Lumpability Abstractions of Rule-based Systems

Jerome Feret[a,1,*], Thomas Henzinger[b], Heinz Koeppl[c,2], Tatjana Petrov[c,3]

[a] *LIENS (INRIA/ÉNS/CNRS), 45, rue d'Ulm, 75230 Paris Cedex, France*
[b] *IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria*
[c] *ETH Zürich, BISON Group, Physikstrasse 3, 8092 Zürich, Switzerland*

Abstract

The induction of a signaling pathway is characterized by transient complex formation and mutual posttranslational modification of proteins. To faithfully capture this combinatorial process in a mathematical model is an important challenge in systems biology. Exploiting the limited context on which most binding and modification events are conditioned, attempts have been made to reduce the combinatorial complexity by quotienting the reachable set of molecular species, into species aggregates while preserving the deterministic semantics of the thermodynamic limit. Recently we proposed a quotienting that also preserves the stochastic semantics and that is complete in the sense that the semantics of individual species can be recovered from the aggregate semantics. In this paper we prove that this quotienting yields a sufficient condition for *weak lumpability* (that is to say that the quotient system is still Markovian for a given set of initial distributions) and that it gives rise to a *backward Markov bisimulation* between the original and aggregated transition system (which means that the conditional probability of being in a given state in the original system knowing that we are in its equivalence class is an invariant of the system). We illustrate the framework on a case study of the epidermal growth factor (EGF)/insulin receptor crosstalk.

Keywords: Markov chains, abstraction, lumpability, bisimulation

1. Introduction

Often a few elementary events of binding and covalent modification [39] in a biomolecular reaction system give rise to a combinatorial number of non-isomorphic reachable species or complexes [22, 23]. Instances of such systems

[*] Corresponding author

[1] Jérôme Feret's contribution was partially supported by the AbstractCell ANR-Chair of Excellence.

[2] Heinz Koeppl's research is supported by the Swiss National Science Foundation, grant no. 200020-117975/1.

[3] Tatjana Petrov's research is supported by SystemsX.ch (the Swiss Initiative in Systems Biology).

are signaling pathways, polymerizations involved in cytoskeleton maintenance, the formation of transcription factor complexes in gene-regulation.

For such biomolecular systems, traditional chemical kinetics face fundamental limitations, that are related to the question of how biomolecular events are represented and translated into a mathematical model [30]. More specifically, chemical reactions can only operate on a collection of fully specified molecular species and each such species gives rise to one differential equation, describing the rate of change of that species' concentration. Many combinatorial systems do not permit the enumeration of all molecular species and thus render their traditional differential description prohibitive. However, even if one could enumerate them, it remains questionable whether chemical reactions are the appropriate way to represent and to reason about such systems.

As the dynamics of a biomolecular reaction mixture comes about through the repeated execution of a few elementary events one may wonder about the effective degrees of freedom of the reaction mixture's dynamics. If the velocity of all events – or their probabilities to occur per time-unit per instance – are different for all molecular species (w.r.t. modification) and pairs of molecular species (w.r.t. binding) to which the events can apply to, then the degrees of freedom would equal to the number of molecular species. However, due to the local nature of physical forces underlying molecular dynamics, the kinetics of most events appear to be ignorant with respect to the global configuration of the molecular species they are operating on. More provocatively, one may say that even if there would be variations of kinetics of an event from one context to another, experimental biology does not – and most likely never will – have the means to discern between all different contexts. For instance, fluorescence resonance energy transfer (FRET), may report on a specific protein-binding event and even its velocity, however we have no means to determine whether the binding partners are already part of a molecular species – not to speak of the composition and modification state of these species. To this end, molecular species remain elusive and appear to be inappropriate entities of descriptions.

To align with the mentioned experimental insufficiencies and with the underlying biophysical locality, rule-based descriptions were introduced as a framework to encode such reaction mixtures succinctly and to enable their mathematical analysis [10, 3]. The biochemical structure of proteins and molecular species is modeled in a transparent way as a graph where nodes are proteins and have a set of binding sites which can be bound pair-wise. Moreover, rules exploit the limited context on which most elementary events are conditioned. They just enumerate that part of a molecular species that is relevant for a rule to be applicable. Thus, in contrast to chemical reactions, a molecular species is not just a name, but documents its biochemical structure and rules can operate on a collection of partially specified molecular species. Consequently, one region of a molecular species being in a particular state may, or may not influence the state of another region of molecular species. The notion of influence is captured by the relation among the sites of molecular species, which we will call *flow of information.* An approximation of such flow of information, formalized as a binary relation over sites, can be derived by only looking at the contexts of rules. The flow of information should not include those pairs of states, whose correlation is irrelevant when tracking the dynamics of the system. As a result, we identify sets of partially specified species – or *fragments* – that allow for a self-consistent description of the rule-set's dynamics. Naturally, as partially specified species usually encompass many fully specified species, the cardinality of that set is less than of the set of molecular species. In [14, 7], these approaches have been used to obtain a self-consistent fragment dynamics based on ordinary differential equations. These equations describe the dynamics in the thermodynamic limit of stochastic kinetics when scaling species multiplicities to infinity while maintaining a constant concentration (multiplicity per unit volume) [27]. In many applications in cellular biology this limiting dynamics is an inappropriate model due to the low multiplicities of some molecular species – think of transcription factor - DNA binding events. Thus stochastic semantics, which takes into account the case of finite populations in finite volume, are often preferred. Yet, the flow of information depends on the chosen semantics, and as illustrated in [13], the flow of information is sparser in semantics based on ordinary differential equations than in stochastic semantics. As a consequence, the obtained *differential fragments* cannot be used to describe stochastic kinetics [15]. Instead, we can derive *stochastic fragments* that represent the effective degrees of freedom in the stochastic case. In contrast to the differential case, stochastic fragments have the property that the probability of being in a concrete state (a state which is counting copy numbers of molecular species) can be recovered from the probability of being in an abstract state (a state which is counting copy numbers of partially specified species). Stochastic fragments could be used to reduce the semantics based on ordinary differential equations, but they would give bigger reduced systems than the ones obtained thanks to differential fragments.

In this paper we translate our abstraction method [15] into the language of well-established contexts of abstraction for probabilistic systems – lumpability and bisimulation. Lumpability is mostly considered from a theoretical point of view in the theory of stochastic processes [24, 17, 36, 33, 34, 4]. A Markov chain is lumpable with respect to a given aggregation (quotienting) of its states, if the lumped chain preserves the Markov property [25]. This property may depend on the initial distribution of the Markov chain. A given Markov chain can be lumpable with respect to a given aggregation of its states for a non-empty subset of initial distributions, in such a case we refer to *weak* lumpability [4, 37]. Whenever a Markov chain is lumpable with respect to a given aggregation of its states for any initial distribution, we refer to *strong* lumpability. Approximate aggregation techniques for Markov chains of biochemical networks are discussed in [19]. Probabilistic bisimulation was introduced as an extension to classic bisimulation in [28]. It is extended to continuous-state and continuous-time in [11] and, for the discrete-state case, to weak bisimulation [2]. For instance, in [11] the authors use bisimulation of labelled Markov processes, the state space of which is not necessarily discrete, and they provide a logical characterization of probabilistic bisimulation. Another notion of weak bisimulation was recently introduced in [12]. Therein two labeled Markov chains are defined to be equiva-

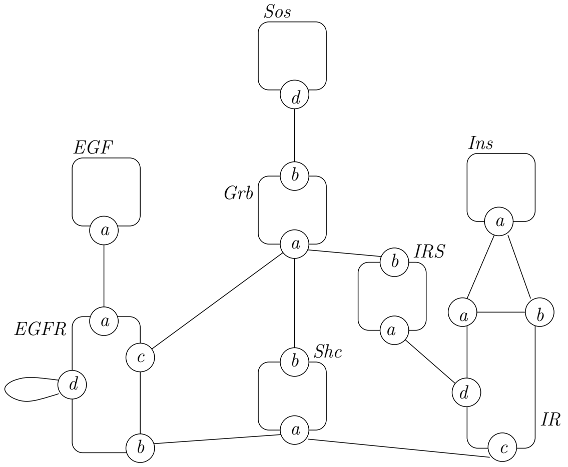


Figure 1: Contact map of the rule-set modeling the crosstalk between the EGF receptor and insulin receptor pathways.

lent if every finite sequence of observations has the same probability of occurring in the two chains. Herein we recognize the sound aggregations of [15] as a form of *backward Markov bisimulations* on weighted labeled transition systems, and we show it to be equivalent to the notion of *weak lumpability* on Markov chains.

The rest of the paper is organized as follows. In the next section, we illustrate on an informal example what the flow of information between the different regions of molecular species is, and how it can be used to reduce the combinatorial complexity of some biological systems. Then, we formalize these intuitions. In Sect. 3, we introduce weighted labeled transition systems and their trace semantics. In Sect. 4, we define the rule-based language, and we assign a weighted labelled transition system (WLTS) to a Kappa specification. In Sect. 5, we give a general procedure to compute stochastic fragments from a set of rules. In Sect. 6, we introduce the characterizations of sound and complete abstractions on weighted labeled transition systems as a backward Markov bisimulation. Moreover, we define it being equivalent to the weak lumpability on Markov chains.

2. Case study

In this section, we informally explain how fragmentation works on a given example. We consider the model of a crosstalk between the *EGF* receptor and the insulin receptor pathways, described in [7]. Two kinds of receptors, *EGF* receptor (*EGFR*) and insulin receptor (*IR*) can recruit a protein called *Sos*.

We give in Fig. 1 a summary of the proteins, and of the potential bindings between proteins, by the mean of a *contact map*, which will be formalized in Sect. 5. A contact map can be extracted automatically from a model written in Kappa [10]. The nodes of a contact map describe the different types of proteins of the model. Each kind of proteins is associated with a set of sites, for instance, the protein *EGFR* has four sites named a,b,c, and d. In the contact map, an edge between the sites of two or of the same protein(s) denotes a potential binding between the sites of two instances of this(these) protein(s). Thus, the edge between the site a of *EGF* and the site a of *EGFR* denotes the fact that the site a of any instance of protein of type *EGF* can be connected to the site a of any instance of protein of type *EGFR*. The edge between the site d of *EGFR* and itself denotes the fact that the sites d of two instances of proteins of type *EGFR* can be bound together. We also notice that some sites are in competition (or in concurrency), since they can be connected to different kinds of sites (as the site a of *Grb* for instance).

The receptors *EGFR* and *IR* have each their own pathway, but these two pathways share some common proteins. We postpone the formal description of the model in Sect. 4, in which the language Kappa is introduced. Firstly, we describe how a receptor *EGFR* can recruit a transport molecule *Grb*. *EGFR* can be activated by binding with a ligand *EGF* on site a. Moreover, two *EGFR*s can bind with each other via their site b and form a dimer. The kinetic rate of the binding between two receptors *EGFR* may depend on the fact that the receptors are connected or not to some ligand(s) *EGF*. Then, a receptor *EGFR* in a dimer can recruit an adapter molecule called Shc and phosphorylate it (the rate depends on the fact whether the receptor is still in a dimer, or not). Shc can then recruit a transport molecule *Grb*. Yet, each receptor has a shorter way to recruit a transport molecule. The site c of a receptor *EGFR* in a dimer can be phosphorylated and then recruit *Grb* directly. Secondly, we describe how an insulin receptor *IR* can recruit *Grb*. A receptor *IR* can recruit insulin molecules Ins on two sites a and b (the rate may depend on the fact whether an insulin molecule has already been recruited). The site c of the *IR* can be phosphorylated at a rate which depends on the number of recruited insulin molecules. Then, *IR* can recruit an adapter Shc. Whenever *IR* is also bound to two insulin molecules, Shc can be phosphorylated. Shc can then recruit *Grb*. But *IR* can also recruit *Grb* by another way. The site d of *IR* can recruit another adapter called *IRS* which can be activated when the insulin receptor is bound to two insulin molecules. Then, *IRS* can recruit *Grb*. Lastly, *Grb* can independently recruit a protein *Sos*. And *Sos* can be phosphorylated at the rate which may depend on the fact whether it is bound to a *Grb*, or not. Moreover, all these interactions are reversible.

In this model, $2,768$ different molecular species may occur. This number is mainly due to the fact that each dimer made of two proteins *EGFR* has 4 sites (the sites b and c for each receptor *EGFR*) to recruit a protein *Grb*, which induces a small combinatorial blow up. To break down this combinatorial blow up, we investigate the flow of information between different areas of molecular species. The flow of information describes the sites whose state value has an influence of the behavior of other sites. Indeed, this abstraction is based on the fact that the biochemical structure of species is described explicitly in Kappa, and thus we can extract directly from the interaction rules the sites whose state may influence the behavior (values) of the other sites. We summarize the flow

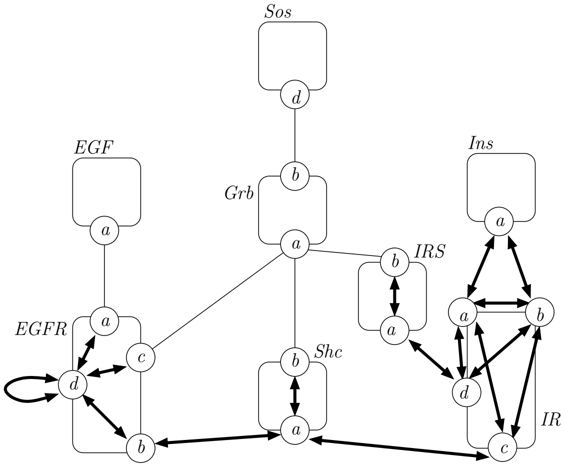


Figure 2: Approximation of the flow of information for the stochastic semantics.

of information by a binary (oriented) relation among the sites of the contact map. A formal description of the flow of information will be given in Sect. 5. Intuitively, a path between two sites indicates that the state of the site at the source of the path may have an influence of the behavior of the state of the site at the target of the path. This information can be used to cut species into fragments of species. Indeed, when two sites have an influence on the behavior of the state of a given site, then the correlation between the state of these two sites may have an influence on the behavior of this third site. Otherwise, this correlation can be safely abstracted away. Moreover, if the state of a given site influences the behavior of two sites, then the state of the latter two sites may be correlated, but it is not necessarily the case. Thus the flow of information can be used to detect invariants (the absence of correlation), and to detect useless information (when a corelation can be safely abstracted away).

The flow of information is a semantics notion. It may thus be different, when observing the differential, and when observing the stochastic semantics of a model. In Fig. 3, we give the contact map annotated with an over-approximation of the flow of information for the differential semantics. We notice that the flow of information is sparser in the differential semantics than in the stochastic semantics. We refer to [13] for a list of toy examples, illustrating the difference between the two notions of flow of information. Moreover, the approximation of the flow of information that we use, is a syntactic over approximation (which is extracted directly from the interaction rules in Kappa), and this over-approximation is qualitative, i.e. it does not take into account the values of the kinetic rates of the rules. Indeed, interaction rules in Kappa already encode how much an interaction depends on its context of application, from which we define our abstraction of the flow of information. This abstraction is sound for any given values of the kinetic rates.

In the remaining part of the paper, we formalize the concepts that were

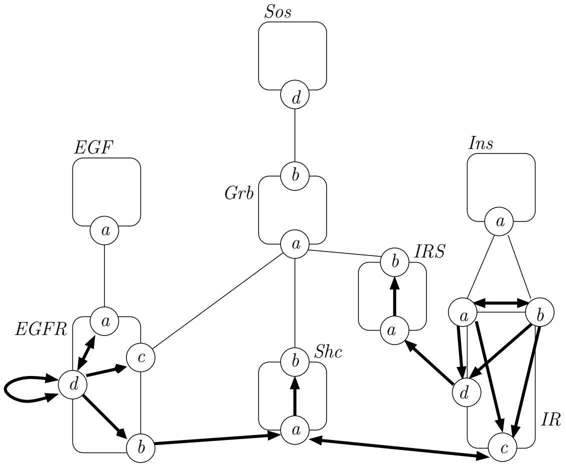


Figure 3: Approximation of the flow of information for the differential semantics.

sketched in this section, and we relate the used abstractions to the notions of lumpability and bisimulation.

3. Weighted labelled transition systems

We define the stochastic semantics of a biochemical network by a continuous-time time-homogeneous Markov chain (CTMC) on a countable state space. Our abstractions that we intend to do, are based on relationships between the potential transitions that update the state of the system. So as to describe explicitly these transitions, we use weighted labelled transition system (WLTSs) as a refinement of CTMCs. We will assign a WLTS to a given Kappa specification, and we manipulate that object when reasoning about abstractions.

3.1. CTMC and WLTS

We will observe the CTMC that is generated by a WLTS on a countable state space. We define the CTMC of a WLTS, by defining the Borel σ-algebra containing all cylinder sets of traces [26] that can occur in the system, and the corresponding probability distribution among them. We also introduce the standard notation of a rate matrix, which we will use when analyzing the lumpability and bisimulation properties in Sect. 6.

Definition 1. *(WLTS) A weighted-labelled transition system $\mathcal{W}$ is a tuple $(\mathcal{X}, \mathcal{L}, w, \pi_0)$, where*

- *$\mathcal{X}$ is a countable state space;*
- *$\mathcal{L}$ is a set of labels;*
- *$w : \mathcal{X} \times \mathcal{L} \times \mathcal{X} \to \mathbb{R}_0^+$ is a weighting function, it maps two states and a label to a real value;*

$$
\begin{aligned}
A &\xrightleftharpoons[k_2]{k_1} A^\star \\
AB &\xrightleftharpoons[k_2]{k_1} A^\star B \\
AB^\star &\xrightleftharpoons[k_2]{k_1} A^\star B^\star
\end{aligned}
$$

(a) (De)Phosphorylation of A.

$$
\begin{aligned}
B &\xrightleftharpoons[k_4]{k_3} B^\star \\
AB &\xrightleftharpoons[k_4]{k_3} AB^\star \\
A^\star B &\xrightleftharpoons[k_4]{k_3} A^\star B^\star
\end{aligned}
$$

(b) (De)Phosphorylation of B.

$$
\begin{aligned}
A + B &\xrightleftharpoons[k_6]{k_5} AB \\
A^\star + B &\xrightleftharpoons[k_6]{k_5} A^\star B \\
A + B^\star &\xrightleftharpoons[k_6]{k_5} AB^\star \\
A^\star + B^\star &\xrightleftharpoons[k_6]{k_5} A^\star B^\star
\end{aligned}
$$

(c) (Un)Binding.

Figure 4: An example of a set of chemical reactions, specified in a rule-based language. Two kinds of proteins, A and B, can change their internal state from being phosphorylated (denoted by a symbol $\star$) to unphosphorylated (no symbol), and back. This is depicted in columns (a) and (b); Moreover, a protein of type A, and a protein of type B may bind to form a complex AB, and the complex may be unbound again (depicted in column (c)).

