Science, 1985.
44. C.C. Jou and S. Smolka. Equivalences, Congruences and Complete Axiomatizations of Probabilistic Processes. Proc. of *CONCUR'90*, volume 458 of LNCS, pages 367–383, 1990.
45. M. Kanehisa. A database for post-genome analysis. *Trends Genet.*, volume 13, pages 375–376, 1997.
46. *KEGG home page*, available at <http://sbml.org/kegg2sbml.html>.
47. A.M. Kierzek and J. Puchalka. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *BIOPHYS J.*, volume 86, pages 1357–1372, 2004.
48. C. Kuttler, C. Lhoussaine and J. Niehren. A Stochastic Pi Calculus for Concurrent Objects. Technical report RR-6076 INRIA, 2006.
49. C. Kuttler and J. Niehren. Gene regulation in the π -calculus: simulating cooperativity at the lambda switch. *Trans. on Computational Systems Biology VII*, volume 4230 of LNCS, pages 24–55, Springer, 2006.
50. P. Lecca and C. Priami. Cell Cycle control in Eukaryotes: a BioSpi model. Proc. of *Bioconcur 03*, 2003.
51. R. Milner. *Communication and Concurrency*. Prentice Hall, International Series in Computer Science, 1989.
52. R. Milner. *Communicating and mobile systems: the π -calculus*. Cambridge University Press, 1999.
53. F. Moller and C. Tofts. A Temporal Calculus for Communicating Systems. Proc. of *Concur'90*. Volume 458 of LNCS, pages 401–415, 1990.

54. X. Nicollin and J. Sifakis. An Overview and Synthesis on Timed Process Algebras. In *Real-Time: Theory in Practice*. Volume 600 of LNCS, pages 526–548, 1991.
55. *NuMSV model checker*, available at <http://nusmv.irst.itc.it>.
56. N. Le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J.L. Snoep, and M. Hucka. BioModels Database: a Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems. *Nucleic Acids Research*, volume 34, pages D689–D691, 2006.
57. G.D. Plotkin. A Structural Approach to Operational Semantics. Technical report DAIMI FM-19, Computer Science Department, Aarhus University, 1981.
58. C. Priami and P. Quaglia. Beta-binders for biological interactions. *Proc. of CMSB'04*, Volume 3082 of LNCS, pages 20–33, Springer, 2005.
59. C. Priami. Stochastic π -calculus. *The Computer Journal*, volume 38, number 6, pages 578–589, 1995.
60. C. Priami, A. Regev, W. Silverman and E. Shapiro. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, volume 80, pages 25–31, 2001.
61. Prism web site. <http://www.prismmodelchecker.org/>
62. A. Regev. Representation and simulation of molecular pathways in the stochastic π -calculus. *Proc. of the 2nd workshop on Computation of Biochemical Pathways and Genetic Networks*, 2001.
63. A. Regev and E. Shapiro. Cells as computation. *Nature*, volume 419, page 343, 2002.
64. A. Romanel, L. Dematté and C. Priami. The Beta Workbench. Technical report TR-03-2007, The Microsoft Research-University of Trento Centre for Computational and Systems Biology, 2007.
65. I.H. Segel. *Enzyme Kinetics: Behaviour and Analysis of Rapid Equilibrium and Steady-State Enzyme Systems*, Wiley-Interscience, New York, 1993.
66. *SPIM, The stochastic Pi-Machine*, Available at www.doc.ic.ac.uk/~anp/spim/.
67. B. Strulo. Process Algebra for Discrete Event Simulation. PhD Thesis, Imperial College, 1993.
68. O. Wolkenhauer, M. Ullah, W. Kolch and K. H. Cho. Modelling and Simulation of Intra-Cellular Dynamics: Choosing an Appropriate Framework. *IEEE Transactions on NanoBioScience*, volume 3, pages 200–207, 2004.
69. C. Versari and N. Busi. Efficient stochastic simulation of biological systems with multiple variable volumes. *Proc. of FBTC 2007, Electronic Notes in Theoretical Computer Science*, volume 194, number 3, 2008.