Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 272 (2011) 3–17
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Stochastic Semantics of Signaling as a Composition of Agent-view Automata

Heinz Koeppl[2] Tatjana Petrov[3]

School of Communication and Computer Sciences
Ecole Polytechnique Federale de Lausanne (EPFL)
Lausanne, Switzerland

Abstract

In this paper we present a formalism based on stochastic automata to describe the stochastic dynamics of signal transduction networks that are specified by rule-sets. Our formalism gives a modular description of the underlying stochastic process, in the sense that it is a composition of smaller units, *agent-views*. The view of an agent is an automaton that identifies all local modification changes of that agent (internal state modifications, binding and unbinding), but also those of interacting agents, which are tested within the same rule. We show how to represent the generator matrix of the underlying Markov process of the whole rule-set as Kronecker sums of the rate matrices belonging to individual view-automata. In the absence of birth the automata are finite, since the number of different contexts in which one agent can appear in a rule-set is finite. We illustrate the framework by an example that is related to cellular signaling events.

Keywords: Cell signaling, Continuous-time Markov chain, Stochastic automata composition

1 Introduction

Internal dependencies of multi-site posttranslational modifications [21,17] and conformational changes [4,18] of signaling proteins, reflect the rich internal logic of proteins and invite the formalization of this logic through an agent automaton. Consider for instance the protein interaction network driving circadian oscillations in cyanobacteria. The central hexameric KaiC protein undergoes cycles of hypo-phosphorylated and hyper-phosphorylated states [14,13], where the sequence of phosphorylation of the two residues of every protein subunit is strictly controlled [15]. Moreover, it is believed that the KaiC hexamer changes conformation upon

[1] Heinz Koeppl acknowledges the support from the Swiss National Science Foundation, grant no. 200020-117975/1. Tatjana Petrov acknowledges the support from SystemsX.ch, the Swiss Initiative in Systems Biology.

[2] Email: heinz.koeppl@epfl.ch
[3] Email: tatjana.petrov@epfl.ch

doi:10.1016/j.entcs.2011.04.002

hyper-phosphorylation. See Fig. 1 for a schematic of the cyclic process that is controlled by two modulator proteins KaiA and KaiB. Such modification events are uni-molecular events and can thus be well encapsulated into an internal logic of a protein. Bi-molecular events, such as modulator binding, can be considered as inputs to this state automata. The construction of individual protein-automata also

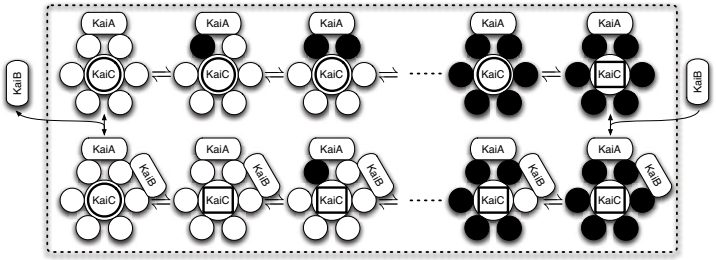
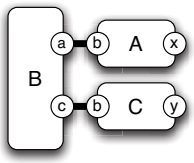


Fig. 1. Internal logic of a multimeric protein. The simplified scheme captures the basic cyclic transitions that the hexameric circadian clock protein KaiC undergoes. Hyperphosporylation (black-filled subunits) induces a conformational change from an allosteric tensed (◯) to a relaxed (□) state – as for instance proposed in [20]. Binding of modulator proteins can be considered as input to this state automaton.