- $\pi_0 : \mathcal{X} \to [0,1]$ *is an initial probability distribution.*

We assume that the label fully identifies the transition, i.e. for any $x \in \mathcal{X}$ *and* $l \in \mathcal{L}$*, there is at most one* $x' \in \mathcal{X}$*, such that* $w(x,l,x') > 0$*. Moreover, we assume that the system is finitely branching, in the sense that (i) the set* $\{x \in \mathcal{X} \mid \pi_0(x) > 0\}$ *is finite, and (ii) for arbitrary* $\hat{x} \in \mathcal{X}$*, the set* $\{(l,x') \in \mathcal{L} \times \mathcal{X} \mid w(\hat{x},l,x') > 0\}$ *is finite.*

The activity of the state x_i*, denoted* $a : \mathcal{X} \to \mathbb{R}_0^+$ *is the sum of all weights originating at* x_i*, i.e.*

$$a(x_i) := \sum \{w(x_i,l,x_j) \mid x_j \in \mathcal{X},\ l \in \mathcal{L}\}.$$

Example 3.1. *We do not describe extensionally the WLTS associated to the example of Sect. 2, because its combinatorial complexity is too high. Thus, we focus on a simpler example, that we will use a running example all along this section. We consider two kinds of proteins, A and B. Each protein can be unphosphorylated, or phosphorylated. Moreover, a protein A and a protein B may form a complex AB. We use the symbol* $\star$ *as a superscript when a protein is phosphorylated. This way, a fully phosphorylated complex is denoted by* $A^\star B^\star$*.*

The behavior of a chemical soup can be described by the twenty chemical reactions which are given in Fig. 4. Each reaction is made of a set of reactants, a set of products, and a rate constant, which denotes the likelihood that such a reaction happens. Our reactions are bidirectional. Moreover, we have assumed that all reactions are purely local. That is to say that the kinetic of phosphorylation and dephosphorylation of both the protein A (see first column) and the protein B (see second column) depends neither on the fact that the protein is in a complex, or not, nor (if it is in a complex) on the phosphorylation state of the other protein in the complex. Moreover, the kinetic of complex formation and dissociation does not depend on the phosphorylation state of the two proteins in a given complex (see third column).

$$\begin{cases}
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_1, (n_A - 1, n_{A^\star} + 1, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_1 n_A \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_2, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} - 1, n_{A^\star B} + 1, n_{AB^\star}, n_{A^\star B^\star})) = k_1 n_{AB} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_3, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star} - 1, n_{A^\star B^\star} + 1)) = k_1 n_{AB^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_4, (n_A + 1, n_{A^\star} - 1, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_2 n_{A^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_5, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} + 1, n_{A^\star B} - 1, n_{AB^\star}, n_{A^\star B^\star})) = k_2 n_{A^\star B} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_6, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star} + 1, n_{A^\star B^\star} - 1)) = k_2 n_{A^\star B^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_7, (n_A, n_{A^\star}, n_B - 1, n_{B^\star} + 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_3 n_B \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_8, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} - 1, n_{A^\star B}, n_{AB^\star} + 1, n_{A^\star B^\star})) = k_3 n_{AB} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_9, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B} - 1, n_{AB^\star}, n_{A^\star B^\star} + 1)) = k_3 n_{A^\star B} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{10}, (n_A, n_{A^\star}, n_B + 1, n_{B^\star} - 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_4 n_{B^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{11}, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} + 1, n_{A^\star B}, n_{AB^\star} - 1, n_{A^\star B^\star})) = k_4 n_{AB^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{12}, (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B} + 1, n_{AB^\star}, n_{A^\star B^\star} - 1)) = k_4 n_{A^\star B^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{13}, (n_A - 1, n_{A^\star}, n_B - 1, n_{B^\star}, n_{AB} + 1, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_5 n_A n_B \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{14}, (n_A, n_{A^\star} - 1, n_B - 1, n_{B^\star}, n_{AB}, n_{A^\star B} + 1, n_{AB^\star}, n_{A^\star B^\star})) = k_5 n_{A^\star} n_B \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{15}, (n_A - 1, n_{A^\star}, n_B, n_{B^\star} - 1, n_{AB}, n_{A^\star B}, n_{AB^\star} + 1, n_{A^\star B^\star})) = k_5 n_A n_{B^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{16}, (n_A, n_{A^\star} - 1, n_B, n_{B^\star} - 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star} + 1)) = k_5 n_{A^\star} n_{B^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{17}, (n_A + 1, n_{A^\star}, n_B + 1, n_{B^\star}, n_{AB} - 1, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_6 n_{AB} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{18}, (n_A, n_{A^\star} + 1, n_B + 1, n_{B^\star}, n_{AB}, n_{A^\star B} - 1, n_{AB^\star}, n_{A^\star B^\star})) = k_6 n_{A^\star B} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{19}, (n_A + 1, n_{A^\star}, n_B, n_{B^\star} + 1, n_{AB}, n_{A^\star B}, n_{AB^\star} - 1, n_{A^\star B^\star})) = k_6 n_{AB^\star} \\
w((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), r_{20}, (n_A, n_{A^\star} + 1, n_B, n_{B^\star} + 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star} - 1)) = k_6 n_{A^\star B^\star} \\
w(_, _, _) = 0
\end{cases}$$

Figure 5: Weight function for the system specified in Fig. 4. The state of the system, $x = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})$, is an 8-tuple of natural numbers, which encodes the number of instances of chemical species A, $A^\star$, B, $B^\star$, AB, $A^\star B$, $AB^\star$, $A^\star B^\star$. By applying the law of mass action, we obtain the weighting function for each of the reactions that can be applied to the state x.

We associate a WLTS to this system. The state of the system is a 8*-tuple of natural numbers, which encodes the number of instances of chemical species* A*,* $A^\star$*,* B*,* $B^\star$*,* AB*,* $A^\star B$*,* $AB^\star$*,* $A^\star B^\star$*. We apply the law of mass action [18] to obtain the weighting function of the WLTS, which is given in Fig. 5. The law of mass action stipulates that the likelihood that a given reaction happens is proportional to the product of the numbers of instances of the reactants and to the rate constant of the reaction. Last, we give the activity of the system. Given a state* $x = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})$*, we have:*

$$\begin{aligned}
a(x) = {} & k_1(n_A + n_{AB} + n_{AB^\star}) + k_2(n_{A^\star} + n_{A^\star B} + n_{A^\star B^\star}) \\
& + k_3(n_B + n_{AB} + n_{A^\star B}) + k_4(n_{B^\star} + n_{AB^\star} + n_{A^\star B^\star}) \\
& + k_5(n_A + n_{A^\star})(n_B + n_{B^\star}) + k_6(n_{AB} + n_{A^\star B} + n_{AB^\star} + n_{A^\star B^\star}).
\end{aligned}$$

□

The definition of a WLTS implicitly defines a *transition relation* $\rightarrow \subseteq \mathcal{X} \times \mathcal{X}$, such that $(x_i, x_j) \in \rightarrow$, if and only if there exists a non-zero transition from state x_i to state x_j, i.e. the total weight over all labels is strictly bigger than zero, written $\sum\{w(x_i, l, x_j) \mid l \in \mathcal{L}\} > 0$. Moreover, we can differentiate the *initial set of states* $\mathcal{I} \subseteq \mathcal{X}$, such that their initial probabilities are positive, i.e. $\mathcal{I} = \{x \in \mathcal{X} \mid \pi_0(x) > 0\}$.

Definition 2. *(Rate matrix of a WLTS) Given a WLTS* $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$*, we assign it the CTMC rate matrix* $R : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$*, given by* $R(x_i, x_j) = \sum\{w(x_i, l, x_j) \mid l \in \mathcal{L}\}$*.*

$$\left\{\begin{array}{l}
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A - 1, n_{A^\star} + 1, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_1 n_A \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} - 1, n_{A^\star B} + 1, n_{AB^\star}, n_{A^\star B^\star})) = k_1 n_{AB} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star} - 1, n_{A^\star B^\star} + 1)) = k_1 n_{AB^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A + 1, n_{A^\star} - 1, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_2 n_{A^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} + 1, n_{A^\star B} - 1, n_{AB^\star}, n_{A^\star B^\star})) = k_2 n_{A^\star B} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star} + 1, n_{A^\star B^\star} - 1)) = k_2 n_{A^\star B^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B - 1, n_{B^\star} + 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_3 n_B \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} - 1, n_{A^\star B}, n_{AB^\star} + 1, n_{A^\star B^\star})) = k_3 n_{AB} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B} - 1, n_{AB^\star}, n_{A^\star B^\star} + 1)) = k_3 n_{A^\star B} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B + 1, n_{B^\star} - 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_4 n_{B^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB} + 1, n_{A^\star B}, n_{AB^\star} - 1, n_{A^\star B^\star})) = k_4 n_{AB^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B} + 1, n_{AB^\star}, n_{A^\star B^\star} - 1)) = k_4 n_{A^\star B^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A - 1, n_{A^\star}, n_B - 1, n_{B^\star}, n_{AB} + 1, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_5 n_A n_B \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star} - 1, n_B - 1, n_{B^\star}, n_{AB}, n_{A^\star B} + 1, n_{AB^\star}, n_{A^\star B^\star})) = k_5 n_{A^\star} n_B \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A - 1, n_{A^\star}, n_B, n_{B^\star} - 1, n_{AB}, n_{A^\star B}, n_{AB^\star} + 1, n_{A^\star B^\star})) = k_5 n_A n_{B^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star} - 1, n_B, n_{B^\star} - 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star} + 1)) = k_5 n_{A^\star} n_{B^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A + 1, n_{A^\star}, n_B + 1, n_{B^\star}, n_{AB} - 1, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = k_6 n_{AB} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star} + 1, n_B + 1, n_{B^\star}, n_{AB}, n_{A^\star B} - 1, n_{AB^\star}, n_{A^\star B^\star})) = k_6 n_{A^\star B} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A + 1, n_{A^\star}, n_B, n_{B^\star} + 1, n_{AB}, n_{A^\star B}, n_{AB^\star} - 1, n_{A^\star B^\star})) = k_6 n_{AB^\star} \\
R((n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}), (n_A, n_{A^\star} + 1, n_B, n_{B^\star} + 1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star} - 1)) = k_6 n_{A^\star B^\star} \\
R(_, _) = 0
\end{array}\right.$$

Figure 6: Rate matrix for the system specified in Fig. 4.

The consequence is that we do not enforce $R(x_i, x_i) = -\sum\{R(x_i, x_j) \mid i \neq j\}$, as it is usual for the generator matrix of CTMCs. This however does not affect the transient, nor the steady-state behavior of the CTMC [1]. We do so for the following reason. When abstracting the WLTS by partitioning the state space, we get another WLTS. If the two states x and x' which have a transition between each other were aggregated in the same partition class $\tilde{x}$, it will result as a prolongation of the residence time in the abstract state $\tilde{x}$, i.e. we will have a self-loop in the abstract WLTS.

Example 3.2 (Ex. 3.1 continued)**.** *We give in the Fig. 6 the rate matrix of the WLTS. Since there exists no pair of distinct transitions between the same pair of states, the rate matrix is obtained directly by removing the transition labels.* □

Now we refer to the generated stochastic Markov process, which is written as a continuous-time random variable $\{X_t\}_{t \in \mathbb{R}_0^+}$, over the countable state space $\mathcal{X}$. We write $\Pr(X_t = x_i)$, the probability that the process takes the value x_i at time point t. It thus holds that $\Pr(X_0 = x_i) = \pi_0(x_i)$, and, when $\mathrm{d}t > 0$ converges toward 0,

$$\begin{aligned}
&\Pr(X_{t+\mathrm{d}t} = x_j \mid X_t = x_i) = R(x_i, x_j)\mathrm{d}t \text{ when } i \neq j, \\
&\Pr(X_{t+\mathrm{d}t} = x_i \mid X_t = x_i) = R(x_i, x_i)\mathrm{d}t + (1 - \sum\{R(x_i, x_{j'})\mathrm{d}t \mid x_{j'} \in \mathcal{X}\}).
\end{aligned}$$

The second equation can be simplified as follows:

$$\Pr(X_{t+\mathrm{d}t} = x_i \mid X_t = x_i) = 1 - \sum\{R(x_i, x_{j'})\mathrm{d}t \mid j' \neq i\}.$$

$$
\begin{aligned}
&\mathrm{P}r(X_{t+\mathrm{d}t} = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) = \\
&\quad k_1(n_A+1)\mathrm{P}r(X_t = (n_A+1, n_{A^\star}-1, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_1(n_{AB}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}+1, n_{A^\star B}-1, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_1(n_{AB^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}+1, n_{A^\star B^\star}-1)) \\
&\quad +k_2(n_{A^\star}+1)\mathrm{P}r(X_t = (n_A-1, n_{A^\star}+1, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_2(n_{A^\star B}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}-1, n_{A^\star B}+1, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_2(n_{A^\star B^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}-1, n_{A^\star B^\star}+1)) \\
&\quad +k_3(n_B+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B+1, n_{B^\star}-1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_3(n_{AB}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}+1, n_{A^\star B}, n_{AB^\star}-1, n_{A^\star B^\star})) \\
&\quad +k_3(n_{A^\star B}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}+1, n_{AB^\star}, n_{A^\star B^\star}-1)) \\
&\quad +k_4(n_{B^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B-1, n_{B^\star}+1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_4(n_{AB^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}-1, n_{A^\star B}, n_{AB^\star}+1, n_{A^\star B^\star})) \\
&\quad +k_4(n_{A^\star B^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}-1, n_{AB^\star}, n_{A^\star B^\star}+1)) \\
&\quad +k_5(n_A+1)(n_B+1)\mathrm{P}r(X_t = (n_A+1, n_{A^\star}, n_B+1, n_{B^\star}, n_{AB}-1, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_5(n_{A^\star}+1)(n_B+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}+1, n_B+1, n_{B^\star}, n_{AB}, n_{A^\star B}-1, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_5(n_A+1)(n_{B^\star}+1)\mathrm{P}r(X_t = (n_A+1, n_{A^\star}, n_B, n_{B^\star}+1, n_{AB}, n_{A^\star B}, n_{AB^\star}-1, n_{A^\star B^\star})) \\
&\quad +k_5(n_{A^\star}+1)(n_{B^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}+1, n_B, n_{B^\star}+1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}-1)) \\
&\quad +k_6(n_{AB}+1)\mathrm{P}r(X_t = (n_A-1, n_{A^\star}, n_B-1, n_{B^\star}, n_{AB}+1, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_6(n_{A^\star B}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}-1, n_B-1, n_{B^\star}, n_{AB}, n_{A^\star B}+1, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad +k_6(n_{AB^\star}+1)\mathrm{P}r(X_t = (n_A-1, n_{A^\star}, n_B, n_{B^\star}-1, n_{AB}, n_{A^\star B}, n_{AB^\star}+1, n_{A^\star B^\star})) \\
&\quad +k_6(n_{A^\star B^\star}+1)\mathrm{P}r(X_t = (n_A, n_{A^\star}-1, n_B, n_{B^\star}-1, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}+1)) \\
&\quad -k_1(n_A+n_{AB}+n_{AB^\star})\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad -k_2(n_{A^\star}+n_{A^\star B}+n_{A^\star B^\star})\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad -k_3(n_B+n_{AB}+n_{A^\star B})\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad -k_4(n_{B^\star}+n_{AB^\star}+n_{A^\star B^\star})\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad -k_5(n_A+n_{A^\star})(n_B+n_{B^\star})\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star})) \\
&\quad -k_6(n_{AB}+n_{A^\star B}+n_{AB^\star}+n_{A^\star B^\star})\mathrm{P}r(X_t = (n_A, n_{A^\star}, n_B, n_{B^\star}, n_{AB}, n_{A^\star B}, n_{AB^\star}, n_{A^\star B^\star}))
\end{aligned}
$$

Figure 7: Chemical master equation for the system specified in Fig. 4.

We notice that $\mathrm{P}r(X_{t+\mathrm{d}t} = x_j \mid X_t = x_i)$ is a well defined distribution of probability if for any $x_j \neq x_i$, $R(x_i, x_j) \neq 0 \implies \mathrm{d}t < \frac{1}{R(x_i,x_j)}$. Since we have assumed that our WLTSs are finitely branching, it is always possible to find a real number $\epsilon > 0$ such that this constraint is satisfied for any state x_j and any $\mathrm{d}t$ in the interval $(0, \epsilon)$.

Example 3.3 (Ex. 3.1 continued)**.** *The differential equation which relates the probability that the system is in a given state at time t is called the chemical master equation and is given in Fig. 7.*

□

Now we define the traces of the system. Each trace observes for a given execution the sequence of visited states, the labels that were assigned to the executed transitions, and the time points of when each transition happened.

Definition 3. *(A trace of a WLTS) Let us observe the WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ and its CTMC. Given a number k in $\mathbb{N}$, we define a trace of length k as $\tau \in (\mathcal{X} \times \mathcal{L} \times \mathbb{R}_0^+)^k \times \mathcal{X}$, written*

$$\tau = x_0 \overset{l_1, t_1}{\rightarrow} x_1 \ldots x_{k-1} \overset{l_k, t_1 + \ldots + t_k}{\rightarrow} x_k.$$

If the trace τ is such that (i) $\pi_0(x_0) > 0$, and (ii) for all i, $0 \leq i \leq k$, we have that $w(x_i, l_i, x_{i+1}) > 0$, then we say that τ belongs to the set of traces of $\mathcal{W}$, and we write it $\tau \in \mathcal{T}(\mathcal{W})$.

The 'time stamps' on each of the transitions denote intuitively the absolute time of the transition, from the moment when the system was started ($t = 0$). Yet, since the likelihood that a particular event occurs exactly at a given time, the probability of a given trace is always 0. We thus introduce the cylinder set of traces, where each event occurs within an interval of times.

Definition 4. *(Cylinder set of traces) If $\mathbb{IR}$ is the set of all nonempty intervals in $\mathbb{R}_0^+$, then a cylinder set of traces $\tau_{\mathbb{IR}}$ is an element in $(\mathcal{X} \times \mathcal{L} \times \mathbb{IR})^k \times \mathcal{X}$. A cylinder set of traces is denoted as follows:*

$$\tau_{\mathbb{IR}} = x_0 \overset{l_1,I_1}{\to} x_1 \ldots x_{k-1} \overset{l_k,I_k}{\to} x_k. \tag{1}$$

and it denotes the set of all traces $\tau = x_0 \overset{l_1,t_1}{\to} x_1 \ldots x_{k-1} \overset{l_k,t_1+\ldots+t_k}{\to} x_k$, such that $t_i \in I_i$, $1 \leq i \leq k$. If the cylinder of traces $\tau_{\mathbb{IR}}$ is such that $\pi_0(x_0) > 0$, and for all $i = 0, ..., k-1$, we have that $w(x_i, l_i, x_{i+1}) > 0$, then we say that $\tau_{\mathbb{IR}}$ belongs to the cylinder set of traces of $\mathcal{W}$, and we write $\tau_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\mathcal{W})$.

In the previous definition, each interval I_i gives a lower bound and an upper bound for the waiting time between the transition $x_{i-2} \overset{l_{i-1}}{\to} x_{i-1}$ (or the beginning of the trace whenever $i = 1$), and the transition $x_{i-1} \overset{l_i}{\to} x_i$.

Let $\Omega(\mathcal{T}_{\mathbb{IR}}(\mathcal{W}))$ be the smallest Borel σ-algebra that contains all the cylinder sets of traces in $\mathcal{T}_{\mathbb{IR}}(\mathcal{W})$ (i.e. the smallest set of sets of traces that contains the cylinder sets of traces, and that is closed upon countable unions and complementation [35]). We define a probability measure over $\Omega(\mathcal{T}_{\mathbb{IR}}(\mathcal{W}))$ as follows.