holds promise to directly uncover the effective degrees of freedom of the interacting protein ensemble. Recently, much progress has been made to determine the effective state-space dimension and the corresponding generalized states of such ensembles. The thread started with [2,5], where a linear projection of the species-based state-space is constructed, allowing for a self-consistent description of the dynamics on a lower dimensional state-space. The generalization of this approach to the automatic reduction of the differential semantics of any rule-based specification is done in [8]. The accompanying stochastic version of this reduction is given in [10]. All these approaches have in common that they start out by a description of the concrete, large state-space, which is then reduced through projection or aggregation methods. In our case, the description already is given in a symbolic, implicit form. We take a bottom-up approach and observe the effective degrees of freedom of each agent and construct its local state-space accordingly. Taking an agent-centric perspective the degrees of freedom are all the different contexts the agent is involved in – *agent views* (although other definitions of views are available [6]). Thus, besides the above agent-centric modularization that encapsulates the agent's internal logic, the approach yields a direct constructing of the reduced state-space. Consider the example shown in Fig. 2 that conveys the basic idea. It involves a scaffold protein that can simultaneously and independently bind two other proteins. Considering the rules in Fig. 2 we can determine what contexts the agent $\mathtt{A}$ encounters. Its views give rise to the set of states $\{\mathtt{A}(\mathtt{b}), \mathtt{A}(\mathtt{b}^{\mathtt{a.B}})\} \times \{\mathtt{A}(\mathtt{x_u}), \mathtt{A}(\mathtt{x_p})\}$. We represent its views using a stochastic automaton and then couple the view-automata of different agents to automata network [16]. Such networks can sometimes be cast into a representation as superposed generalized stochastic Petri-nets (GSPN) [12] - a collection of Petri-subnets that share transitions but no places. The case of example Fig. 2 is illustrated in Fig. 3, where the stochastic automaton is shown for the case of a single copy number per agent and the Petri-net representation for an arbitrary marking is given. We recognize that due to the independence between binding and



$$R_1: \quad \mathtt{A(b)}, \mathtt{B(a)} \rightleftharpoons \mathtt{A(b^1)}, \mathtt{B(a^1)}$$
$$R_2: \quad \mathtt{C(b)}, \mathtt{B(c)} \rightleftharpoons \mathtt{C(b^1)}, \mathtt{B(c^1)}$$
$$R_3: \quad \mathtt{A(x_u)} \rightleftharpoons \mathtt{A(x_p)}$$
$$R_3: \quad \mathtt{C(y_u)} \rightleftharpoons \mathtt{C(y_p)}$$

Fig. 2. Scaffold protein B recruits independently the proteins A and C (left). For the sake of illustration we assume that the latter two are phosphorylated and dephosphorylated spontaneously. Kappa syntax [7] to express this interactions (right).

modification the view-automaton in Fig. 3 can be constructed as automata product of two smaller automata obtained from their respective rules R_1 and R_3. In this example, the view-automata states are equivalent to the states expressed in fragments obtained by [10]. Mapping a rule-based specification to a network of stochastic au-

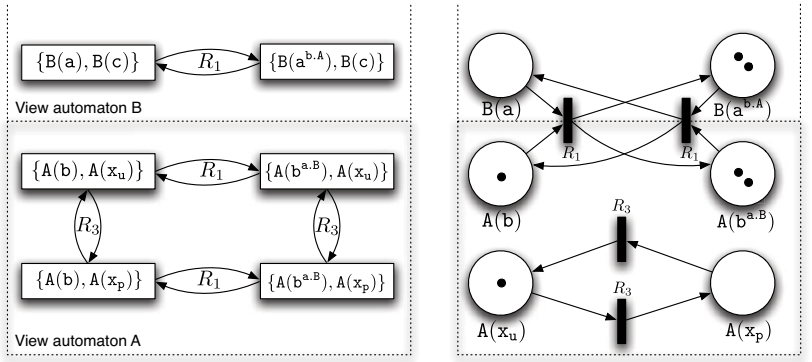


Fig. 3. Stochastic view-automata network for agent A and partially shown for agent B (left); drawn for one copy number per agent. Only transitions with the same label across different automata are synchronized.Corresponding Petri-net representation where view-nets share transitions but no places.

tomata allows one to use the compositional methods developed for such networks [3]. Furthermore, we can exploit the compositional structure to obtain an expression for the generator matrix of the network's continuous-time Markov chain involving the Kronecker sums of the generator matrices of the individual automata [16,3]. The first use of stochastic automata networks to describe stochastic chemical kinetics can be found in [22]. The work considers a species-based state-space and associates a counter automaton with each species.