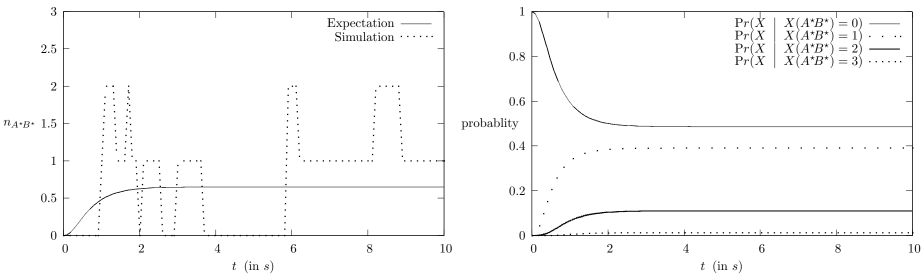
Definition 5. *(Trace density semantics on a WLTS) Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, and a number k in $\mathbb{N}$, the probability of the cylinder set of traces $\tau_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\mathcal{W})$, specified as in expression (1), is given by:*

$$\begin{array}{rcl} \pi(\tau_{\mathbb{IR}}) & = & \pi(x_0 \overset{l_1,I_1}{\to} x_1 \ldots x_{k-1} \overset{l_k,I_k}{\to} x_k) \\ & := & \pi_0(x_0) \prod_{i=1}^{k} \frac{w(x_{i-1},l_i,x_i)}{a(x_{i-1})} \left(e^{-a(x_{i-1}) \cdot \inf(I_i)} - e^{-a(x_{i-1}) \cdot \sup(I_i)} \right). \end{array}$$

Note that $\int_{I_i} a(x_{i-1}) e^{-a(x_{i-1}) \cdot t} \mathrm{d}t = e^{-a(x_{i-1}) \cdot \inf(I_i)} - e^{-a(x_{i-1}) \cdot \sup(I_i)}$ is the probability of exiting the state x_{i-1} in a time interval I_{i-1}, since the probability density function of the residence time of x_{i-1} is equal to $a(x_{i-1}) e^{-a(x_{i-1})}$.

Example 3.4 (Ex. 3.1 continued)**.** *We consider the following cylinder of traces. We start from the state $x_0 = (4, 0, 4, 0, 0, 0, 0, 0)$ that contains exactly four instances of the protein A and four instances of the protein B. We assume that the first reaction happens between the time $t_1 = 10^{-3}$ s and $t_2 = 0.1$ s and that this reaction binds a protein A and a protein B, thus, we get the state $x_1 = (3, 0, 3, 0, 1, 0, 0, 0, 0)$. Then, we assume that the next reaction is a phosphorylation of a free A, and that the duration between the two first reactions is between $t_3 = 0.01$ s and $t_4 = 0.1$ s.*

Thus, we obtain the following cylinder of traces:



(a)

(b)

Figure 8: Numerical simulations, with the rates $k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = 1s^{-1}$ and the initial state $n_A = n_B = 4$, $n_{A^\star} = n_{B^\star} = n_{AB} = n_{A^\star B} = n_{AB^\star} = n_{A^\star B^\star} = 0$. In Fig. 8(a), we plot both the expectation of the number of fully phosphorylated dimers $A^\star B^\star$ along the time, and its empiric value on a stochastic simulation. In Fig. 8(b), we plot the probability along the time that there is exactly p instances of $A^\star B^\star$, for $p = 0, 1, 2, 3$. For $p = 4$, the probability is always less than 0.0005 which is too low to be plotted.

$$\tau_{\mathbb{IR}} = x_1 \overset{r_1,[t_1,t_2]}{\rightarrow} x_2 \overset{r_7,[t_3,t_4]}{\rightarrow} x_3.$$

By definition, $\pi(\tau_{\mathbb{IR}})$ is equal to:

$$\pi_0(x_0)\frac{w(x_1,r_1,x_2)}{a(x_1)}(e^{-t_1 a(x_1)} - e^{-t_2 a(x_1)})\frac{w(x_2,r_7,x_3)}{a(x_2)}(e^{-t_3 a(x_2)} - e^{-t_4 a(x_2)}).$$

Thus, we get that:

$$\pi(\tau_{\mathbb{IR}}) = \pi_0(x_0)\frac{48k_1k_5}{(4(k_1+k_3)+16k_5)(4(k_1+k_3)+9k_5+k_{16})}\delta_1\delta_2,$$

where $\delta_1 = (e^{-(4(k_1+k_3)+16k_5)t_1} - e^{-(4(k_1+k_3)+16k_5)t_2})$,
and $\delta_2 = (e^{-(4(k_1+k_3)+9k_5+k_{16})t_3} - e^{-(4(k_1+k_3)+9k_5+k_{16})t_4})$.

Now we assume that $\pi_0(x_0) = 1$ and $k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = 1s^{-1}$. Under these assumptions, we get that:

$$\pi(\tau_{\mathbb{IR}}) = \frac{1}{9}(e^{-0.0024} - e^{-0.24})(e^{-0.018} - e^{-0.18}).$$

We notice that: $(e^{-0.0024} - e^{-0.24}) \approx 0.89$ and $(e^{-0.018} - e^{-0.18}) \approx 0.67$. Moreover, the density probability of the cylinder $\tau_{\mathbb{IR}}$ of traces is approximately equal to 0.066.

□

Example 3.5 (Ex. 3.1 continued)**.** *For a small initial population, we can make some numerical experimentations about the density distribution of the traces of our system. In Fig. 8(a), we plot the number of instances of fully phosphorylated dimers $A^\star B^\star$, along a stochastic simulation [9] of the model. We also plot* *the expectation of the number of $A^\star B^\star$ which we have obtained in solving the chemical master equation. In Fig. 8(b), we plot the probability that the system contains exactly respectively* 0, 1, 2, *and* 3 *instances of $A^\star B^\star$ along the time. The simulation has been performed by using the simulator* KaSim *[9, 32], while the other computations have been done by solving the chemical master equation in* Maple *[29]. Graphs have been generated by using* Gnuplot *[16].*

□

4. Kappa

We present Kappa in a process-like notation. We start with an operational semantics, then define the stochastic semantics of a Kappa model.

4.1. Syntax

We assume a finite set of agent names $\mathcal{A}$, representing different kinds of proteins; a finite set of sites $\mathcal{S}$, corresponding to protein domains; a finite set of internal states $\mathbb{I}$, and Σ_ι,Σ_λ two signature maps from $\mathcal{A}$ to $\wp(\mathcal{S})$[4], listing the domains of a protein which can bear respectively an internal state and a binding state. We denote by Σ the signature map that associates to each agent name $A \in \mathcal{A}$ the combined interface $\Sigma_\iota(A) \cup \Sigma_\lambda(A)$.

Example 4.1. *We will use a running example all along this section, so as to illustrate the different features of Kappa. In this running example, the set of agent names is given by $\mathcal{A} := \{A\}$, the set of sites is given by $\mathcal{S} := \{a, b\}$, the set of internal states is given by $\mathbb{I} := \{u, p\}$ (u stands for unphosphorylated, whereas p stands for phosphorylated). Moreover, the signature maps are defined by $\Sigma_\iota(A) := \{a, b\}$ and $\Sigma_\lambda(A) := \{a\}$. Thus, the site a can bear both a binding an an internal state, while the site b can only bear an internal state.*

□

Definition 6. *(Kappa agent) A Kappa agent $A(\sigma)$ is defined by its type $A \in \mathcal{A}$ and its interface σ. In $A(\sigma)$, the interface σ is a sequence of sites s in $\Sigma(A)$, with internal states (as subscript) and binding states (as superscript). The internal state of the site s may be written as s_ϵ, which means that either it does not have internal states (when $s \in \Sigma(A) \setminus \Sigma_\iota(A)$), or it is not specified. A site that bears an internal state $m \in \mathbb{I}$ is written s_m (in such a case $s \in \Sigma_\iota(A)$). The binding state of a site s can be specified as s^ϵ, if it is free, otherwise it is bound (which is possible only when $s \in \Sigma_\lambda(A)$). There are several levels of information about the binding partner: we use a binding label $i \in \mathbb{N}$ when we know the binding partner, or a wildcard bond $-$ when we only know that the site is bound. The detailed description of the syntax of a Kappa agent is given by the following*

[4] Given a set X, $\wp(X)$ denotes the power set of X (i.e. the set of all subsets of X).

grammar:

$$\begin{array}{rcll}
a & ::= & N(\sigma) & \textit{(agent)} \\
N & ::= & A \in \mathcal{A} & \textit{(agent name)} \\
\sigma & ::= & \varepsilon \mid s{,}\sigma & \textit{(interface)} \\
s & ::= & n_{\iota}^{\lambda} & \textit{(site)} \\
n & ::= & x \in \mathcal{S} & \textit{(site name)} \\
\iota & ::= & \epsilon \mid m \in \mathbb{I} & \textit{(internal state)} \\
\lambda & ::= & \epsilon \mid - \mid i \in \mathbb{N} & \textit{(binding state)}
\end{array}$$

We generally omit the symbol ϵ.

Definition 7. *(Kappa expression) Kappa expression E is a set of agents $A(\sigma)$ and fictitious agents $\emptyset$. Thus the syntax of a Kappa expression is defined as follows:*

$$E ::= \varepsilon \mid a \ , \ E \mid \emptyset \ , \ E.$$

Example 4.2 (Ex. 4.1 continued)**.** *The following Kappa expression*

$$A\big(a_u^1{,}b_p\big) \ , \ A\big(a_p^1{,}b_u\big)$$

denotes a soup of two agents A. In this expression, the first agent has the site a unphosphorylated, and the site b phosphorylated, whereas the second agent has the site a phosphorylated and the site b unphosphorylated. Moreover, the two agents are connected through their site a.

□

The structural equivalence $\equiv$, defined as the smallest binary equivalence relation between expressions that satisfies the rules given as follows

$$\begin{array}{rcl}
E \ , \ A(\sigma{,}s{,}s'{,}\sigma') \ , \ E' & \equiv & E \ , \ A(\sigma{,}s'{,}s{,}\sigma') \ , \ E' \\
E \ , \ a \ , \ a' \ , \ E' & \equiv & E \ , \ a' \ , \ a \ , \ E' \\
E & \equiv & E \ , \ \emptyset \\
\multicolumn{3}{c}{\dfrac{i, j \in \mathbb{N} \text{ and } i \text{ does not occur in } E}{E[i/j] \equiv E}} \\
\multicolumn{3}{c}{\dfrac{i \in \mathbb{N} \text{ and } i \text{ occurs only once in } E}{E[\epsilon/i] \equiv E}}
\end{array}$$

stipulates that neither the order of sites in interfaces nor the order of agents in expressions matters, that a fictitious agent might as well not be there, that binding labels can be injectively renamed and that *dangling bonds* can be removed.

Example 4.3 (Ex. 4.1 continued)**.** *Since neither the order of agents, nor the order of sites, nor the choice of binding labels matter, the following Kappa expression*

$$A\big(a_u^1{,}b_p\big) \ , \ A\big(a_p^1{,}b_u\big)$$

is $\equiv$-equivalent to the following one:

$$A\big(a_p^3{,}b_u\big) \ , \ A\big(b_p{,}a_u^3\big).$$

□

Definition 8. *(Kappa pattern, Kappa mixture) A Kappa pattern is a Kappa expression which satisfies the following five conditions: (i) no site name occurs more than once in a given interface; (ii) each site name s in the interface of the agent A occurs in $\Sigma(A)$; (iii) each site s which occurs in the interface of the agent A with a non empty internal state occurs in $\Sigma_\iota(A)$; (iv) each site s which occurs in the interface of the agent A with a non empty binding state occurs in $\Sigma_\lambda(A)$; and (v) each binding label $i \in \mathbb{N}$ occurs exactly twice if it does at all — there are no dangling bonds. A mixture is a pattern that is fully specified, i.e. each agent A documents its full interface $\Sigma(A)$, a site can only be free or tagged with a binding label $i \in \mathbb{N}$, a site in $\Sigma_\iota(A)$ bears an internal state in $\mathbb{I}$, and no fictitious agent occurs.*

Example 4.4 (Ex. 4.1 continued)**.** *We notice that the following expression*

$$A\big(a_u^1, b_p\big)\ ,\ A\big(a_p^1, b_u\big)$$

is indeed a mixture.

□

Definition 9. *(Kappa rule) A Kappa rule r is defined by two Kappa patterns E_ℓ and E_r, and a rate $k \in \mathbb{R}_0^+$, and is written: $r = E_\ell \to E_r @ k$.*

A rule r is well-defined, if the expression E_r is obtained from E_ℓ by finite application of the following operations: (i) creation (some fictitious agents $\emptyset$ are replaced with some fully defined agents of the form $A(\sigma)$, moreover σ documents all the sites occurring in $\Sigma(A)$ and all site in $\Sigma_\iota(A)$ bears an internal state in $\mathbb{I}$), (ii) unbinding (some occurrences of the wild card and binding labels are removed), (iii) deletion (some agents with only free sites are replaced with fictitious agent $\emptyset$), (iv) modification (some non-empty internal states are replaced with some non-empty internal states), (v) binding (some free sites are bound pair-wise by using binding labels in $\mathbb{N}$).

Example 4.5 (Ex. 4.1 continued)**.** *Now we introduce the following two rules:*

$$\begin{array}{rcll} A() & \to & \emptyset & @1 \\ A(a_u^-) & \to & A(a_u) & @1. \end{array}$$

The first rule deletes an agent A whatever the states of its sites are, whereas the second rule releases the binding stemming from the unphosphorylated site a of an agent A. The rate of both rules is 1*.*

□

From now on, we assume all rules to be well-defined. We sometimes omit the rate of a rule. Moreover, we denote by $E_\ell \leftrightarrow E_r @ k_1, k_2$ the two rules $E_\ell \to E_r @ k_1$ and $E_r \to E_\ell @ k_2$.

Definition 10. *(Kappa system) A Kappa system $\mathcal{R} = (\pi_0^{\mathcal{R}}, \{r_1, \ldots, r_n\})$ is given by finite distribution over initial mixtures $\pi_0^{\mathcal{R}} : \{M_{0_1}, \ldots, M_{0_k}\} \to [0, 1]$, and a finite set of rules $\{r_1, \ldots, r_n\}$.*

r01: $\mathit{EGF}(a)\ ,\ \mathit{EGFR}(a,d) \longleftrightarrow \mathit{EGF}(a^1)\ ,\ \mathit{EGFR}(a^1,d)$

r02: $\mathit{EGF}(a)\ ,\ \mathit{EGFR}(a,d^-) \longleftrightarrow \mathit{EGF}(a^1)\ ,\ \mathit{EGFR}(a^1,d^-)$

r03: $\mathit{EGFR}(a,d)\ ,\ \mathit{EGFR}(a^-,d) \longleftrightarrow \mathit{EGFR}(a,d^1)\ ,\ \mathit{EGFR}(a^-,d^1)$

r04: $\mathit{EGFR}(a,d)\ ,\ \mathit{EGFR}(a,d) \longleftrightarrow \mathit{EGFR}(a,d^1)\ ,\ \mathit{EGFR}(a,d^1)$

r05: $\mathit{EGFR}(a^-,d)\ ,\ \mathit{EGFR}(a^-,d) \longleftrightarrow \mathit{EGFR}(a^-,d^1)\ ,\ \mathit{EGFR}(a^-,d^1)$

r06: $\mathit{EGFR}(b_u,d) \longleftrightarrow \mathit{EGFR}(b_p,d)$

r07: $\mathit{EGFR}(b_u,d^-) \longleftrightarrow \mathit{EGFR}(b_p,d^-)$

r08: $\mathit{EGFR}(b_p)\ ,\ \mathit{Shc}(a) \longleftrightarrow \mathit{EGFR}(b_p^1)\ ,\ \mathit{Shc}(a^1)$

r09: $\mathit{EGFR}(b_p^1,d)\ ,\ \mathit{Shc}(a^1,b_u) \longleftrightarrow \mathit{EGFR}(b_p^1,d)\ ,\ \mathit{Shc}(a^1,b_p)$

r10: $\mathit{EGFR}(b_p^1,d^-)\ ,\ \mathit{Shc}(a^1,b_u) \longleftrightarrow \mathit{EGFR}(b_p^1,d^-)\ ,\ \mathit{Shc}(a^1,b_p)$

r11: $\mathit{Grb}(a)\ ,\ \mathit{Shc}(b_p) \longleftrightarrow \mathit{Grb}(a^1)\ ,\ \mathit{Shc}(b_p^1)$

r12: $\mathit{EGFR}(c_u,d) \longleftrightarrow \mathit{EGFR}(c_p,d)$

r13: $\mathit{EGFR}(c_u,d^-) \longleftrightarrow \mathit{EGFR}(c_p,d^-)$

r14: $\mathit{EGFR}(c_p,d)\ ,\ \mathit{Grb}(a) \longleftrightarrow \mathit{EGFR}(c_p^1,d)\ ,\ \mathit{Grb}(a^1)$

r15: $\mathit{EGFR}(c_p,d^-)\ ,\ \mathit{Grb}(a) \longleftrightarrow \mathit{EGFR}(c_p^1,d^-)\ ,\ \mathit{Grb}(a^1)$

r16: $\mathit{IR}(a,b)\ ,\ \mathit{Ins}(a) \longleftrightarrow \mathit{IR}(a^1,b)\ ,\ \mathit{Ins}(a^1)$

r17: $\mathit{IR}(a,b^-)\ ,\ \mathit{Ins}(a) \longleftrightarrow \mathit{IR}(a^1,b^-)\ ,\ \mathit{Ins}(a^1)$

r18: $\mathit{IR}(a,b)\ ,\ \mathit{Ins}(a) \longleftrightarrow \mathit{IR}(a,b^1)\ ,\ \mathit{Ins}(a^1)$

r19: $\mathit{IR}(a^-,b)\ ,\ \mathit{Ins}(a) \longleftrightarrow \mathit{IR}(a^-,b^1)\ ,\ \mathit{Ins}(a^1)$

r20: $\mathit{IR}(a,b,c_u) \longleftrightarrow \mathit{IR}(a,b,c_p)$

r21: $\mathit{IR}(a^-,b,c_u) \longleftrightarrow \mathit{IR}(a^-,b,c_p)$

r22: $\mathit{IR}(a,b^-,c_u) \longleftrightarrow \mathit{IR}(a,b^-,c_p)$

r23: $\mathit{IR}(a^-,b^-,c_u) \longleftrightarrow \mathit{IR}(a^-,b^-,c_p)$

r24: $\mathit{IR}(c_p)\ ,\ \mathit{Shc}(a) \longleftrightarrow \mathit{IR}(c_p^1)\ ,\ \mathit{Shc}(a^1)$

r25: $\mathit{IR}(a^-,b^-,c^1)\ ,\ \mathit{Shc}(a^1,b_u) \longleftrightarrow \mathit{IR}(a^-,b^-,c^1)\ ,\ \mathit{Shc}(a^1,b_p)$

r26: $\mathit{IR}(a,b,d_u) \longleftrightarrow \mathit{IR}(a,b,d_p)$

r27: $\mathit{IR}(a^-,b,d_u) \longleftrightarrow \mathit{IR}(a^-,b,d_p)$

r28: $\mathit{IR}(a,b^-,d_u) \longleftrightarrow \mathit{IR}(a,b^-,d_p)$

r29: $\mathit{IR}(a^-,b^-,d_u) \longleftrightarrow \mathit{IR}(a^-,b^-,d_p)$

r30: $\mathit{IR}(d_p)\ ,\ \mathit{IRS}(a) \longleftrightarrow \mathit{IR}(d_p^1)\ ,\ \mathit{IRS}(a^1)$

r31: $\mathit{IR}(a^-,b^-,d^1)\ ,\ \mathit{IRS}(a^1,b_u) \longleftrightarrow \mathit{IR}(a^-,b^-,d^1)\ ,\ \mathit{IRS}(a^1,b_p)$

r32: $\mathit{Grb}(a)\ ,\ \mathit{IRS}(b_p) \longleftrightarrow \mathit{Grb}(a^1)\ ,\ \mathit{IRS}(b_p^1)$

r33: $\mathit{Grb}(b)\ ,\ \mathit{Sos}(d_u) \longleftrightarrow \mathit{Grb}(b^1)\ ,\ \mathit{Sos}(d_u^1)$

r34: $\mathit{Grb}(b)\ ,\ \mathit{Sos}(d_p) \longleftrightarrow \mathit{Grb}(b^1)\ ,\ \mathit{Sos}(d_p^1)$

r35: $\mathit{Grb}(b^1)\ ,\ \mathit{Sos}(d_u^1) \longleftrightarrow \mathit{Grb}(b^1)\ ,\ \mathit{Sos}(d_p^1)$

r36: $\mathit{Sos}(d_u) \longleftrightarrow \mathit{Sos}(d_p)$

r37: $\mathit{Shc}(b_u) \longleftrightarrow \mathit{Shc}(b_p)$

r38: $\mathit{IRS}(b_u) \longleftrightarrow \mathit{IRS}(b_p)$

Table 1: Rule set for the *EGFR*/Insulin crosstalk. We omit the rate constants because the reduction procedure (introduced in Sect. 5) does not depend on the choice of rate constants.