The remaining part of the work is organized as follows. In Section 2, the site graphs and their encoding as a valuation over a set of Boolean variables is presented. The encoding is inspired by how the site-graphs are defined in Kappa [7]. Section 3 continues the formalism by defining a rule and a rule-based system. Each rule-based system accompanied with the initial conditions is assigned the (continuous-time) stochastic semantics by the interpreted labelled transition system (referred to as ILTS from now on). Furthermore, agent-view and population-view projections are defined.

The main result is stated in Section 4, where we propose when and how the ILTS of a rule-based system can be represented as a composition of smaller ILTS,

each corresponding to a subset of rules. The decomposition criterion is derived by analysing the set of variables that appear in each rule. Based on that agent-centred compositional approach Section 5 makes use of the explicit construction of the Markov chain generator available for stochastic automata networks. The procedure is outlined using the simple scaffolding example of Fig. 2. Conclusions are drawn in Section 6.

2 A simple agent-based framework

We build a formalism on the rule-based language Kappa [7]. The main data structure which we use to describe the structure of the protein network, and to encode the reaction mixture are *site graphs*. Whereas standard graphs are a pair structure defined by a set of nodes and a set of edges formed over pairs of nodes, site-graphs have a slightly richer structure: each node is defined by (i) its name, (ii) a set of sites with internal state, and (iii) a set of binding sites of that node; The edges are then established, not between the node names, but between a pair of binding sites, each belonging to a different node.

Definition 2.1 *(Site graph)* Consider a set of agent names $\mathcal{A}$ and a set of site names $\mathcal{S}$. Site graph is a pair $G = (\mathcal{V}, \mathcal{E})$ where the set of nodes are triples of an agent name, the set of its internal and the set of its binding sites, ie

$$\mathcal{V} \subseteq \{(A, \Sigma_{int}, \Sigma_l) \mid A \in \mathcal{A}; \Sigma_{int}, \Sigma_l \subseteq 2^{\mathcal{S}}\},$$

and edges are pairs of sites:

$$\mathcal{E} \subseteq \{((A, s), (A', s')) \mid (A, \Sigma_{int}, \Sigma_l), (A', \Sigma_{int}{}', \Sigma_l') \in \mathcal{V}, s \in \Sigma_l, s' \in \Sigma_l'\}.$$

Having a node $(A, \Sigma_{int}, \Sigma_l)$, the collection of sites of the agent A, ie $\Sigma_{int} \cup \Sigma_l$, is sometimes referred to as the *interface* of agent A, and is denoted $\Sigma(A)$. When we model the protein interaction network with a site graph, a set of agents $\mathcal{A}$ represents a set of protein names and a set of sites $\mathcal{S}$ denotes the different relevant amino acid residues of the protein. The site graph which summarizes the protein names and their possible bindings in a model of a protein network we call a *contact map* (CM in further text).

Example 2.2 (Fig.2 revisited) The contact map is a site graph $(\mathcal{V}, \mathcal{E})$ with agent names $\mathcal{A} = \{A, B, C\}$ and site names $\mathcal{S} = \{a, b, c, x, y\}$; Set of nodes is $\mathcal{V} = \{(A, \{x\}, \{b\}), (B, \emptyset, \{a, c\}), (C, \{y\}, \{b\})\}$, and edges $\mathcal{E} = \{((A, b), (B, a)), ((B, c), (C, b))\}$.

Moreover, given the contact map and agents' multiplicities $n : \mathcal{A} \to \mathbb{N}_0$, we define the *full contact map* as a site graph where each agent name $A \in \mathcal{A}$ is instantiated $n(A)$ times, so that each copy of the agent is identified by a number in its subscript – copies are assigned names $A_1, \ldots, A_{n(A)}$. Bonds are generated between any (A_i, s) and (A_j', s') such that the bond existed between (A, s) and (A', s') in the contact map. Formally, a full CM over a CM $(\mathcal{V}, \mathcal{E})$ with agent names $\mathcal{A}$ and $\mathcal{S}$ is a site

graph $(\mathcal{V}', \mathcal{E}')$ with agent names $\mathcal{A}'$ and site names $\mathcal{S}$, such that $\mathcal{A}' = \{A_i \mid A \in \mathcal{A}, i = 1, .., n(A)\}$, and

$$\text{if } (A, \Sigma_{int}, \Sigma_l) \in \mathcal{V}, \text{ then } (A_i, \Sigma_{int}, \Sigma_l) \in \mathcal{V}', \text{ for } i = 1, \ldots, n(A)$$

and the set of edges $\mathcal{E}'$ is such that if $((A, s), (A', s')) \in \mathcal{E}$, then

$$((A_i, s), (A'_j, s)) \in \mathcal{E}' \text{ for all } i = 1, \ldots, n(A),\ j = 1, \ldots, n(A').$$

Example 2.3 (Fig.2 revisited) For $n(A) = 1$, $n(B) = 2$, $n(C) = 1$, we get the full contact map $(\mathcal{V}', \mathcal{E}')$, where $\mathcal{A}' = \{A_1, B_1, B_2, C_1\}$, and $\mathcal{V}' = \{(A_1, \{x\}, \{b\}), (B_1, \emptyset, \{a, c\}), (B_2, \emptyset, \{a, c\}), (C_1, \{y\}, \{b\})\}$, and $\mathcal{E}' = \{((A_1, b), (B_1, a)), ((A_1, b), (B_2, a)), ((B_1, c), (C_1, b))((B_2, c), (C_1, b))\}$.

If we model a protein interaction network, we need to represent a reaction mixture at a certain time point. A full contact map is a summary of which sites appear on which agent, but it does not tell us what is the value of the internal state; Moreover, the bonds specified in the site graph are potentially formed, but they may or may not exist in a reaction mixture. In other words, given a site graph, there are several mixtures which correspond to that site graph, depending on the internal states of internal sites, and depending on which bonds are present in the mixture. For simplicity we assume that the internal states can take exactly two values and we assign a set of Boolean variables to a full contact map, such that one valuation of these variables encodes a reaction mixture. One variable is spent per each agent's site, and one variable is spent per each edge:

$$Var_{(\mathcal{V},\mathcal{E})} \cong \{(A, s) \mid (A, \Sigma_{int}, \Sigma_l) \in \mathcal{V} \text{ and } s \in \Sigma_{int} \cup \Sigma_l\} \cup \mathcal{E}.$$

Each of the site variables is represented by a letter a with the corresponding agent–site name combination in its subscript. We use letter b indexed by the bond description for the binding variables. The set of variables which refer to agent $A \in \mathcal{A}$ we denote by Var_A. Any valuation of the variables from the set $Var_{(\mathcal{V}',\mathcal{E}')}$ to Boolean values sets the internal states of agents to a value 'on' or 'off', and the bond variables respectively.

Given the full CM $(\mathcal{V}', \mathcal{E}')$ which is derived from the CM $(\mathcal{V}, \mathcal{E})$, and agents' multiplicities $n : \mathcal{A} \to \mathbb{N}_0$, we observe the set of variables $Var_{(\mathcal{V}',\mathcal{E}')}$ and the valuations

$$Val_{(\mathcal{V}',\mathcal{E}')} = \{\mathbf{x} \mid \mathbf{x} : Var_{(\mathcal{V}',\mathcal{E}')} \to \{0, 1\}\}.$$

Example 2.4 (Ex.2 revisited). Let us set $n(A) = 1$, $n(B) = 2$ and $n(C) = 1$. We have that

$$\begin{aligned} Var = \{&a_{(A_1,x)}, a_{(A_1,b)}, a_{(B_1,a)}, a_{(B_1,c)}, a_{(B_2,a)}, a_{(B_2,c)}, a_{(C_1,b)}, a_{(C_1,y)}, \\ &b_{((A_1,b),(B_1,a))}, b_{((A_1,b),(B_2,a))}, b_{((B_1,c),(C_1,b))}, b_{((B_2,c),(C_1,b))}\}. \end{aligned}$$

The state $\mathbf{x}_1 = (0, 1, 1, 1, 0, 0, 1, 1; 1, 0, 1, 0)$ represents the mixture shown in Fig. 5b).