Example 4.6 (case study (Sect. 2) continued)**.** *We give in Table 1 the set of rules for the model of crosstalk between the EGF receptor and the insulin receptor pathways. We leave the signature of the model implicit. Rules (*r01,r02*) describe the (un)binding between a ligand EGF and the site a of a receptor EGFR. We have used two rules to encode EGF/EGFR binding, in order to model the fact that the rate of association may depend on whether EGFR is in a dimer, or not. Rules (*r03,r04,r05*) describe dimer formation and dissociation. We notice that the rate of dimer formation/dissociation depends on the number of ligands that are bound to the receptors. Rules (*r06,r07*) describe the (de)phosphorylation of the site b of EGFR at a rate which depends on whether the receptor is in a dimer, or not. The rule (*r08*) describes the (un)binding between EGFR and Shc. Rules (*r09,r10*) describe the (de)phosphorylation of Shc by EGFR (the rate depends on the fact whether the receptor is still in a dimer, or not). Rule (*r11*) describes the recruitment of a transport molecule Grb by Shc. Rules (*r12,r13*) describe the (de)phosphorylation of the site c of EGFR and rules (*r14,r15*) describe the recruitment of Grb directly by EGFR at rates which depends on whether or not EGFR is in a dimer. Rules (*r16,r17,r18,r19*) describe the (un)binding between an insulin receptor (IR) and insulin molecule (the rate may depend on the fact whether an insulin molecule has already been recruited). Rules (*r20,r21,r22,r23*) describe the (un)phosphorylation of the site c of the IR at a rate which depends on the number of recruited insulin molecules (in practice the rates of rules* r21 *and* r22 *are the same). Rule (*r24*) describes the (un)binding between IR and Shc. Rule (*r25*) describes the (un)phosphorylation of Shc by IR. Rules (*r26,r27,r28,r29*) describe the (un)phosphorylation of the site d of IR at a rate which depends on the number of recruited insulin molecules. Rule (*r30*) describes the (un)binding between IR and IRS. Rule (*r31*) describes the (un)phosphorylation of IRS. Rule (*r32*) describes the (un)binding between IRS and Grb. Rules (*r33,r34*) describe the (un)binding between Grb and Sos at a rate which depends on whether or not Sos is phosphorylated. Rules (*r35,r36*) describe the (un)phosphorylation of Sos at a rate which depends on whether or not it is bound to Grb. Rule (*r37*) describes the spontaneous dephosphorylation of Shc and rule (*r28*) the spontaneous dephosphorylation of IRS.*

□

4.2. Operational semantics

In order to apply a rule $r := E_\ell \to E_r@k$ to a mixture M, we use the structural equivalence $\equiv$ to bring the participating agents to the front of E (with their sites in the same order as in E_ℓ), rename binding labels if necessary and introduce a fictitious agent for each agent that is created by r. This yields an equivalent expression E' that *matches* the left hand side (lhs) E_ℓ, which is written $E \models E_\ell$, defined as follows:

$$\begin{gathered}
E \models \varepsilon \\
a \models a_\ell \wedge E \models E_\ell \implies (a\ ,\ E) \models (a_\ell\ ,\ E_\ell) \\
\emptyset \models \emptyset \\
\sigma \models \sigma_\ell \implies N(\sigma) \models N(\sigma_\ell) \\
\sigma \models \varepsilon \\
s \models s_\ell \wedge \sigma \models \sigma_\ell \implies (s, \sigma) \models (s_\ell, \sigma_\ell) \\
\iota \models \iota_\ell \wedge \lambda \models \lambda_\ell \implies n_\iota^\lambda \models n_{\iota_\ell}^{\lambda_\ell} \\
\iota_\ell \in \{\epsilon, \iota\} \implies \iota \models \iota_\ell \\
\lambda = \lambda_\ell \vee [\lambda \neq \epsilon \wedge \lambda_\ell = -] \implies \lambda \models \lambda_\ell
\end{gathered}$$

Note that in order to find a matching, we only use structural equivalence on E, not E_ℓ. We then replace E' by $E'[E_r]$ which is defined as follows:

$$\begin{gathered}
E[\varepsilon] = E \\
(a\ ,\ E)[a_r\ ,\ E_r] = a[a_r]\ ,\ E[E_r] \\
\emptyset[a_r] = a_r \\
a_r[\emptyset] = \emptyset \\
N(\sigma)[N(\sigma_r)] = N(\sigma[\sigma_r]) \\
\sigma[\varepsilon] = \sigma \\
(s, \sigma)[s_r, \sigma_r] = s[s_r], \sigma[\sigma_r] \\
\lambda[-] = \lambda \\
n_\iota^\lambda[n_{\iota_r}^{\lambda_r}] = n_{\iota[\iota_r]}^{\lambda[\lambda_r]} \\
\iota_r \in \mathbb{I} \implies \iota[\iota_r] = \iota_r \\
\lambda_r \in \mathbb{N} \cup \{\epsilon\} \implies \lambda[\lambda_r] = \lambda_r
\end{gathered}$$

This may produce dangling bonds (if r unbinds a wildcard bond or destroys an agent on one side of a bond) or fictitious agents (if r destroys agents), so we use $\equiv$ to resolve them.

Definition 11 (rule application)**.** *Given a Kappa rule $r = E_\ell \to E_r$ and a Kappa mixture E. We assume that E is $\equiv$-equivalent to a Kappa expression E' such that $E' \models E_\ell$. Then, the Kappa expression $E'[E_r]$ is well-defined and $\equiv$-equivalent to some mixtures. Let E'' be a Kappa mixture which is $\equiv$-equivalent to $E'[E_r]$. The Kappa mixture E'' is called the potential result of an application of r with E, which is denoted as follows:*

$$E \to_r E''.$$

Example 4.7 (Ex. 4.1 continued)**.** *The rule*

$$r_1 := A() \to \emptyset$$

can be applied with the mixture

$$A\big(a_u^1, b_p\big)\ ,\ A\big(a_p^1, b_u\big)$$

in two different ways.

1. *We have:*
$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big) \models A().$$
Thus, we can apply r_1 *with the agent* $A\big(a_u^1,b_p\big)$ *and replace this agent with the fictitious agent* $\emptyset$ *(as stated in the rhs of* r_1*), as done in the following computation:*
$$\big(A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big)\big)[\emptyset] = \emptyset\ ,\ A\big(a_p^1,b_u\big).$$
Yet, the Kappa expression
$$\emptyset\ ,\ A\big(a_p^1,b_u\big)$$
is not a Kappa mixture, but it is $\equiv$*-equivalent to the following Kappa mixture*
$$A\big(a_p,b_u\big).$$
(We notice that we have cleaned the dangling bond 1*.) Thus we get:*
$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big) \rightarrow_{r_1} A\big(a_p,b_u\big).$$

2. *But, we also have:*
$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big) \equiv A\big(a_p^1,b_u\big)\ ,\ A\big(a_u^1,b_p\big).$$
and:
$$A\big(a_p^1,b_u\big)\ ,\ A\big(a_u^1,b_p\big) \models A().$$
Thus, we can also apply r_1 *with the agent* $A\big(a_p^1,b_u\big)$ *and replace this agent with the fictitious agent* $\emptyset$ *(as stated in the rhs of* r_1*), as done in the following computation:*
$$\big(A\big(a_p^1,b_u\big)\ ,\ A\big(a_u^1,b_p\big)\big)[\emptyset] = \emptyset\ ,\ A\big(a_u^1,b_p\big).$$
Yet, the Kappa expression
$$\emptyset\ ,\ A\big(a_u^1,b_p\big)$$
is not a Kappa mixture, but it is $\equiv$*-equivalent to the following Kappa mixture*
$$A\big(a_u,b_p\big).$$
Thus we get:
$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big) \rightarrow_{r_1} A\big(a_u,b_p\big).$$

□

Example 4.8 (Ex. 4.1 continued)**.** *The rule*
$$r_2 := A\big(a_u^-\big) \rightarrow A\big(a_u\big)$$
can be applied with the mixture
$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big)$$
only by aligning the first agent of the lhs of the rule to the first agent of the mixture.

1. *We have:*

$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big) \models A\big(a_u^-\big)$$

and

$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big)[A(a_u)] = A\big(a_u,b_p\big)\ ,\ A\big(a_p^1,b_u\big).$$

Yet, the Kappa expression

$$A\big(a_u,b_p\big)\ ,\ A\big(a_p^1,b_u\big)$$

is not a Kappa mixture, but it is $\equiv$*-equivalent to the following Kappa mixture*

$$A\big(a_u,b_p\big)\ ,\ A\big(a_p,b_u\big)$$

(We notice that we have cleaned the dangling bond 1 *this way.) Thus we get:*

$$A\big(a_u^1,b_p\big)\ ,\ A\big(a_p^1,b_u\big) \rightarrow_{r_2} A\big(a_u,b_p\big)\ ,\ A\big(a_p,b_u\big).$$

□

4.3. Population-based stochastic semantics

In addition to the rate constants k, careful counting of the number of times each rule can be applied to a mixture is required to define the system's quantitative semantics correctly [8, 31]. Thus we define the notions of embedding between patterns. Let $Z = a_1\ ,\ \dots\ ,\ a_m$ and $Z_\ell = c_1,\dots,c_n$ be two patterns with no occurrence of the fictitious agent and such that there exists a pattern $Z' = b_1,\dots,b_m$ that satisfies both $Z \equiv Z'$ and $Z' \models Z_\ell$ (and so, in particular, $n \leq m$). The agent permutations used in the proof that $Z \equiv Z'$ allow us to derive a permutation $\mathtt{p}$ such that $a_{\mathtt{p}(i)} \equiv b_i$. The restriction ϕ of $\mathtt{p}$ to the integers between 1 and n is called an *embedding* between Z_ℓ and Z. This is written $Z_\ell \lhd_\phi Z$. There may be several embeddings between Z_ℓ and Z for the same Z'; if so, this influences the relative weight of the reaction in the stochastic semantics. We denote by $[Z, Z']$ the set of embeddings between Z and Z'. This notion of embedding is extended to patterns (including fictitious agents) by defining $Z_\ell \lhd_\phi Z$ if, and only if, $(\downarrow_\emptyset Z_\ell) \lhd_\phi (\downarrow_\emptyset Z)$, where $\downarrow_\emptyset$ removes all occurrences of the fictitious agent in patterns.

We assume that E_ℓ is the lhs of a rule $r := E_\ell \rightarrow E_r@k$ and Z is a mixture such that $E_\ell \lhd_\phi Z$. Let $Z = a_1,\dots,a_m$ and $\downarrow_\emptyset E_\ell = c_1,\dots c_n$. Given $Z' \equiv Z$ (we write $\downarrow_\emptyset Z' = b_1,\dots,b_m$) and a bijection $\mathtt{p}$ such that we have $Z' \models E_\ell$, $b_i \equiv a_{\mathtt{p}(i)}$ for $1 \leq i \leq m$ and $\phi(j) = \mathtt{p}(j)$ for $1 \leq j \leq n$. The result of applying r along ϕ to the mixture Z is defined (modulo $\equiv$) as any mixture that is $\equiv$-equivalent to $Z'[E_r]$. In other words the embedding ϕ between E_ℓ and Z fully defines the action of r on Z up to structural equivalence.

We are now ready to define the stochastic semantics by the mean of a WLTS. In this semantics, the state is a soup of agents, that is to say that we do not care about the order of agents in mixture. So the states of the system are the class of $\equiv$-mixture.

Defining species as connected mixture, the state of the system can be seen as a multi-set of species. The formal definition of a Kappa species is as follows:

Definition 12. *(Kappa species) A pattern E is reducible whenever $E \equiv E', E''$ for some non-empty patterns E', E''; A Kappa species is the $\equiv$-equivalence class of an irreducible Kappa mixture.*

Example 4.9 (Ex. 4.1 continued)**.** *The Kappa species are the following:*

$$[A(a_{x_1},b_{x_2})]_{\equiv}$$

for any $x_1, x_2 \in \{u, p\}$ and

$$[A(a^1_{x_1},b^1_{x_2})\ ,\ A(a^1_{x_3},b^1_{x_4})]_{\equiv},$$

for any $x_1, x_2, x_3, x_4 \in \{u, p\}$. Thus there are $4 + \frac{16-4}{2} + 4 = 14$ Kappa species.

□

As explained earlier, the action of a rule r on a mixture E is fully defined (up to $\equiv$) by an embedding ϕ between the lhs E_ℓ of the rule r and the mixture. So as to consider computation steps over $\equiv$-equivalent of mixtures, we introduce an equivalence relation $\equiv_{\mathcal{L}}$ over triples (r, E, ϕ) where ϕ is an embedding of the lhs E_ℓ of r into E. We say that $(r_1, E_1, \phi_1) \equiv_{\mathcal{L}} (r_2, E_2, \phi_2)$ if, and only if, (i) $r_1 = r_2$ and (ii) there exists an embedding $\psi \in [E_1, E_2]$ such that $\phi_2 = \psi \circ \phi_1$.

Definition 13. *(WLTS of a Kappa system) Let $\mathcal{R} = (\pi_0^{\mathcal{R}}, \{r_1, \dots, r_n\})$ be a Kappa system. We define the WLTS $\mathcal{W}_{\mathcal{R}} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ where:*

1. *$\mathcal{X}$ is the set of all $\equiv$-equivalent classes of mixtures;*
2. *$\mathcal{L}$ is the set of all $\equiv_{\mathcal{L}}$-equivalence classes of triples (r, E, ϕ) such that ϕ is an embedding between the lhs E_ℓ of r and E;*
3. *$w(x, l, x') = \frac{k}{|[E_\ell, E_\ell]|}$ whenever there exist a rule $r = E_\ell \rightarrow E_r@k$, two mixtures E and E', and an embedding $\phi \in [E_\ell, E]$, such that $x = [E]_{\equiv}$, $l = [r, E, \phi]_{\equiv_{\mathcal{L}}}$, $x' = [E']_{\equiv}$, and E' is the result (up to $\equiv$) of the application of r along ϕ to the mixture E; otherwise $w(x, l, x') = 0$;*
4. *$\pi_0(x) = \sum\{\pi_0^{\mathcal{R}}(E') \mid E' \in Dom(\pi_0^{\mathcal{R}}) \cap x\}$.*

The stochastic semantics of a Kappa system $\mathcal{R}$ is then defined as the trace distribution semantics of the WLTS $\mathcal{W}_{\mathcal{R}}$.

Example 4.10 (Ex. 4.1 continued)**.** *We now give an example. Consider the mixture E which is defined as follows:*

$$E := A(a_u,b_u)\ ,\ A(a_u,b_u)\ ,\ A(a_u,b_p),$$

and the following rule:

$$A(a_u) \rightarrow A(a_p)@k.$$

The rule r can be applied on $[E]_{\equiv}$ in three ways, which gives three distinct labels:

1. *Taking l_1 as the $\equiv_{\mathcal{L}}$-equivalent class of $(r, E, [1 \mapsto 1])$, we get:*

$$[E]_{\equiv} \xrightarrow{l_1} [A(a_p,b_u)\ ,\ A(a_u,b_u)\ ,\ A(a_u,b_p)]_{\equiv}.$$

2. *Taking* l_2 *as the* $\equiv_{\mathcal{L}}$*-equivalent class of* $(r, E, [1 \mapsto 2])$*, we get:*

$$[E]_{\equiv} \xrightarrow{l_2} [A(a_u, b_u)\ ,\ A(a_p, b_u)\ ,\ A(a_u, b_p)]_{\equiv}.$$

3. *Taking* l_3 *as the* $\equiv_{\mathcal{L}}$*-equivalent class of* $(r, E, [1 \mapsto 3])$*, we get:*

$$[E]_{\equiv} \xrightarrow{l_3} [A(a_u, b_u)\ ,\ A(a_u, b_u)\ ,\ A(a_p, b_p)]_{\equiv}.$$

One notices that:

$$A(a_p, b_u)\ ,\ A(a_u, b_u)\ ,\ A(a_u, b_p) \equiv A(a_u, b_u)\ ,\ A(a_p, b_u)\ ,\ A(a_u, b_p),$$

thus the first two transitions give the same result. Using distinct labels for the transitions allows counting precisely the number of embeddings between the lhs of a rule and a mixture, which is crucial for defining sound quantitive semantics.

□

5. Reduction procedure

In this section, we describe an approximation of the flow of information between the different regions of molecular species. Then we use it to define stochastic fragments. This framework is a simplification of the one which is described in [15]. To make the things easier, we assume, without any loss of generality that, in this section, Σ_ι and Σ_λ are disjoint sets. This can always be achieved by taking two disjoint copies $\mathcal{S}_\iota$ and $\mathcal{S}_\lambda$ of $\mathcal{S}$ and using site names in $\mathcal{S}_\iota$ to bear internal states, and site names in $\mathcal{S}_\lambda$ to bear binding states.

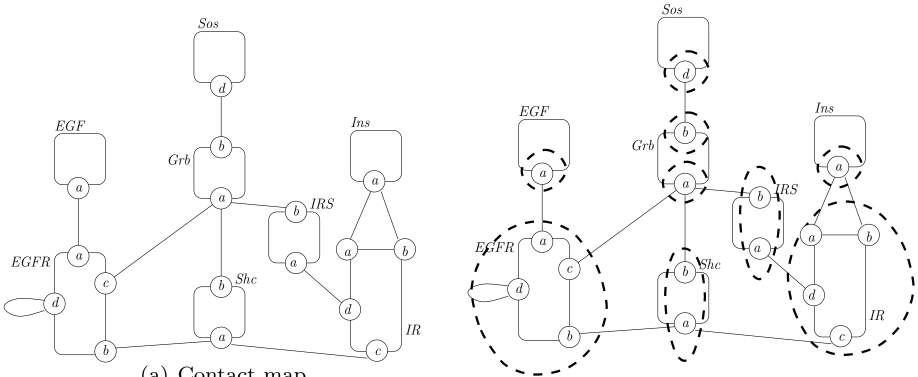
We introduce *contact maps* which summarize the agents of a model and their potential bindings. More formally, a contact map is a graph, the nodes of which itemize the different types of agents of the model. Each node is documented by the set of sites in the interface of the agent. Then, an edge between the sites of two or of the same protein(s) denotes a potential binding between the sites of two instances of this/these protein(s).

Definition 14. *(Contact map) Given a Kappa system* $\mathcal{R}$*, a contact map (CM) is a graph object* $(\mathcal{N}, \mathcal{E})$*, where the set of nodes* $\mathcal{N}$ *are agent types equipped with the corresponding interface, and the edges are specified between the sites of the nodes.*

Formally, we have that:

$$\mathcal{N} := \{(A, \Sigma(A)) \mid A \in \mathcal{A}\},$$
$$\mathcal{E} \subseteq \{((A, s), (A', s')) \mid A, A' \in \mathcal{A} \text{ and } s \in \Sigma(A), s' \in \Sigma(A')\},$$

and there is an edge between (A, s) *and* (A', s') *(i.e.* $((A, s), (A', s')) \in \mathcal{E}$*) if and only if the site* s *of an agent of type* A *and the site* s' *of an agent of type* A' *bear the same binding label in the rhs* E_r *of a rule.*



(a) Contact map

(b) Annotated contact map

Figure 9: Maps for the *EGFR*/Insulin crosstalk. (a) A contact map summarizes the potential bindings between proteins (b) The contact map is annotated by partitioning the set of sites: two sites are in the same annotation class, if their values are correlated by the flow of information in the stochastic dynamics of the model. For example, the sites *a* and *b* of protein *Grb* not being in the same annotation class means that the values of these two sites are modified independently of each other.