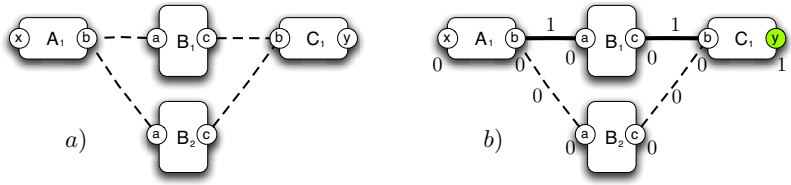


Fig. 4. a) A full contact map for Ex.2 and agent multiplicities $n(A) = 1$, $n(B) = 2$, and $n(C) = 1$; b) One reaction mixture corresponding to the state $\mathbf{x}_1 = (0,1,1,1,0,0,1,1;1,0,1,0) \in Val$. An internal state being set to 1 (ie $\mathbf{x}(b_{(C_1,y)}) = 1$) is marked by highlighting the circle which represents this internal state in green colour.

However, not all valuations will describe one valid reaction mixture. Firstly, there can be no two bonds stemming from one site of identified agent's site: for any node $(A_i, \Sigma_{int}, \Sigma_l) \in \mathcal{V}'$, and its binding site $s \in \Sigma_l$, there can be at maximum one bond established from the site (A_i, s). Secondly, the existence of the bond, let's say $((A_i, b), (B_j, a)) \in \mathcal{E}$ will be reflected in the valuation doubly: the variable $b_{((A_i,b),(B_j,a))}$ will be set to 1, but as well, the variables $a_{(A_i,b)}$ and $a_{(B_j,a)}$ will be set to 1. The valuations which describe one valid reaction mixture we call 'well-defined'.

Definition 2.5 *(Well-defined valuation)* The valuation $\mathbf{x} \in Val$ is well-defined if

- $\prod_{A_i \in \mathcal{A}', s \in \mathcal{S}} |\{(A_i, s) \text{ such that } \sum_{A' \in \mathcal{A}, s' \in \mathcal{S}} \mathbf{x}(b_{((A_i,s),(A'_j,s'))}) = 1\}| \leq 1$, and
- $\mathbf{x}(b_{((A,s),(A',s'))}) = 1$ if and only if $\mathbf{x}(a_{(A,s)}) = 1$ and $\mathbf{x}(a_{(A',s)}) = 1$.

Example 2.6 (Ex.2 revisited) The valuation $\mathbf{x}_2 = (0,1,1,1,0,0,1,1,1,0,1,1)$ is not well-defined because $b_{((B_2,c),(C_1,b))} = 1$, but $a_{(B_2,c)} = 0$. Moreover, the valuation $\mathbf{x}_3 = (0,1,1,1,0,1,1,1,1,0,1,1)$ is neither well-defined because there are two bonds stemming from the site (C_1, b).

One may wonder why we encode each bond two times, in the sense that the existence of the bond $((A_i, s), (A'_j, s'))$ can be concluded from $a_{(A_i,s)} = a_{(A'_j,s')} = 1$. Let us go back to the Ex. 2, and assume that we have two copies of agent A, and two copies of agent B, ie $n(A) = n(B) = 2$, and that there are all bound, ie $\mathbf{x}(a_{(A_1,b)}) = \mathbf{x}(a_{(A_2,b)}) = \mathbf{x}(a_{(B_1,a)}) = \mathbf{x}(a_{(B_2,a)}) = 1$. However, we may have either bonds between A_1, B_1 and A_2 and B_2, or between A_1, B_2, and A_2, B_1. We use the bond variables $b_{((A_1,b),(B_1,a))}, \ldots$, to avoid this ambiguity.