Example 5.1 (case study (Sect. 2) continued)**.** *We give again, in Fig. 9(a), the contact map for the EGF/Insulin crosstalk. We can notice that some sites are in competition (or in concurrency), since they can be connected to different kind of sites: this is the case with the site a of Grb for instance. Moreover, a site in a contact map can be connected with itself (which encodes the fact that the sites of two instances of the same agent can be connected together), as the site d of EGFR for instance.*

An *annotated contact map* is obtained by annotating a contact map with information about the flow of information between the states of the different sites. In the stochastic semantics, it turns out that our over-approximation of the flow of information is symmetric, that is to say that whenever we detect a potential flow from a site to another one, then we also detect a potential flow in the converse direction. Moreover, the flow of information is transitive. Thus, we can describe the flow of information between the same sites of an agent thanks to a binary equivalence relation. Moreover, there is no need to describe explicitly the flow of information across bindings, because in our approximation, whenever two sites can be bound together, there is a potential flow of information from one to the other, as soon as none of their equivalence class is a singleton.

Now we can formally define the annotated contact map.

Definition 15. *(Annotated Contact map) Given a Kappa system $\mathcal{R}$, a valid annotated contact map (ACM) is a contact map where all agents are annotated with respect to the rule set $\mathcal{R}$. The annotation on the agent of type $A \in \mathcal{A}$ is given by an equivalence relation on its set of sites $\approx_A \subseteq \Sigma(A) \times \Sigma(A)$ such that:*

- *If a rule r tests*[5] *the sites s_1 and site s_2 of agents a_1,a_2 (it is possible that $a_1 = a_2$) of type A, then $s_1 \approx_A s_2$;*
- *If a rule r creates an agent a of type A, then all the sites of $\Sigma(A)$ are in the same equivalence class, i.e. $\approx_A = \Sigma(A) \times \Sigma(A)$;*

Note that there can be several annotations of the agent type $A \in \mathcal{A}$ which satisfy the conditions. More precisely, if the equivalence relation $\approx_A$ meets the conditions, then any coarser equivalence relation satisfies them as well. This allows to define the smallest such equivalence relation $\approx_A$ which we call the minimal annotation of agent A. An ACM is minimal whenever each agent type is annotated by its minimal annotation.

Example 5.2 (case study (Sect. 2 continued))**.** *We give in Fig. 9(b) an annotated contact map which is compatible with the rules in Table 1. We notice that the hypothesis that for any agent $A \in \mathcal{A}, \Sigma_\iota(A) \cap \Sigma_\lambda(A) = \emptyset$, is not satisfied. Indeed, we assume implicitly, that whenever a site s belongs to $\Sigma_\iota(A) \cap \Sigma_\lambda(A)$, for a given agent type $A \in \mathcal{A}$, then there is a flow of information between the internal state and the binding state (and conversely) of the instance of the site s in A. Unlike in Fig. 2, we have described the approximation of the information flow thanks to equivalence classes, and not with oriented edges. One can get back the annotation of Fig. 2, by putting an oriented edge between (i) any two sites which belong to the same equivalence class, and between (ii) any two sites that can be bound together and such that the equivalence class of none of them is a singleton.*

Now we justify the annotation of the contact map. Sites a and d of EGFR are both tested in the rules r01*,* r02*,* r03*,* r04*, and* r05 *(one rule would have been enough), so they should belong to the same $\approx_{EGFR}$-equivalence class. The sites b and d of EGFR are both tested in the rules* r06*,* r07*,* r09*, and* r10*, so we should have b $\approx_{EGFR}$ d. The sites c and d of EGFR are both tested in the rules* r12*,* r13*,* r14*, and* r15*, so we should have c $\approx_{EGFR}$ d. The sites a and b of Shc are both tested in the rules* r09*,* r10*,* r25*. So we should have a $\approx_{Shc}$ b. The sites a and b of IR are both tested in the rules* r16*,* r17*,* r18*,* r19*, so we should have a $\approx_{IR}$ b. Moreover, the sites a, b, and c of IR are all tested in the rules* r20*,* r21*,* r22*,* r23*,* r25*, so they should belong to the same $\approx_{IR}$-equivalence class. And the sites a, b, and d of IR are all tested in the rules* r26*,* r27*,* r28*,* r29*,* r31*, so they should belong to the same $\approx_{IR}$-equivalence class. The sites a and b of IRS are both tested in the rule* r31*, thus we should have a $\approx_{IRS}$ b.*

Importantly, we notice that the sites a and b of Grb occur in no rule together. Thus, they can be in two distinct $\approx_{Grb}$-equivalence class. Thanks to this, our reduction procedure will simplify the system, by cutting each instance of Grb into two parts.

□

Let r be a rule and let us consider an ACM which is valid with respect

[5] We say that the site s of the agent a is *tested* by the rule r, if it occurs in the lhs of r.

to the singleton $\{r\}$. For any agent type $A \in \mathcal{A}$, either A does not occur in the lhs of r, or A occurs but all occurrences of A have an empty interface, or A occurs, tests some sites which are all $\approx_A$-equivalent. In the latter case, we define $\mathbf{test}_r^{ACM}(A) = C$ where C is the $\approx_A$-equivalent class of the sites, otherwise, we define $\mathbf{test}_r^{ACM}(A) = \emptyset$.

The meaning of the ACM is to summarize the dependences between sites that can occur during the simulation of a Kappa system. If the two sites s and s' in the $\Sigma(A)$ are related by the relation $\approx_A$, i.e. $s \approx_A s'$, it suggests that they are *dependent* in the following way. We must not aggregate in the same equivalence class any two states x and x', such that they contain the agent A in a different evaluation of the sites s and s'. On the other hand, if the two sites s and s' are not related by $\approx_A$, then we may aggregate the states by the 'marginal' criteria, i.e. the condition which involves only one of the sites. Therefore, the less states are related by $(\approx_A)_{A\in\mathcal{A}}$, the better the reduction will be. To numerically justify this, we can imagine having an agent of type A whose interface has n different sites $s_1, ..., s_n$, and each of them has two possible internal state modifications. Let us observe the two limiting relations $\approx_A$, i.e. $\approx_A = \{(s_i, s_j) \mid 1 \leq i \leq n, 1 \leq j \leq n\}$, and $\approx'_A = \{(s_i, s_i) \mid 1 \leq i \leq n\}$. The annotation $\approx_A$ enforces at least to 2^n states to describe all modifications of the agent A, whereas the annotation $\approx'_A$ suggests that it is enough to use only $2 \cdot n$ of them.

The ACM can be used to identify parts of Kappa species that we call fragments.

Definition 16. *(Kappa fragments) A fragment is the $\equiv$-equivalent class of a non empty irreducible pattern E such that: (i) the set of sites in the interface σ of an agent $A(\sigma)$ in E is an equivalence class of $\approx_A$, (ii) sites can only be free or tagged with a binding label $i \in \mathbb{N}$ and sites in Σ_ι are tagged with an internal state in $\mathbb{I}$, (iii) there is no occurrence of fictitious agent $\emptyset$.*

Example 5.3 (case study (Sect. 2 continued))**.** *Since, we do not have $a \approx_{Grb} b$, each instance of a Grb is cut into two parts. For instance the molecular species:*

$$EGFR\big(a,b_u,c_p^1,d^2\big)\ ,\ EGFR\big(a,b_p,c_p^3,d^2\big)\ ,\ Grb\big(a^1,b\big)\ ,\ Grb\big(a^3,b^4\big)\ ,\ Sos\big(d_u^4\big)$$

is cut into three fragments:

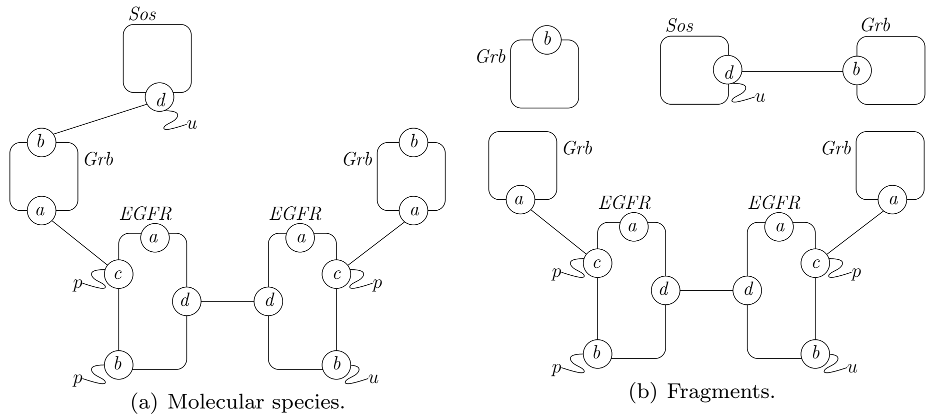
1. $EGFR\big(a,b_u,c_p^1,d^2\big)\ ,\ EGFR\big(a,b_p,c_p^3,d_u^2\big)\ ,\ Grb\big(a^1\big)\ ,\ Grb\big(a^3\big)$;
2. $Grb\big(b\big)$;
3. $Grb\big(b^1\big)\ ,\ Sos\big(d_u^1\big)$;

as shown in Fig. 10.

In this case study, 2, 768 *different molecular species may occur at run time. Since in the reduced model, each instance of a protein Grb can be safely cut into two parts, we get only* 609 *stochastic fragments.*

□

We can use fragments to abstract the WLTS $\mathcal{W}_{\mathcal{R}}$, by identifying the mixtures which have the same (multi-)set of fragments. To reach that goal, we first



(a) Molecular species.

(b) Fragments.

Figure 10: A fragmentation. A chemical species is cut into three fragments, by splitting each *Grb* into two parts and abstracting away which parts belong to the same protein.

overload the definition of $\equiv$ in order to identify mixtures having the same fragments. We introduce the binary relation $\equiv^\sharp$ as the smallest equivalence relation over patterns which is compatible with $\equiv$ and such that:

$$(A(\sigma)\ ,\ A(\sigma')\ ,\ E) \equiv^\sharp (A(\upharpoonright_C \sigma', \upharpoonright_{\Sigma(A)\setminus C} \sigma)\ ,\ A(\upharpoonright_C \sigma, \upharpoonright_{\Sigma(A)\setminus C} \sigma')\ ,\ E)$$

for any agent type $A \in \mathcal{A}$, σ,σ' interfaces, E pattern, and C an $\approx_A$-equivalence class of sites. For any set of sites $X \subseteq \mathcal{S}$, the projection function $\upharpoonright_X$ over interfaces keeps only the sites in X, formally $\upharpoonright_X$ is defined by $\upharpoonright_X \varepsilon = \varepsilon$, $\upharpoonright_X (s_\iota^\lambda,\sigma') = (s_\iota^\lambda, \upharpoonright_X \sigma')$ whenever $s \in X$, and $\upharpoonright_X (s_\iota^\lambda,\sigma') = \upharpoonright_X \sigma'$ otherwise.

Now we define the relation $\sim_{\mathcal{L}^\sharp}$ which stipulates that the rule r_1 applies on E_1 along ϕ_1 the same way as the rule r_2 on E_2 along ϕ_2. More formally, we write $(r_1, E_1, \phi_1) \sim_{\mathcal{L}^\sharp} (r_2, E_2, \phi_2)$ whenever the following properties are all satisfied:

1. $r_1 = r_2$;
2. $E_1 \equiv^\sharp E_2$;
3. $\phi_2 = \psi \circ \phi_1$, where ψ is the permutation which tracks how the sub-interface $\upharpoonright_{\mathbf{test}_r^{ACM}(A_i)} (A_i(\sigma_i))$ is moved in the proof that $E_1 \equiv^\sharp E_2$, for any agent $A_i(\sigma_i)$ occurring in E_1.
   More precisely, the transposition $\genfrac{[}{]}{0pt}{}{i}{i+1}$ is associated to an agent permutation of the agents at position i and $i+1$; the transposition $\genfrac{[}{]}{0pt}{}{1}{2}$ is associated to a step which permutes the sub-interface $\mathbf{test}_r^{ACM}(A)$ of two agents of type A, for any agent type $A \in \mathcal{A}$; any other step is associated with the identity function (over $\mathbb{N}$). The function ψ is defined as the composition of all the permutations (in the reverse order) which are associated to the elementary steps in the proof that $E_1 \equiv^\sharp E_2$.
4. the result of the application of r_1 to E_1 along ϕ_1 is $\equiv$-equivalent to the result of the application of r_2 to E_2 along ϕ_2.

Definition 17. *(Abstract WLTS of a Kappa system) Let $\mathcal{R} = (\pi_0^{\mathcal{R}}, \{r_1, \ldots, r_n\})$ be a Kappa system. We define the WLTS $\tilde{\mathcal{W}_{\mathcal{R}}} = (\tilde{\mathcal{X}}, \tilde{\mathcal{L}}, \tilde{w}, \tilde{\pi_0})$ where:*

- $\tilde{\mathcal{X}}$ *is the set of all* $\equiv^\sharp$*-equivalent class of mixture;*
- $\tilde{\mathcal{L}}$ *is the set of all* $\equiv^\sharp_{\mathcal{L}}$*-equivalent class of triples* (r, E, ϕ) *such that* ϕ *is an embedding between the lhs* E_ℓ *of* r *and* E*;*
- $\tilde{w}(\tilde{x}, \tilde{l}, \tilde{x}')$ *is equal to* $\frac{k}{|[E_\ell, E_\ell]|}$ *whenever there exist a rule* $r = E_\ell \to E_r@k$*, two mixtures* E *and* E'*, and an embedding* $\phi \in [E_\ell, E]$*, such that* $\tilde{x} = [E]_{\equiv^\sharp}$*,* $\tilde{l} = [r, E, \phi]_{\equiv^\sharp_{\mathcal{L}}}$*,* $\tilde{x}' = [E']_{\equiv^\sharp}$*, and* E' *is the result (up to* $\equiv$*) of the application of* r *along* ϕ *to the mixture* E*; otherwise* $\tilde{w}(\tilde{x}, \tilde{l}, \tilde{x}') = 0$*;*
- *for any* $\tilde{x} \in \tilde{\mathcal{X}}$*,* $\tilde{\pi}_0(\tilde{x}) = \sum_{E' \in Dom(\pi_0^{\mathcal{R}}) \cap \tilde{x}} \pi_0^{\mathcal{R}}(E')$*.*

We define the relation $\sim$ over $\mathcal{X}$ by $[E_1]_\equiv \sim [E_2]_\equiv$ if, and only if, $E_1 \equiv^\sharp E_2$ and the relation $\sim_{\mathcal{L}}$ over $\mathcal{L}$ by $[\lambda_1]_{\equiv_{\mathcal{L}}} \sim_{\mathcal{L}} [\lambda_2]_{\equiv_{\mathcal{L}}}$ if, and only if, $\lambda_1 \equiv^\sharp_{\mathcal{L}} \lambda_2$. The pair $(\sim, \sim_{\mathcal{L}})$ of relations induces an abstraction of $\mathcal{W}_{\mathcal{R}}$ as formalized in Sect. 6.

6. Abstraction

We introduce abstractions on WLTS by aggregating the states and labels into partition classes. We obtain a new WLTS defined over the aggregated states and labels. Each non-trivial abstraction is a loss of information. However some of them are such that it is possible to do the stochastic analysis on the aggregates rather than on concrete states. We address the problem of characterizing when this is possible, and if so, how the weights in the abstracted system are computed. We also discuss the reverse process - given the abstracted system, and a particular probability distributions over the aggregates, whether we can make conclusions about the traces in the concrete system. We do the general theoretical analysis of the abstractions on WLTS, and afterwards we show the relation with the reduction of Kappa systems, that is presented in Sect. 5.

6.1. Abstraction of WLTS

Definition 18. *(Abstraction) Consider a WLTS* $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$*, and a pair of equivalence relations* $(\sim, \sim_{\mathcal{L}}) \in \mathcal{X}^2 \times \mathcal{L}^2$*, such that each* $\sim$*-equivalence class and each* $\sim_{\mathcal{L}}$*-equivalence class is finite. We denote the equivalence classes by* $\tilde{x}$*,* $\tilde{l}$*, and we write* $x \in \tilde{x}$*, to indicate that* x *belongs to the equivalence class* $\tilde{x}$*, and* $l \in \tilde{l}$ *to indicate that the label* l *belongs to the equivalence class* $\tilde{l}$*. Moreover, we denote by* $\mathcal{X}_{/\sim}$ *and by* $\mathcal{L}_{/\sim_{\mathcal{L}}}$ *the set of equivalence classes of* $\mathcal{X}$ *and* $\mathcal{L}$*.*

A WLTS of the form $\tilde{\mathcal{W}} = (\mathcal{X}_{/\sim}, \mathcal{L}_{/\sim_{\mathcal{L}}}, \tilde{w}, \tilde{\pi}_0)$*, where* $\tilde{\pi}_0(\tilde{x}) = \sum\{\pi_0(x) \mid x \in \tilde{x}\}$ *is called an abstraction of* $\mathcal{W}$*, induced by the pair of equivalence relations* $(\sim, \sim_{\mathcal{L}})$*. Note that several abstractions can be induced by* $\mathcal{W}$*, depending on how* $\tilde{w}$ *is defined.*

Moreover, for any two cylinder sets of traces $\tilde{\tau}_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\tilde{\mathcal{W}})$ *and* $\tau_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\mathcal{W})$*, we say that* $\tilde{\tau}_{\mathbb{IR}} = \tilde{x}_0 \overset{\tilde{l}_1, I_1}{\to} \tilde{x}_1 \ldots \tilde{x}_{k-1} \overset{\tilde{l}_k, I_k}{\to} \tilde{x}_k$ *is an abstraction of* $\tau_{\mathbb{IR}} = x_0 \overset{l_1, I_1}{\to} x_1 \ldots x_{k-1} \overset{l_k, I_k}{\to} x_k$*, and we write it* $\tau_{\mathbb{IR}} \in \tilde{\tau}_{\mathbb{IR}}$*.*

Definition 19. *(Sound abstraction: Aggregation) We say that the abstraction $\tilde{\mathcal{W}}$ is a sound abstraction of $\mathcal{W}$, if the probability of any cylinder set of traces $\tilde{\tau}_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\tilde{\mathcal{W}})$ is equal to the sum of the probabilities of all the cylinder sets of traces $\tau_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\mathcal{W})$, whose abstraction is $\tilde{\tau}_{\mathbb{IR}}$:*

$$\pi(\tilde{\tau}_{\mathbb{IR}}) = \sum \{\pi(\tau_{\mathbb{IR}}) \mid \tau_{\mathbb{IR}} \in \tilde{\tau}_{\mathbb{IR}}\}.$$

We introduce a function $\gamma : \mathcal{X}_{/\sim} \to (\mathcal{X} \to [0,1])$ which assigns to each partition class $\tilde{x} \in \mathcal{X}_{/\sim}$ a probability distribution over the states $x \in \tilde{x}$ of this partition class. The set of all such vectors γ, denoted by $\Gamma_{\mathcal{X},\sim}$, is defined as:

$$\{\gamma \mid \gamma : \mathcal{X}_{/\sim} \to (\mathcal{X} \to [0,1]) \wedge \forall \tilde{x} \in \tilde{\mathcal{X}}, \sum_{x \in \tilde{x}} \gamma(\tilde{x}, x) = 1\}.$$

We can think of the value $\gamma(\tilde{x}, x)$ as the *conditional probability* of being in the state x, knowing that we are in state $\tilde{x}$, i.e. $\Pr(X_t = x \mid X_t \in \tilde{x}) = \gamma(\tilde{x}, x)$. We note that, when thinking of γ as the conditional probability, it should be a time-dependent value. However, we refer to γ as to a single, constant distribution. This will be justified in Lem. 1.

Definition 20. *(Complete abstraction: Deaggregation) We say that the abstraction $\tilde{\mathcal{W}}$ is a complete abstraction of $\mathcal{W}$ for $\gamma \in \Gamma_{\mathcal{X},\sim}$, if the following holds. Given the probability of an arbitrary abstract cylinder set of traces of length $k \geq 1$, that ends in the abstract state $\tilde{x}_k$ (written $\tilde{\tau}_{\mathbb{IR}} \to \tilde{x}_k$), we can recompute the probability of ending the trace in the concrete state $x_k \in \tilde{x}_k$ as follows:*

$$\pi(\tilde{\tau}_{\mathbb{IR}} \to x_k) = \gamma(\tilde{x}_k, x_k) \cdot \pi(\tilde{\tau}_{\mathbb{IR}} \to \tilde{x}_k).$$

Sound and complete abstractions $\tilde{\mathcal{W}}$ cannot be induced by any pair of relations $(\sim, \sim_{\mathcal{L}})$, because there might not exist a weighting function $\tilde{w} : \mathcal{X}_{/\sim} \times \mathcal{L}_{/\sim_{\mathcal{L}}} \times \mathcal{X}_{/\sim} \to \mathbb{R}$, such that the conditions from Dfn. 19 and Dfn. 20 are met. Moreover, even if such $\tilde{w}$ exists, the remaining question is whether the information on the abstract system is enough to compute them.

We now restate the main Theorem from [15], that the abstractions for Kappa systems, that we resumed in Sect. 5, are sound and complete.

Theorem 1. *(The abstraction induced by the ACM is sound and complete) Given a Kappa system $\mathcal{R} = (\pi_0^{\mathcal{R}}, \{r_1, ..., r_n\})$, and a valid ACM for the rule set $\mathcal{R}$, the abstraction $\tilde{\mathcal{W}_{\mathcal{R}}} = (\mathcal{X}_{/\sim}, \mathcal{L}_{/\sim}, \tilde{w}, \tilde{\pi}_0)$ induced by the pair of equivalence relations $(\sim, \sim_{\mathcal{L}}) \subseteq \mathcal{X}^2 \times \mathcal{L}^2$, as proposed in the Def. 17 is a sound and complete abstraction of the $\mathcal{W}_{\mathcal{R}} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, provided that for any two mixtures M and M' such that $M \equiv^\sharp M'$, we have:*

$$\pi_0([M]_\equiv) \cdot |[M', M']| = \pi_0([M']_\equiv) \cdot |[M, M]|.$$

We consider a mixture M. We denote by $x \in \mathcal{X}$ the equivalence class $[M]_\equiv$, and by $\tilde{x} \in \tilde{\mathcal{X}}$ the equivalence class $[M]_{\equiv^\sharp} = [x]_\sim$. The conditional probability

$\gamma(\tilde{x}, x)$ is computed as the ratio of the number of automorphisms of x (embedding between x and x) and the sum of the number of automorphisms of any $\sim$-equivalent state. Thus we have:

$$\gamma(\tilde{x}, x) = \frac{|[x, x]|}{\sum\{|[x', x']| \ \mid \ x \sim x'\}}.$$

The reader can find the detailed proof in [15].

Interestingly, our reduction procedure does not depend on the kinetic rates of the rules. Indeed, rules describe explicitly which context can impact on the kinetic of some interactions. This information is enough to define a sound approximation of the flow of information. Thus our analysis is *semi-quantitative*, it provides properties that are correct whatever the valuation of the kinetic rates is. That is why, we do not give the values to the kinetic rates in our case study. Last, one can notice that, given some additional hypotheses on the rate of some rules, the models could be refined further. For instance, in the case of study (Sect. 2), if the rules r21 and r22 have the same kinetic rates, then the sites a and b of *IR* have a symmetric role in the system. We could consider this symmetry to reduce the set of considered fragments, for instance, by identifying the fragments $\mathit{IR}\big(a^1,b\big)$, $\mathit{Ins}\big(a^1\big)$ and $\mathit{IR}\big(a,b^1\big)$, $\mathit{Ins}\big(a^1\big)$, as done in [6].

6.2. Lumpability

Now we define different versions of lumpability and investigate the relationship with sound and complete abstractions.

Definition 21. *(Lumped process) Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, where $\mathcal{X} = \{x_1, x_2, ...\}$, and a partition $\sim \subseteq \mathcal{X} \times \mathcal{X}$ on its state space, we observe the continuous-time stochastic process $\{X_t\}_{t\in\mathbb{R}_0^+}$, that is generated by $\mathcal{W}$ (Dfn. 2). We define the lumped process $\{Y_t\}$ on the state space $\mathcal{X}_{/\sim} = \{\tilde{x}_1, \tilde{x}_2, ...\}$ (denoted by capital indices, i.e. $\tilde{x}_I$, $\tilde{x}_J$) and with initial distribution $\tilde{\pi}_0$, so that*

$$\Pr(Y_t = \tilde{x}_J \ \mid \ Y_0 = \tilde{x}_0) = \Pr(X_t \in \tilde{x}_J \ \mid \ X_0 \in \tilde{x}_0).$$

The lumped process is not necessarily a Markov process.

Definition 22. *(Lumpability) Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ that generates the process $\{X_t\}$, we say that it is lumpable with respect to the equivalence relation $\sim \subseteq \mathcal{X} \times \mathcal{X}$ if and only if its lumped process $\{Y_t\}$ has the Markov property.*

The evolution of a process depends on the initial distribution, and so does the lumpability property. We thus define the set of initial distributions, for which the lumpability holds. We denote the set of all probability distributions over $\mathcal{X}$ as $\mathcal{P}_{\mathcal{X}}$:

$$\mathcal{P}_{\mathcal{X}} = \{\pi \ \mid \ \pi : \mathcal{X} \to [0, 1] \text{ and } \sum_{x_i \in \mathcal{X}} \pi(x_i) = 1\}.$$

Moreover, we denote the set of initial distributions that produce a chain lumpable with respect to the given equivalence relation $\sim$ by $\mathcal{P}^I_{\mathcal{X},\sim}$:

$$\mathcal{P}^I_{\mathcal{X},\sim} = \left\{ \pi \;\middle|\; \begin{array}{l} \text{the lumped process initialized with } \pi \\ \text{is lumpable with respect to } \sim \end{array} \right\}.$$

Whenever a distribution $\pi \in \mathcal{P}_{\mathcal{X}}$ is positive on the equivalence class $\tilde{x}$, i.e. $\sum\{\pi(x) \mid x \in \tilde{x}\} > 0$, we denote by $\pi|_{\tilde{x}}(x)$, the conditional distribution over the states of $\tilde{x}$: $\pi|_{\tilde{x}}(x) = \pi(x)/\pi(\tilde{x})$, when $x \in \tilde{x}$, and $\pi|_{\tilde{x}}(x) = 0$, otherwise.

Definition 23. *(Strong and weak lumpability) Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ that generates the process $\{X_t\}$, and an equivalence relation $\sim \subseteq \mathcal{X} \times \mathcal{X}$, we say that $\{X_t\}$ is:*

- *strongly lumpable with respect to $\sim$, if the lumped process $\{Y_t\}$ is Markov with respect to any initial distribution, i.e. $\mathcal{P}^I_{\mathcal{X},\sim} = \mathcal{P}_{\mathcal{X}}$;*
- *weakly lumpable with respect to $\sim$, if there exists an initial distribution that makes the lumped process $\{Y_t\}$ Markov, i.e. $\mathcal{P}^I_{\mathcal{X},\sim} \neq \emptyset$.*

Note that the definitions of strong and weak lumpability involve the quantifiers "for all" and "exists" over the probability distributions over a set of states. Thus, checking for either of them involves in general an infinite number of checks. People have given sufficient conditions of strong and weak lumpability on discrete-time Markov chains (DTMC's) [25, 33]. The results had been extended to the continuous-time case [4, 34]. We rephrase the sufficient conditions stated therein.

In order to understand the sense of the weak lumpability characterization, we discuss the meaning of γ. We recall the semantics of a WLTS $\mathcal{W}$ by observing the cylinder sets of traces, i.e. $\tau_{\mathbb{IR}} = x_0 \overset{l_1,I_1}{\to} x_1 \ldots x_{k-1} \overset{l_k,I_k}{\to} x_k \in \mathcal{T}_{\mathbb{IR}}(\mathcal{W})$. The abstraction $\tilde{\mathcal{W}}$ of $\mathcal{W}$, induced by $(\sim, \sim_L)$ generates an abstract cylinder set of traces, denoted $\tilde{\tau}_{\mathbb{IR}} = \tilde{x}_0 \overset{l_1,I_1}{\to} \tilde{x}_1 \ldots \tilde{x}_{k-1} \overset{l_k,I_k}{\to} \tilde{x}_k \in \mathcal{T}_{\mathbb{IR}}(\tilde{\mathcal{W}})$.

For any cylinder set of traces $\tilde{\tau}_{\mathbb{IR}} \in \mathcal{T}_{\mathbb{IR}}(\tilde{\mathcal{W}})$, we denote by $\gamma_{\tilde{\tau}_{\mathbb{IR}}}$ the distribution of the conditional probabilities over the lumped state $\tilde{x}_k$, knowing that the abstract cylinder of traces $\tilde{\tau}_{\mathbb{IR}}$, which ends in the abstract state $\tilde{x}_k$, was observed, i.e.

$$\gamma_{\tilde{\tau}_{\mathbb{IR}}}(x_k) = \frac{\pi(\tilde{\tau}_{\mathbb{IR}} \to x_k)}{\pi(\tilde{\tau}_{\mathbb{IR}})}.$$

The definition of the complete abstraction suggests that, if $\gamma_{\tilde{\tau}_{\mathbb{IR}}}$ was independent of the traces on which it is conditioned, i.e. $\tilde{\tau}_{\mathbb{IR}}$, then the completeness would hold.

Theorem 2. *(Lumpability on CTMCs) Let us observe a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ that generates the process $\{X_t\}$, and an equivalence relation $\sim \subseteq \mathcal{X} \times \mathcal{X}$. We consider the rate matrix $\mathbf{R} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If the lumped process is Markov, then we denote its rate matrix by $\tilde{\mathbf{R}} : \mathcal{X}_{/\sim} \times \mathcal{X}_{/\sim} \to \mathbb{R}$. Then we have the following characterizations about the lumped process $\{\tilde{X}_t\}$:*

- *If for all* $x_{i_1}, x_{i_2} \in \mathcal{X}$ *such that* $x_{i_1} \sim x_{i_2}$*, and for all* $\tilde{x}_J \in \mathcal{X}_{/\sim}$*, we have that*

$$\sum_{x_j \in \tilde{x}_J} R(x_{i_1}, x_j) = \sum_{x_j \in \tilde{x}_J} R(x_{i_2}, x_j), \tag{2}$$

  then $\{X_t\}$ *is strongly lumpable with respect to* $\sim$*; We have:*

$$\tilde{R}(\tilde{x}_I, \tilde{x}_J) = \sum \{R(x_{i_1}, x_j) \ \mid \ x_j \in \tilde{x}_J\};$$

- *If there exists a family of probability distributions over the lumped states,* $\gamma \in \Gamma_{\mathcal{X},\sim}$*, such that for all* $x_{j_1}, x_{j_2} \in \mathcal{X}$ *such that* $x_{j_1} \sim x_{j_2}$ *and for all* $\tilde{x}_I \in \mathcal{X}_{/\sim}$*, we have that*

$$a(x_{j_1}) = a(x_{j_2}) \text{ and } \frac{\sum_{x_i \in \tilde{x}_i} R(x_i, x_{j_1})}{\gamma(\tilde{x}_J, x_{j_1})} = \frac{\sum_{x_i \in \tilde{x}_I} R(x_i, x_{j_2})}{\gamma(\tilde{x}_J, x_{j_2})}, \tag{3}$$

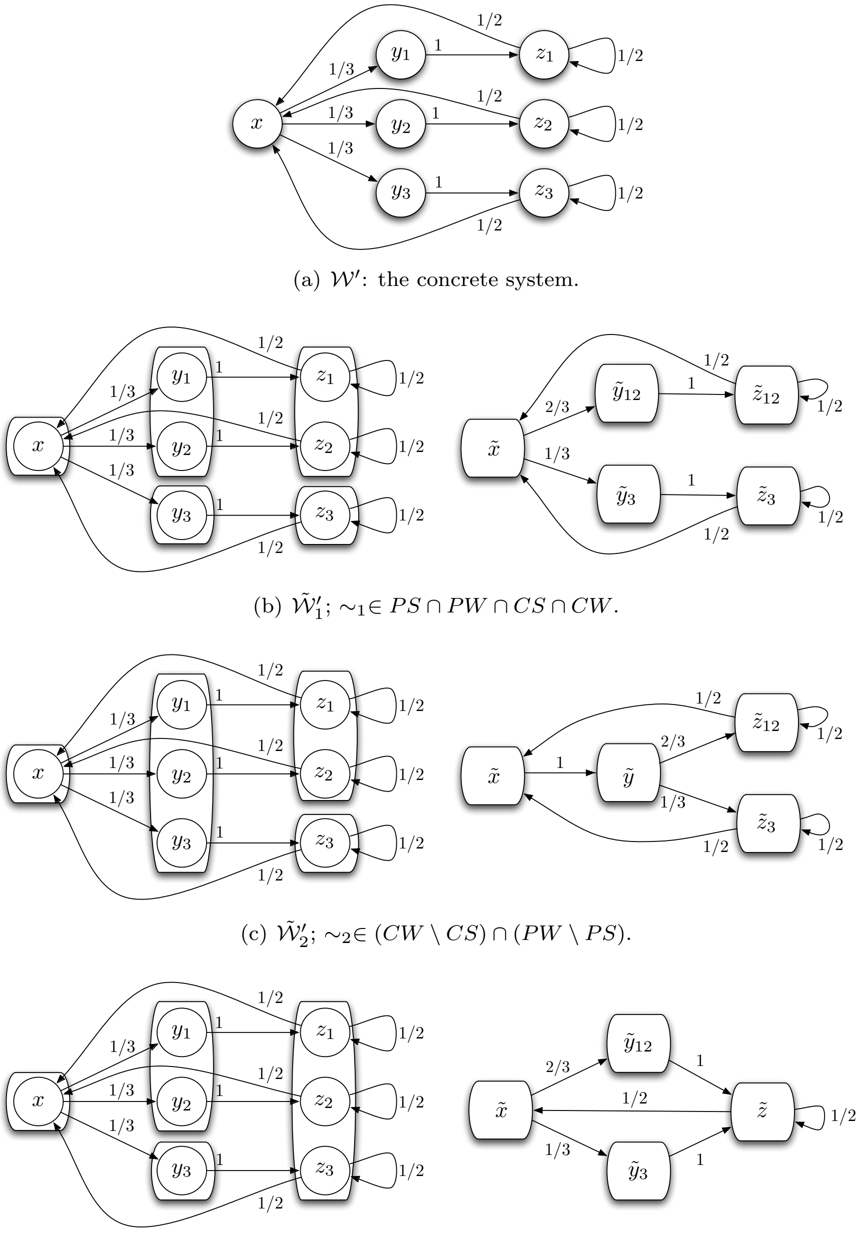
  then

  1. *If the distribution* γ *is in accordance with* π_0*, i.e.* $\pi_0|_{\mathcal{X}_{/\sim}} = \gamma$*, then for any finite sequence of states* $(x_0, \ldots, x_k) \in \mathcal{X}^{k+1}$ *and any sequence of time intervals* $(I_1, \ldots, I_k) \in \mathbb{IR}^k$*, we consider the set* $\tilde{\tau}_{\mathbb{IR}}$ *of the traces of the form* $x'_0 \overset{l_1,t_1}{\rightarrow} x'_1 \ldots x'_{k-1} \overset{l_k,t_1+\ldots+t_k}{\rightarrow} x'_k$*. For all* i*,* $0 \leq i \leq k$ *and* $x_i \sim x'_i$*, and for all* i*,* $1 \leq i \leq k$*,* $t_i \in I_i$ *and* $l_i \in \mathcal{L}$*, we have that: if* $\pi(\tilde{\tau}_{\mathbb{IR}}) > 0$ *then* $\gamma_{\tilde{\tau}_{\mathbb{IR}}} = \gamma$*.*
     In other words, knowing that we are in state $\tilde{x}_I$*, the conditional probability of being in state* $x \in \tilde{x}_I$ *is invariant of time.*
  2. *The process* $\{X_t\}$ *is weakly lumpable with respect to* $\sim$*. Moreover, we have:*

$$\tilde{R}(\tilde{x}_I, \tilde{x}_J) = \frac{\sum \{R(x_i, x_{j_2}) \ \mid \ x_i \in \tilde{x}_I\}}{\gamma(\tilde{x}_J, x_{j_2})};$$

One shall notice that Thm. 2 gives a weaker condition than the completeness of WLTS abstraction (eg see Dfn. 20). The main reason is that we do not 'track' transition labels, in the sense that we observe the abstraction on the cylinder sets of traces induced only by $\sim$, and not also by $\sim_L$. Yet, in the particular case when states fully define the transition labels (ie, if $w(x_1, l_1, x'_1) > 0$, $w(x_2, l_2, x'_2) > 0$, $x_1 \sim x_2$, and $x'_1 \sim x'_2$, then $l_1 \sim_{\mathcal{L}} l_2$), the given condition for weak lumpability coincides with the definition of the complete abstraction of WLTS.

The characterization of weak (resp. strong) lumpability given in Thm. 2 is sufficient, but not a necessary condition: there exist systems which are strongly or weakly lumpable, but do not satisfy the conditions given in the theorem. Interestingly, there are systems, such that the characterization from Thm. 2 would detect as strong, but not weakly lumpable, which is counter-intuitive with the terminology. This is indeed a consequence of the fact that we have strengthened our conditions: whereas the strong lumpability (in Def. 23) implies weak lumpability (in Def. 23), neither strong lumpability (in Def. 23), nor the



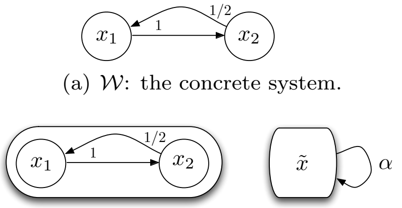
(a) $\mathcal{W}'$: the concrete system.

(b) $\tilde{\mathcal{W}}_1'$; $\sim_1 \in PS \cap PW \cap CS \cap CW$.

(c) $\tilde{\mathcal{W}}_2'$; $\sim_2 \in (CW \setminus CS) \cap (PW \setminus PS)$.

(d) $\tilde{\mathcal{W}}_3'$; $\sim_3 \in CS \setminus CW$.