3 Rule-based model

The transformation kernel for the ensemble of agents that we observe is defined by a set of *rules*. A rule is defined over the set of variables which correspond to a contact map $(\mathcal{V}, \mathcal{E})$, ie $Var_{(\mathcal{V},\mathcal{E})}$, and it consists of the left-hand-side (lhs in further text) and the right-hand-side (rhs in further text), which are propositional formulae over the variables from the set $Var_{(\mathcal{V},\mathcal{E})}$. We will think of a rule in the following way: the left-hand-side of the rule, α, defines the precondition for the event to occur. The right-hand-side, α_d, defines an update of the valuation, which is a finite composition of the following atomic operations: (i) 'switch' of an internal state variable, ie

$\alpha \equiv \neg a_{(A,s)}$ and $\alpha_d \equiv a_{(A,s)}$, (ii) change of a pair of variables from free to bound state or vice versa (binding/unbinding), ie $\alpha \equiv \neg a_{(A,s)} \wedge \neg a_{(A',s')} \wedge \neg b_{((A,s),(A',s'))}$ and $\alpha_d \equiv a_{(A,s)} \wedge a_{(A',s')} \wedge b_{((A,s),(A',s'))}$. We restrict to the case where there is no birth, nor deletion of an agent. We also assume that there is at maximum one occurence of the same agent name in a rule. Note that both these constraints are a restriction with respect to the Kappa language – Kappa does support more occurences of the same agent name in one rule. The set of variables appearing in rule R, we denote by Var_R.

Definition 3.1 *(Rule)* Consider the set of propositional formulae $\mathcal{P}$ over variables $Var_{(\mathcal{V},\mathcal{E})}$ (denoted also $\mathcal{P}_{(\mathcal{V},\mathcal{E})}$), generated by the grammar $p \equiv \mathbf{0} \mid \mathbf{1} \mid a \in Var_{(\mathcal{V},\mathcal{E})} \mid \neg p \mid p \wedge p$. We denote by Var_p the set of variables that occur in proposition p, and the *satisfaction region* of formula p by $[\![p]\!] = \{\mathbf{x} \mid \mathbf{x} \models p\}$. A *rule* is a triple $(\alpha, \alpha_d; k) \in \mathcal{P} \times \mathcal{P} \times \mathbb{R}_0$, such that $Var_\alpha = Var_{\alpha_d}$.

We remark that the rules are defined over the contact map, and the agents' multiplicities are not mentioned. We observe the set of variables $Var_{(\mathcal{V}',\mathcal{E}')}$ over the full CM $(\mathcal{V}', \mathcal{E}')$ which is derived from the CM $(\mathcal{V}, \mathcal{E})$, and agents' multiplicities $n : \mathcal{A} \to \mathbb{N}_0$. Each *rule* over the variables $Var_{(\mathcal{V},\mathcal{E})}$ generates a set of rules over the variables $Var_{(\mathcal{V}',\mathcal{E}')}$, where the agents' identifiers are specified: instead of a single rule R, we observe a family of rules $\{R^{id_A=i}\}_{A\in\mathcal{A};i\in\{1,\ldots,n(A)\}}$, where each agent A is assigned a unique identifier $id_A \in \{1, \ldots, n(A)\}$. Such set of rules we call *identified rules*, and we denote $\mathcal{R}^{id}$.