Figure 11: Different abstractions of the same system: (a) $\mathcal{W}'$, a concrete WLTS; (b) $\tilde{\mathcal{W}}_1'$ is derived by aggregating the state space with the equivalence relation $\sim_1$. The Markov chain of $\mathcal{W}$ is strongly and weakly lumpable with respect to the relation $\sim_1$ ($\sim_1 \in PS \cap PW$). Moreover, it also satisfies the conditions for weak and strong lumpability given by Thm. 2: $\sim_1 \in CS \cap CW$; (c) $\sim_2$ is an example of weak, and not strong lumping; (d) $\sim_3$ is an example of strong, but not weak lumping. More detailed discussion on these examples is in the proof of Lem. 1. The summary of possible relations between properties PS, PW, CS, CW is given in Fig. 13.



(a) $\mathcal{W}$: the concrete system.

(b) $\tilde{\mathcal{W}}$; $\sim \in (PS \backslash CS) \cap (PW \backslash CW)$.

Figure 12: The system $\mathcal{W}$ is abstracted by the relation $\sim$ to a system $\tilde{\mathcal{W}}$. Aggregating all the states of a WLTS is trivially both strongly and weakly lumpable (Dfn. 23). Yet, none of the criteria from Thm. 2 are met. In other words, in Thm. 2, only sufficient conditions for strong (weak) lumpability are stated.

strengthened version of strong lumpability (in Thm. 2) implies the strengthened version of weak lumpability (in Thm. 2).

One shall also notice that the conditions of Thm. 2 imply that: in order to aggregate two states in the CTMC, they must not have different waiting times until the next transition (e.g. they should have the same activity). It is stated explicitly in the characterization of weak lumpability and it can be obtained by summation over the outgoing class in the characterization of strong lumpability.

We consider a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, and the set of all equivalence relations $\sim$ on $\mathcal{X}$, denoted $PT_{\mathcal{X}}$. We introduce the subsets of $PT_{\mathcal{X}}$, denoted PS, PW, CS, CW in the following meaning: (i) PS -the set of all equivalence relations such that $\{X_t\}$ is strongly lumpable with respect to $\sim$; (ii) PW - the set of all equivalence relations such that $\{X_t\}$ is weakly lumpable with respect to $\sim$; (iii) CS - the set of all equivalence relations such that $\{X_t\}$ satisfies the condition for strong lumpability given in the Thm. 2; (iv) CW - the set of all equivalence relations such that $\{X_t\}$ satisfies the condition for weak lumpability given in the Thm. 2.

Example 6.1. *We give in Figs. 11 and 12 some examples of equivalence relations over a WLTS, and check whether or not they belong to PS, PW, CS, and/or CW. The result will be shown later in the proof of Lem. 1.*

□

Lemma 1. *(Relations on lumpability properties and conditions) Consider an arbitrary WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ and the equivalence relation $\sim \subseteq \mathcal{X} \times \mathcal{X}$. We have the following relations: (1a) if $\sim \in CS$ then $\sim \in PS$; (1b) if $\sim \in CW$ then $\sim \in PW$; (1c) if $\sim \in PS$ then $\sim \in PW$, (2a) If $\sim \in PS$, that does not imply $\sim \in CS$; (2b) If $\sim \in PW$, that does not imply $\sim \in CW$; (2c) If $\sim \in PW$, that does not imply $\sim \in PS$; (2d) If $\sim \in CW$, that does not imply $\sim \in CS$; (2e) If $\sim \in CS$, that does not imply $\sim \in CW$. These relations are summarized in Fig. 13.*

Proof. The statements (1a), (1b), and (1c) trivially follow from the Dfn. 22 and Thm. 2.

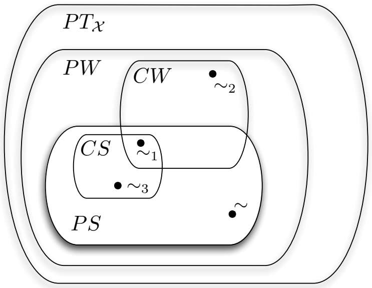


Figure 13: Graphical illustration of Lemma 1: Among all the partitions on the state space $\mathcal{X}$, the sets PW, CW, CS, PS relate as following: CS is a subset of PS, CW is a subset of PW and PS is a subset of PW. Moreover, none of the relations hold in the opposite direction, and the witnesses for this are $\sim$ (defined in Fig. 12), and $\sim_2$ (defined in Fig. 11). Furthermore, CW and CS intersect properly: neither is a subset of another. The witnesses for this are relations $\sim_2$ and $\sim_3$ (both defined in Fig. 11).

To show (2a) and (2b), we notice that in every WLTS, if we lump all the states, then both strong and weak lumpability holds. This is because a process that has only one state necessarily satisfies the Markovian property. In other words, to show (2a) and (2b), it is thus enough to consider a WLTS with two states which have different activity, and an equivalence relation which relates each pair of states. We give such an example in Fig. 12.

To show (2c), (2d) and (2e), we consider the WLTS $\mathcal{W}'$ specified in the Fig. 11(a), with the state space $\mathcal{X} = \{x, y_1, y_2, y_3, z_1, z_2, z_3\}$. Let $\sim_1$ be an equivalence relation on $\mathcal{X}$, such that $y_1 \sim_1 y_2$ and $z_1 \sim_1 z_2$. By lumping the states by $\sim_1$, we get the system $\tilde{\mathcal{W}}'_1$, as shown in Fig. 11(b). It is easy to check that $\sim_1 \in CS$. Moreover, we have that $\sim_1 \in CW$, since for

$$\gamma = \left(\begin{array}{ccccc} x & y_{12} & y_3 & z_{12} & z_3 \\ 1 & (0.5, 0.5) & 1 & (0.5, 0.5) & 1 \end{array} \right)$$

the weak lumpability condition is satisfied, so $\sim_1 \in CS \cap CW$. It follows from (1a) and (1b) that $\sim_1 \in CS \cap CW \cap PS \cap PW$.

We further lump the states y_{12} and y_3, by taking the transitive closure of the relation $\sim_1$ union (y_1, y_3), denoted $\sim_2 = tc(\sim_1 \cup (y_1, y_3))$ (Fig. 11(c)). This lumping is such that $\sim_2 \notin CS$ because we have

$$y_1 \sim y_3, \text{ but } w(y_1, l, z_{12}) > 0, \text{ and } w(y_3, l, z_{12}) = 0.$$

On the other hand, for

$$\gamma = \left(\begin{array}{cccc} x & y_{123} & z_{12} & z_3 \\ 1 & (1/3, 1/3, 1/3) & (0.5, 0.5) & 1 \end{array} \right)$$

we argue that $\sim_2 \in CW$ (which proves (2d)). Therefore, if the initial distribution is in accordance with γ, the abstraction $\tilde{\mathcal{W}}'_2$ is sound and complete. Since

$\sim_2 \in CW$, it follows from (1b) that $\sim_2 \in PW$. But $\sim_2 \notin PS$, since the $\mathcal{W}'$ is not lumpable with respect to $\sim_2$ for the initial distribution which maps the state y_3 to 1, and any other state to 0. This proves (2c).

If we rather lump z_1 and z_2, by $\sim_3$, the transitive closure of $(\sim_1 \cup (z_1, z_3))$, we get the system $\tilde{\mathcal{W}}_3'$ (Fig. 11(d)). This system is such that $\sim_3 \in CS \setminus CW$. More precisely, we cannot find a γ which would witness $\sim_3 \in CW$: if such a γ existed, we would have $\gamma(\{x\})(x) = 1$, and consequently $\gamma(y_{12}) = (0.5, 0.5)$, and $\gamma(y_3) = 1$. This implies that the conditional distribution $\gamma(z_{123})$ cannot be invariant of time - it will alternate between the distributions $(0, 0, 1)$ and $(0.5, 0.5, 0)$, depending on the choice made in x. Note that, since $\sim_3 \in CS$, it follows that $\sim_3 \in PS$, and this implies $\sim_3 \in PW$ (which proves (2e)). □

This discussion indicates that if we decide to check for weak lumpability instead of for strong by using the characterization from Thm. 2, it might happen that we eliminate the aggregations that are strongly lumpable. In the case of reductions of Kappa systems, we will use the weak lumpability characterization.

6.3. Bisimulations

Aiming to define the algorithm that is abstracting the WLTS of a Kappa system, we start by redefining the lumpability properties in the bisimulation notions. Bisimulation is typically defined by logically characterizing the distinguishing property of the states that may be aggregated.

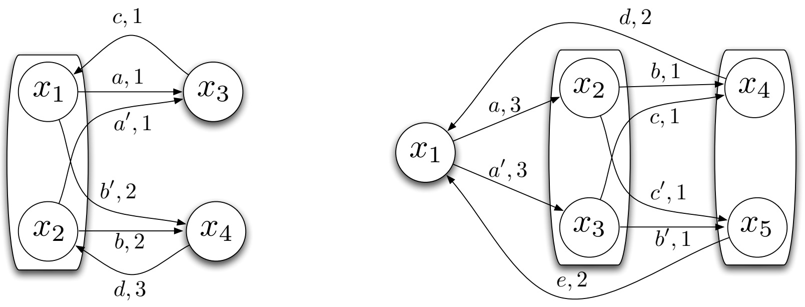
We define three kinds of bisimulation relations on the WLTS, which are based on the lumpability characterizations given in Thm. 2. We adopt the terminology of [5]. The *forward* bisimulations arise from the characterization for strong lumpability: the bisimilar states have the same forward behavior in the sense that they are each targeting any other lumped state with the same total affinity (total outgoing rate). This concept is well established for dependability or performance analysis [21, 20]. What we use in the abstractions of Kappa systems is *backward* bisimulation. The bisimilar states have the same backward behavior in the sense that they are reached by the predecessors from one lumped state with the same probabilistic quantity, which becomes the rate in the abstract system. It is however less established and only applied in very few approaches for stochastic modelling [38]. The *backward uniform* bisimulation is an instance of a backward bisimulation with an additional constraint that only the equally-probable states may be aggregated.

Definition 24. *Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, we define the function $\delta_F : \mathcal{X} \times \wp(\mathcal{L}) \times \wp(\mathcal{X}) \to \mathbb{R}_0^+$ as follows:*

$$\delta_F(x_i, L, X) = \sum \{ |w(x_i, l, x_j)| \mid l \in L \text{ and } x_j \in X \}.$$

Furthermore, given an equivalence relation $\sim$ over $\mathcal{X}$ and a family of probability distributions over the partitions $\gamma \in \Gamma_{\mathcal{X},\sim}$, we define the function $\delta_B : \wp(\mathcal{X}) \times \wp(\mathcal{L}) \times \mathcal{X} \to \mathbb{R}_0^+$ as follows:

$$\delta_B(X, L, x_j) = \sum \{ \gamma(\tilde{x}_i, x_i) \cdot |w(x_i, l, x_j)| \mid l \in L, x_i \in X \}.$$



(a) A forward bisimulation.

(b) A uniform backward bisimulation.

Figure 14: Examples of bisimulations: (a) aggregating the labels so that $a \sim_L a'$ and $b \sim_L b'$ induces a foward bisimulation; (b) aggregating the labels so that $a \sim_L a'$, $b \sim_L b'$ and $c \sim_L c'$ induces a uniform backward bisimulation.

Specifically, if γ is a family of uniform distributions over each $\sim$-equivalence classes, we can express the latter expression in terms of cardinalities of the equivalence classes as follows:

$$\delta_{BU}(X, L, x_j) = \sum \left\{ \frac{|w(x_i, l, x_j)|}{|\tilde{x}_i|} \mid l \in L, x_i \in X \right\}.$$

Intuitively, the quantity $\delta_F(x_i, L, X)$ is the sum of the weights of outcoming transitions from a state x_i to a state in X with labels in the state L. Conversely, whenever X is a $\sim$-equivalence class, $\delta_B(X, L, x_j)$ is the expected weight of the incoming transitions from a state in the X to the state x_j with labels in the set L, under the assumption that whenever the system is in a state in the class $\tilde{x}_i$, the distribution of states is given by the mapping $[x_i \mapsto \gamma(\tilde{x}_i, x_i)]$.

Example 6.2. *In the WLTS in Fig. 14(b), we have:*

$$\delta_F(x_1, \{a, a'\}, \{x_2, x_3\}) = w(x_1, a, x_2) + w(x_1, a', x_3) = 3 + 3 = 6.$$

In the WLTS in Fig. 14(a), for the equivalence relation $\sim$ over the states which identifies x_1 and x_2, and for the family γ of probability distributions over the partition of states, which is defined by $\gamma(\tilde{x}, x) = \frac{1}{|\tilde{x}|}$, we have:

$$\delta_B(\{x_1, x_2\}, \{b, b'\}, x_4) = \frac{w(x_1, b', x_4)}{2} + \frac{w(x_2, b, x_4)}{2} = \frac{2}{2} + \frac{2}{2} = 2.$$

□

Definition 25. *(Forward and backward Markov bisimulation) Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, and $(\sim, \sim_L)$ a pair of equivalence relations respectively over $\mathcal{X}$ and $\mathcal{L}$, we say that $(\sim, \sim_L)$ is a*

1. *Forward Markov Bisimulation, if for all x_i and x_j, the following is satisfied: $x_i \sim x_j$, iff for all equivalence classes $\tilde{x} \in \mathcal{X}_{/\sim}$,$\tilde{l} \in \mathcal{L}_{/\sim_L}$, we have that $a(x_i) = a(x_j)$ and $\delta_F(x_i, \tilde{l}, \tilde{x}) = \delta_F(x_j, \tilde{l}, \tilde{x})$.*

   Remark. Note that this involves the bisimulation in the classical sense: if x_i has a successor in some class, x_j has it as well, and they are related by appropriate labels (and probabilities in this case).

2. *Backward Markov bisimulation, if there exists an $\gamma \in \Gamma_{\mathcal{X},\sim}$ such that for all x_i and x_j the following is satisfied: $x_i \sim x_j$, iff for all equivalence classes $\tilde{x} \in \mathcal{X}_{/\sim}$, $\tilde{l} \in \mathcal{L}_{/\sim_L}$, we have that $a(x_i) = a(x_j)$ and $\gamma(\tilde{x}_i, x_i) \cdot \delta_B(\tilde{x}, \tilde{l}, x_i) = \gamma(\tilde{x}_j, x_j) \cdot \delta_B(\tilde{x}, \tilde{l}, x_j)$.*

   Remark. Note that this involves that if the system is in the class of states $\tilde{x}_i$ with a distribution of states given by the mapping $[x \mapsto \gamma(\tilde{x}_i, x)]$, and if a transition with a label in the class $\tilde{\lambda}$ to a state in the class $\tilde{x}_j$ is picked stochastically according to the weight of these transitions, then, the probability that the system is in a given state x is given by the mapping $[x \mapsto \gamma(\tilde{x}_j, x)]$. Thus, conditional probability that the system is in a given state, knowing the equivalence class of this state, is an invariant of the system, denoted by γ.

Theorem 3. *(Forward Markov bisimulation implies sound abstraction) Let $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ be a WLTS. If $(\sim, \sim_L)$ induces a forward Markov bisimulation, then for any aggregates $\tilde{x}_i$, $\tilde{l}$, and $\tilde{x}_j$, we can define*

$$\tilde{w}(\tilde{x}_i, \tilde{l}, \tilde{x}_j) = \delta_F(x_i, \tilde{l}, \tilde{x}_j).$$

The so defined abstraction $\tilde{\mathcal{W}} = (\mathcal{X}_{/\sim}, \mathcal{L}_{/\sim_L}, \tilde{w}, \tilde{\pi_0})$ is sound. We then say that $\mathcal{W}$ refines $\tilde{\mathcal{W}}$ by a forward Markov bisimulation $(\sim, \sim_L)$, written $\mathcal{W} \preceq_{F,(\sim,\sim_L)} \tilde{\mathcal{W}}$.

Theorem 4. *(Backward Markov bisimulation implies sound and complete abstraction) Given a WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, if $(\sim, \sim_L)$ induces a backward Markov bisimulation with conditional probabilities over the aggregates $\gamma \in \Gamma_{\mathcal{X},\sim}$, then for any aggregates $\tilde{x}_i$,$\tilde{l}$, and $\tilde{x}_j$, we can define*

$$\tilde{w}(\tilde{x}_i, \tilde{l}, \tilde{x}_j) = \frac{\delta_B(\tilde{x}_i, \tilde{l}, x_j)}{\gamma(\tilde{x}_j, x_j)}. \tag{4}$$

If $\gamma(\tilde{x}) = \pi_0|_{\tilde{x}}$, then the so defined abstraction $\tilde{\mathcal{W}} = (\mathcal{X}_{/\sim}, \mathcal{L}_{/\sim_L}, \tilde{w}, \tilde{\pi_0})$ is sound and complete. We then say that $\mathcal{W}$ refines $\tilde{\mathcal{W}}$ by a backward Markov bisimulation $(\sim, \sim_L)$ with conditional distributions γ, written $\mathcal{W} \preceq_{B,(\sim,\sim_L),\gamma} \tilde{\mathcal{W}}$.

In particular, if we know that γ is uniform, if follows from the equation (4) that $\tilde{w}(\tilde{x}_i, \tilde{l}, \tilde{x}_j) = \delta_{BU}(x_i, \tilde{l}, \tilde{x}_j)$, written also $\mathcal{W} \preceq_{BU,(\sim,\sim_L)} \tilde{\mathcal{W}}$.

Example 6.3. *Now we illustrate the difference between forward and backward bisimulations in Fig. 14.*

In Fig. 14(a), we can notice that $a(x_1) = a(x_2) = 3$. Moreover, not only we have $\delta_F(x_1, \{a, a'\}, \{x_3\}) = \delta_F(x_2, \{a, a'\}, \{x_3\}) = 1$, but we also have $\delta_F(x_1, \{b, b'\}, \{x_4\}) = \delta_F(x_1, \{b, b'\}, \{x_4\}) = 2$. Thus, the pair $(\sim, \sim_L)$ of equivalence relations where $\sim$ identifies the states x_1 and x_2, and $\sim_L$ identifies pair-wisely the labels a and a', and b and b', is a forward bisimulation.

Moreover, the pair of relation $(\sim, \sim_L)$ is not a backward bisimulation. Otherwise, there would exist a family of distributions γ such that:

$$\gamma(\{x_1, x_2\}, x_1) \cdot \delta_B(\{x_3\}, \{c\}, x_1) = \gamma(\{x_1, x_2\}, x_2) \cdot \delta_B(\{x_3\}, \{c\}, x_2),$$
$$\gamma(\{x_1, x_2\}, x_1) \cdot \delta_B(\{x_4\}, \{d\}, x_1) = \gamma(\{x_1, x_2\}, x_2) \cdot \delta_B(\{x_4\}, \{d\}, x_2),$$

which would be absurd since $\delta_B(\{x_3\}, \{c\}, x_1) = 1$, $\delta_B(\{x_3\}, \{c\}, x_2) = 0$, $\delta_B(\{x_4\}, \{d\}, x_2) = 4$, and $\gamma(\{x_1, x_2\}, x_1) + \gamma(\{x_1, x_2\}, x_2) = 1$.

In Fig. 14(b), we can notice that $a(x_2) = a(x_3) = 2$ and $a(x_4) = a(x_5) = 2$. Moreover, we have:

$$\delta_{BU}(\{x_1\}, \{a, a'\}, x_2) = \delta_{BU}(\{x_1\}, \{a, a'\}, x_3) = 3,$$
$$\delta_{BU}(\{x_2, x_3\}, \{b, b'\}, x_4) = \delta_{BU}(\{x_2, x_3\}, \{b, b'\}, x_5) = 1,$$
$$\delta_{BU}(\{x_1\}, \{c, c'\}, x_4) = \delta_{BU}(\{x_1\}, \{c, c'\}, x_5) = 1.$$

Thus, the pair $(\sim, \sim_L)$ of equivalence relations, where $\sim$ identifies pair-wisely the states x_2 and x_3, and the states x_4 and x_5, and $\sim_L$ identifies pair-wisely the labels a and a', and the labels b and b', and c and c', is a uniform backward bisimulation. Nevertheless, $(\sim, \sim_L)$ is not a forward bisimulation, since, for instance, $\delta_F(x_4, \{d\}, \{x_1\} = 1$, whereas $\delta_F(x_5, \{d\}, \{x_1\}) = 0$.