Example 3.2 (Ex.2 revisited). The rules described in Fig.5 rewritten in this framework are

$$\begin{aligned}
(R1)\ \neg a_{(A,b)}, \neg a_{(B,a)}, \neg b_{(A,b),(B,a)} &\to a_{(A,b)}, a_{(B,a)}, b_{(A,b),(B,a)}\\
(R2)\ \neg a_{(C,b)}, \neg a_{(B,c)}, \neg b_{(C,b),(B,c)} &\to a_{(C,b)}, a_{(B,c)}, b_{(C,b),(B,c)}\\
(R3)\ \neg a_{(A,x)} &\to a_{(A,x)}\\
(R4)\ \neg a_{(C,y)} &\to a_{(C,y)},
\end{aligned}$$

where we write a rule $r = (\alpha, \alpha_d; k)$ in the form $\alpha \to \alpha_d$ (we do not write rates where it is not necessary for the illustration purpose). Setting the agent multiplicities on $n(A) = 1$, $n(B) = 2$ and $n(C) = 1$, the rule $(R1)$ has the following two instantiations:

$$\begin{aligned}
(R1^{id_A=1,id_B=1})\ \neg a_{(A_1,b)}, \neg a_{(B_1,a)}, \neg b_{(A_1,b),(B_1,a)} &\to a_{(A_1,b)}, a_{(B_1,a)}, b_{(A_1,b),(B_1,a)}\\
(R1^{id_A=1,id_B=2})\ \neg a_{(A_1,b)}, \neg a_{(B_2,a)}, \neg b_{(A_1,b),(B_2,a)} &\to a_{(A_1,b)}, a_{(B_2,a)}, b_{(A_1,b),(B_2,a)}.
\end{aligned}$$

Definition 3.3 *(Rule-based system)* A rule-based system $\mathcal{B} = (\mathcal{V}, \mathcal{E}, n, \mathcal{R}, p_0)$ over the set of agents $\mathcal{A}$ and set of sites $\mathcal{S}$ is defined by (i) a full contact map $(\mathcal{V}', \mathcal{E}')$ over the contact map $(\mathcal{V}, \mathcal{E})$ and initial agent multiplicities $n : \mathcal{A} \to \mathbb{N}_0$, (iii) a set of rules $\mathcal{R} = \{R_1, \ldots, R_m\}$ defined over the contact map $(\mathcal{V}, \mathcal{E})$, (iv) an initial mixture expressed by the proposition $p_0 \in \mathcal{P}_{(\mathcal{V},\mathcal{E})}$. A set of rules is *well-defined* if each of the rules is well-defined.

We will define the semantics of a rule-based system by the transition system with a countable state space. Each state is assigned one or several reaction mixtures, expressed by a propositional formula over variables $Var_{(\mathcal{V}',\mathcal{E}')}$; Transitions are labelled by the name of the rule which defines it.

Definition 3.4 *(Labelled transition system)* A labelled transition system is a tuple $M = (S, L, \delta, S_0)$, where

- S is a set of states,
- L is a set of labels,
- $\delta : S \times L \to S$ is a transition function that maps a state and a label to another state,
- $S_0 \subseteq S$ is the set of initial states,

A *trace* of M of length k is a sequence $s_0 \overset{l_1,t_1}{\to} s_1 \to \ldots. \to s_{k-1} \overset{l_k,t_k}{\to} s_k \in S \times (L \times \mathbb{R} \times S)^k$, such that $\delta(s_{j-1}, l_j) = s_j$, $j = 0, 1, \ldots, k$ and $s_0 \in S_0$.

Definition 3.5 *(Interpreted labelled transition system – ILTS)* Given a labelled transition system $M = (S, L, \delta, S_0)$, a set of variables Var, and set Val of well-defined valuations over these variables, each state is interpreted by a set of valuations, given by $\mathcal{L} : S \to 2^{Val}$. Such a system we call an *interpreted LTS*, and we denote by $M_{\mathcal{L}}$. We say that the ILTS $M_{\mathcal{L}}$ is *well-defined*, if for all $s, s' \in S$ we have that $\mathcal{L}(s) \cap \mathcal{L}(s') = \emptyset$, i.e. the valuation sets assigned to different states must be disjoint.