□

6.4. Proving bisimulations

The forward bisimulation relation for abstracting the transition systems with CTMC semantics has been established and used in applications (eg, [21, 20]). Moreover, computing the backward uniform bisimulation when γ is uniform is defined in [5, 38]. It is based on an alternative characterization of the backward uniform Markov bisimulation, which eases the analysis.

Lemma 2. *(Proving backward uniform Markov bisimulation) We consider $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ a WLTS and $(\sim, \sim_L)$ be a pair of equivalence relations respectively over $\mathcal{X}$ and $\mathcal{L}$. For any state $x' \in \mathcal{X}$, and any pair of classes $\tilde{x}, \tilde{l} \in \mathcal{X}_{/\sim} \times \mathcal{L}_{/\sim_L}$, let us define the set $\operatorname{Pred}(\tilde{x}, \tilde{l}, x')$ of transitions from a state in $\tilde{x}$ to the state x' and with a label in $\tilde{l}$ as follows:*

$$\operatorname{Pred}(\tilde{x}, \tilde{l}, x') = \{(x, l) \in \tilde{x} \times \tilde{l} \mid w(x, l, x') > 0\}.$$

Assume that: (1) $\pi_0|_{\mathcal{X}_{/\sim}} = \tilde{\pi_0}$, and (2) for any $x'_i, x'_j \in \mathcal{X}$ such that $x'_i \sim x'_j$ and any $\tilde{x} \in \mathcal{X}_{/\sim}$, $\tilde{l} \in \mathcal{L}_{/\sim_L}$, there exists a bijective map ϕ between $\operatorname{Pred}(\tilde{x}, \tilde{l}, x'_i)$ and

$\mathrm{Pred}(\tilde{x}, \tilde{l}, x'_j)$, *such that for any* $(x_i, l_i) \in \mathrm{Pred}(\tilde{x}, \tilde{l}, x'_i)$, *if* $\phi(x_i, l_i) = (x_j, l_j)$, *then we have that* $w(x_i, l_i, x'_i) = w(x_j, l_j, x'_j)$.

Then we have that $\mathcal{W}$ *is the backward uniform bisimulation of the abstraction* $\tilde{\mathcal{W}} = (\mathcal{X}_{/\sim}, \mathcal{L}_{/\sim}, \tilde{w}, \tilde{\pi_0})$, *i.e.* $\mathcal{W} \preceq_{BU,(\sim,\sim_L)} \tilde{\mathcal{W}}$.

On the other hand, as soon as γ over the aggregates is not uniform, we cannot observe the bijection between predecessors over the states. Proving that the given abstraction is a backward bisimulation cannot be established unless we have a right 'guess' of the distributions γ. Lem. 3 states how to avoid proving backward bisimulation by instead proving two uniform backward bisimulations. More precisely, if we want to prove the backward refinement between the systems $\mathcal{W}$ and $\tilde{\mathcal{W}}$, it is enough to observe the system $\mathcal{W}^i$, which is a backward uniform refinement of both of the systems $\mathcal{W}$ and $\tilde{\mathcal{W}}$ (Fig. 15).

Lemma 3. *(Proving backward Markov bisimulation) Consider a WLTS* $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$, *and any aggregation relation* $(\sim, \sim_L)$, *which would satisfy* $\tilde{\mathcal{W}} = (\mathcal{X}_{/\sim}, \mathcal{L}_{/\sim_L}, \tilde{w}, \tilde{\pi}_0)$. *We assume that there exist a system* $\mathcal{W}^i = (\mathcal{X}^i, \mathcal{L}^i, w^i, \pi^i_0)$, *and two pairs of equivalence relations* $(\sim_1, \sim_{L1})$, $(\sim_2, \sim_{L2})$, *such that* $\sim_1 \preceq \sim_2$ *(in the sense that, for any* $x^i_1, x^i_2 \in \mathcal{X}^i$, $x^i_1 \sim_1 x^i_2 \Rightarrow x^i_1 \sim_2 x^i_2$*),* $\sim_{L1} \preceq \sim_{L2}$, *and, for any* x^i_1, x^i_2 *such that* $x^i_1 \sim_2 x^i_2$, *the number of states which are* $\sim_{L1}$*-equivalent to* x^i_1 *is equal to the number of states which are* $\sim_{L1}$*- equivalent to* x^i_2*),* $\mathcal{W}^i \preceq_{BU,(\sim_1,\sim_{L1})} \mathcal{W}$, *and* $\mathcal{W}^i \preceq_{BU,(\sim_2,\sim_{L2})} \tilde{\mathcal{W}}$. *Under this assumption, we have that* $\mathcal{W} \preceq_{B,(\sim,\sim_L),\gamma} \tilde{\mathcal{W}}$, *where* γ *is defined as*

$$\gamma(\tilde{x}, x) = \frac{|\{x^i \in \mathcal{X}^i \ | \ x^i \sim_1 x^i_0\}|}{|\{x^i \in \mathcal{X}^i \ | \ x^i \sim_2 x^i_0\}|}, \text{ for any } [x^i_0]_{\sim 1} = x.$$

This Lemma contains the key observation for the abstraction of Kappa systems, and for proving Thm.1. It thus completes the intention of the theoretical analysis in this paper. More precisely, we observe the WLTS $\mathcal{W}^{\mathcal{R}}$ of a given Kappa system $\mathcal{R}$, as defined in Dfn. 13 and its abstraction generated as proposed in the reduction procedure (Sect. 5, Dfn. 17). The main observation is that the system $\mathcal{W}$ is already an abstraction. More concretely, the states of $\mathcal{W}$ are multisets of species, and as such, they abstract the *individual* species. For example, a state that contains two agents of type $A(s_u)$ abstracts away the potential individual behavior of these two agents, for example $A_1(s_u)$ and $A_2(s_u)$. To show that the abstraction is sound and complete, we observe the system $\mathcal{W}^i$, which is the *individual-based* semantics of a Kappa system, where each individual agent is uniquely identified. The backward uniform refinement is established between $\mathcal{W}^i$ and $\mathcal{W}$ by the modeling assumptions. We are left to prove the backward uniform refinement between $\mathcal{W}^i$ and $\tilde{\mathcal{W}}$. This is done by inspections on the ACM's (Dfn. 15).

Example 6.4. *We consider the following Kappa system. We have the agent types* $\mathcal{A} = \{A, B\}$, *the site names* $\{s, t\}$, *the signatures* $\Sigma_\iota(A) = \Sigma_\iota(B) = \{s\}$ *and* $\Sigma_\lambda(A) = \Sigma_\lambda(B) = \{t\}$, *the alphabet of internal states* $\mathbb{I} = \{u, p\}$. *The contact map is defined by* $(\mathcal{N}, \mathcal{E})$, *such that* $\mathcal{N} = \{(A, s), (A, t), (B, s), (B, t)\}$

$$\begin{tikzcd} \mathcal{W}^i \arrow[r, "\preceq_{BU,(\sim_1,\sim_{L1})}"] \arrow[rr, bend right, "\preceq_{BU,(\sim_2,\sim_{L2})}"] & \mathcal{W} & \tilde{\mathcal{W}} & \Rightarrow & \mathcal{W} \arrow[r, "\preceq_{B,(\sim,\sim_L),\gamma}"] & \tilde{\mathcal{W}} \end{tikzcd}$$

Figure 15: Proving backward refinement (Lem. 3): instead of proving that a given abstraction is a backward bisimulation ($\mathcal{W} \preceq_{B,(\sim,\sim_L),\gamma}$), it is enough to prove two uniform backward bisimulations (find a system $\mathcal{W}^i$, such that $\mathcal{W}^i \preceq_{BU,(\sim_1,\sim_{L1})}$, and $\mathcal{W}^i \preceq_{BU,(\sim_2,\sim_{L2})}$).

and $\mathcal{E} = \{((A,t),(B,t))\}$ and the following rules:

$$\begin{aligned} &r1 : A(s_u) \leftrightarrow A(s_p)@k_1, k_{1-} \\ &r2 : B(s_u) \leftrightarrow B(s_p)@k_2, k_{2-} \\ &r3 : A(t)\ ,\ B(t) \leftrightarrow A(t^1)\ ,\ B(t^1)@k_3, k_{3-} \end{aligned}$$

Moreover, using the minimal ACM for annotating the agents, as written in Dfn. 15, we get that $\approx_A$ has two equivalence classes $\{s\}$ and $\{t\}$; and that $\approx_B$ has two equivalence classes $\{s\}$ and $\{t\}$ as well.

The fragments derived from an ACM (Dfn. 16) are the following: $F_1 = A(s_u)$, $F_2 = A(s_p)$, $F_3 = A(t)$, $F_4 = A(t^1)\ ,\ B(t^1)$, $F_5 = B(s_u)$, $F_6 = B(s_p)$, $F_7 = B(t)$.

Let us pick a (finite) initial distribution π_0. Now we observe the WLTS $\mathcal{W} = (\mathcal{X}, \mathcal{L}, w, \pi_0)$ assigned to the Kappa system $\mathcal{R}_{AB}$ (introduced in Dfn. 13), and the state y which is the $\equiv$-equivalence class of the mixture E_y defined as follows:

$$A(s_p,t^1)\ , B(s_p,t^1)\ , A(s_u,t^2)\ , B(s_u,t^2)\ , A(s_u,t^3)\ , B(s_u,t^3).$$

The unique (up to $\equiv$) non $\equiv^\sharp$-equivalent mixtures is $E_{y'}$, defined as follows:

$$A(s_p,t^1)\ , B(s_u,t^1)\ , A(s_u,t^2)\ , B(s_p,t^2)\ , A(s_u,t^3)\ , B(s_u,t^3)$$

We denote $y' = [E_{y'}]_\equiv$, $\tilde{y}' = [E_{y'}]_{\equiv^\sharp}$. We compute however that the distribution among state $\tilde{y} = [E_y]_{\equiv^\sharp}$ is such that $\gamma(\tilde{y}, y) = 1/3$, and $\gamma(\tilde{y}, y') = 2/3$. Roughly speaking, this comes from the fact that if we annotate fragments of type A and B in $\tilde{y}$ with the identifiers 1, 2, 3 (there are 36 possible annotations), and if we assume that agents with the same identifiers are bound together, then there are 12 annotations such that the phosphorilated A and B are bound together, and 24 where this is not the case. A more detailed analysis of this model is given in [15].

□

7. Conclusions

Reducing the complexity of combinatorial reaction mixtures is an important milestone towards simulation and analysis of large-scale realistic models of cellular signal transduction. In this paper we study a scalable reduction method, that is applicable to any rule-based specification. The reduction is sound and moreover complete, i.e. the sample traces of individual molecular species can be reconstructed from the traces of aggregated species in the reduced model. We put this method into the general context of abstractions of probabilistic transition systems and show that it yields a sufficient condition for weak lumpability and that it is equivalent to backward Markov bisimulation. The reduction factor depends on the number of independent molecular events and is strictly smaller than that of the less-demanding reduction based on the differential semantics.

A compelling problem for future work is thus to analyze differential fragments in the context of stochastic semantics and to obtain error bounds for this reduction as a function of the kinetic parameters of the system.

Acknowledgements

We would like to thank the anonymous reviewers for their comments on the different versions of the paper. We would also like to thank Ferdinanda Camporesi for her careful reading and the useful insights that she gave us about the paper.

References

[1] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):2003, 2003.

[2] Christel Baier and Holger Hermanns. Weak bisimulation for fully probabilistic processes. In *CAV'97*, pages 119–130, Haifa, Israel, 1997. Springer.

[3] Michael L. Blinov, James R. Faeder, and William S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20:3289–3292, 2004.

[4] Peter Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31(1):59–75, 1994.

[5] Peter Buchholz. Bisimulation relations for weighted automata. *Theoretical Computer Science*, 393(1-3):109–123, 2008.

[6] Ferdinanda Camporesi, Jérôme Feret, Heinz Koeppl, and Tatjana Petrov. Combining model reductions. In *MFPS XXVI*, volume 265 of *Electonic Notes in Theoretical Computer Science*, pages 73–96, Ottawa, Canada, 2010. Elsevier.

[7] Holger Conzelmann, Dirk Fey, and Ernst D. Gilles. Exact model reduction of combinatorial reaction. *BMC Systems Biology*, 2(78):342–351, 2008.

[8] Vincent Danos, Jérôme Feret, Walter Fontana, Russel Harmer, and Jean Krivine. Rule-based modelling, symmetries, refinements. In *FMSB'08*, volume 5054 of *Lecture Notes in Bioinformatics*, pages 103–122, Cambridge, UK, 2008. Springer.

[9] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks, invited paper. In *APLAS'07*, volume 4807 of *Lecture Notes in Computer Science*, pages 139–157, Singapore, 2007. Springer.

[10] Vincent Danos and Cosimo Laneve. Core formal molecular biology. *Theoretical Computer Science*, 325:69–110, 2003.

[11] Josée Desharnais, Richard Blute, Abbas Edalat, and Prakash Panagaden. Bisimulation for labelled markov processes. In *Information and Computation*, pages 95–106, 1997.

[12] Laurent Doyen, Thomas A. Henzinger, and Jean-Francois Raskin. Equivalence of labeled Markov chains. *International Journal of Foundations of Computer Science*, 19(3):549–563, 2008.

[13] Jérôme Feret. Fragements-based model reduction: some case studies. In *CS2Bio'10*, volume 265 of *Electonic Notes in Theoretical Computer Science*, pages 73–96, Amsterdam, Netherlands, 2010. Elsevier.

[14] Jérôme Feret, Vincent Danos, Jean Krivine, Russ Harmer, and Walter Fontana. Internal coarse-graining of molecular systems. *Proceedings of the National Academy of Sciences of the United States of America*, 106(16):6453–6458, April 2009.

[15] Jérôme Feret, Heinz Koeppl, and Tatjana Petrov. Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models. *International Journal of Software and Informatics*, to appear.

[16] Gnuplot. www.gnuplot.info.

[17] Donald Gross and Douglas R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.

[18] Cato Maxilian Guldberg and Peter Waage. Concerning chemical affinity. *Erdmann's Journal für Praktische Chemie*, 127:69–114, 1879.

[19] Thomas A. Henzinger, Maria Mateescu, and Verena Wolf. Sliding window abstraction for infinite Markov chains. In *CAV'09*, volume 5643 of *Lecture Notes in Computer Science*, pages 337–352, Grenoble, France, 2009. Springer.

[20] Holger Hermanns. *Interactive Markov Chains And the Quest for Quantified Quality*. PhD thesis, University of Marburg, 2002.

[21] Jane Hillston. *A compositional approach to performance modelling.* Cambridge University Press, New York, NY, USA, 1996.

[22] William S. Hlavacek, James R. Faeder, Michael L. Blinov, Alan S. Perelson, and Byron Goldstein. The complexity of complexes in signal transduction. *Biotechnology Bioengineering*, 84:783–794, 2005.

[23] William S. Hlavacek, James R. Faeder, Michael L. Blinov, Richard G. Posner, Michael Hucka, and Walter Fontana. Rules for Modeling Signal-Transduction Systems. *Science's STKE*, 2006(344), 2006.

[24] D. Kannan Jianjun Paul Tian. Lumpability and commutativity of Markov processes. *Stochastic Analysis and Applications*, 24(3):685–702, 2006.

[25] John G. Kemeny and James L. Snell. *Finite Markov Chains.* Van Nostrand, 1960.

[26] John G. Kemeny, James L. Snell, and Anthony W. Knapp. *Denumerable Markov Chains.* Springer-Verlag, New York, NY, USA, 1976.

[27] Thomas G. Kurtz. Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *Journal of Applied Probability*, 8(2):344–356, 1971.

[28] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing (preliminary report). In *POPL'89*, pages 344–352, New York, NY, USA, 1989. ACM.

[29] MAPLE. www.maplesoft.org.

[30] Donald A. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4(3):413–478, 1967.

[31] Elaine Murphy, Vincent Danos, Jerome Feret, Russell Harmer, and Jean Krivine. Rule based modelling and model refinement. In *Elements of Computational Systems Biology*. Wiley, 2009.

[32] OPENKAPPA. www.kappalanguage.org.

[33] Gerardo Rubino and Bruno Sericola. A finite characterization of weak lumpable Markov processes. part I: The discrete time case. *Stochastic processes and their applications*, 38:195–204, 1991.

[34] Gerardo Rubino and Bruno Sericola. A finite characterization of weak lumpable Markov Processes. part II: The continuous time case. *Stochastic processes and their applications*, 45:115–125, 1993.

[35] Walter Rudin. *Real and complex analysis, 3rd ed.* McGraw-Hill, Inc., New York, NY, USA, 1987.

[36] Roger B. Sidje, Kevin Burrage, and Shev MacNamara. Inexact uniformization method for computing transient distributions of Markov chains. *SIAM Journal on Scientific Computing*, 29, issue 6:2562–2580, 2007.

[37] Ana Sokolova and Erik P. de Vink. On relational properties of lumpability. In *PROGRESS'03*, pages 220–224, Utrecht, Netherlands, 2003.

[38] Jeremy Sproston and Susanna Donatelli. Backward stochastic bisimulation in CSL model checking. In *QEST '04*, pages 220–229, Washington, DC, USA, 2004.

[39] Christopher T. Walsh. *Posttranslation Modification of Proteins: Expanding Nature's Inventory.* Roberts and Co. Publisher, 2006.

Jérôme Feret is a Research Scientist at INRIA Paris-Rocquencourt, since October 2008, within the project "Abstraction" which is hosted at the École normale supérieure. He defended his PhD on the abstract interpretation of mobile systems in 2005. Since 2006, he has been working on static analysis for rules-based languages. In 2007-2008, he was a Post-doctoral Research Fellow in Harvard Medical School in Walter Fontana's lab. In 2009, he was awarded the ANR Junior Chair of Excellence "AbstractCell" so as to keep on his studies on the abstraction of quantitative semantics for rules-based models.

Thomas A. Henzinger is President of IST Austria (Institute of Science and Technology Austria) and Adjunct Professor of Electrical Engineering and Computer Sciences at the University of California, Berkeley. He holds a Ph.D. degree in Computer Science from Stanford University (1991). His research focuses on modern systems theory, especially models, algorithms, and tools for the design and verification of reliable software, hardware, and embedded systems. His HyTech tool was the first model checker for mixed discrete-continuous systems. He is the recipient of an ERC Advanced Investigator Grant.

Heinz Koeppl is Assistant Professor in the Automatic Control Laboratory of the ETH Zurich. He got his Ph.D. in electrical engineering from Graz University of Technology, Austria (2004). He spent a PostDoc year at UC Berkeley, with an Erwin Schroedinger postdoctoral fellowship from the Austrian Science Foundation. In spring 2008, he was a research fellow in Harvard Medical School working together with Prof. Fontana. His goal is to advance design and analysis methods for bio-molecular circuits. This includes new modeling approaches like rule-based or agent-based models, new analysis methods from control theory.

Tatjana Petrov is a PhD student in Computer Science at the Automatic Control Laboratory of the ETH Zurich, supervised by Heinz Koeppl. She got her Diploma and a M. Sc. degree in Theoretical Computer Science from the Faculty of Sciences, Department for Mathematics and Informatics, University of Novi Sad (2007). Her research interest is using formal techniques for modeling highly complex biological systems. In particular, the mission of Tatjana's PhD is to develop techniques for automatic reductions of stochastic semantics of a set of biological hypothesis, that are summarized in a rule-based program.