The cylinder of traces $\mathbf{r} = s_0 \overset{l_1,I_1}{\to} s_1 \to \ldots. \to s_{k-1} \overset{l_k,I_k}{\to} s_k \in S \times (L \times \mathbb{IR} \times S)^k$ denotes a set of all traces which start by the given sequence of k transitions, and each transition happens within the interval of time indicated on the arrow. The initial distribution is such that, if $s \in S_0$, then $\pi_0(s) = \frac{1}{|S_0|}$ (we use notation $|\cdot|$ to denote the cardinality of a set), and otherwise $\pi_0(s) = 0$. The probability of the cylinder of traces $\mathbf{r}$ is given by the expression

$$\pi(\mathbf{r}) = \pi_0(s_0) \cdot \prod_{j=1}^{k} \frac{a(s_{j-1}, l_j, s_j)}{a(s_{j-1})} \cdot \left(e^{-a(s_{j-1}) \cdot \inf(I_i)} - e^{-a(s_{j-1}) \cdot \sup(I_i)} \right),$$

where $a(s_{j-1}, l_j, s_j)$ is the activity of the transition from state s_{j-1} to state s_j via label l_j and within I_j interval of time, which will be specified depending on the set of rules which the ILTS models.. The total activity of state s_{j-1} is a sum $a(s_{j-1}) = \sum\{a(s_{j-1}, l_j, s_j) \mid l_j \in L, s_j \in S\}$.

Given a rule-based system $\mathcal{B}$, we interpret its semantics by assigning it the ILTS $M_{\mathcal{L}}$. Then we say that $M_{\mathcal{L}}$ *models* the rule-based system, written $M_{\mathcal{L}} \models \mathcal{B}$. Roughly speaking, we relate each state of the ILTS with the interpretation, so that the assigned valuations describe the reaction mixture, either by identifying each of the agents, or at a certain level of abstraction. Moreover, the transitions are labelled by the rule which enables the transition. The origin of the transition is the state whose interpretation satisfies the left-hand-side of the rule, and the activity is

proportional to the rate of that rule.

Definition 3.6 *(Full ILTS which models the rule-based system)* Given a rule-based system $\mathcal{B} = (\mathcal{V}, \mathcal{E}, n, \mathcal{R}, p_0)$ defined over the set of agent types $\mathcal{A}$ and set of sites $\mathcal{S}$. We construct the ILTS $M_{\mathcal{L}}$ that has as many states as many valuations there are in the set $Val_{(\mathcal{V}',\mathcal{E}')}$, and each state is interpreted with a set with exactly one valuation. Such an ILTS is well-defined, since the intersection between any two satisfaction sets is trivially empty. The initial states are the states whose valuation satisfies $[\![p_0]\!]$ [4] . The set of labels is the set of identified rules. The transition is labeled with R between the states s such that $\mathcal{L}(s) = \{\mathbf{x}\}$ and s', such that $\mathcal{L}(s') = \{\mathbf{x}'\}$ if and only if $\mathbf{x}$ and $\mathbf{x}'$ are such that $\mathbf{x} \in [\![\alpha]\!]$ and $\mathbf{x}' \in [\![\alpha_d]\!]$, and they evaluate all the variables that are not mentioned in the rule R to the same value; Moreover, the activity is given by $a(s,' R', s') = k(R)$. If this holds for all rules $R \equiv (\alpha, \alpha_d; k) \in \mathcal{R}^{id}$, then we say that the transition system $M = (S, L, \delta, S_0)$ *models* the set of rules $\mathcal{R}^{id}$ in interpretation $\mathcal{L}$, written $M_{\mathcal{L}} \models \mathcal{R}^{id}$.

Such an ILTS has dynamics which coincides to the standard way of defining stochastic chemical kinetics over a continuous-time Markov chain [11],[1], [9].

Example 3.7 (Fig.2 revisited) There are 36 different well-formed valuations of the variables for this example: there are 9 ways to set the bonds: one where there are no bonds, four different valuations which encode for a mixture with one bond, and four different valuations which encode for mixtures with two bonds. Moreover, any of these configurations may be encoded with in four different ways, depending on the values of internal states of A_1 and C_1, ie the valuations of variables $a_{(A_1,s)}$ and $a_{(C_1,s)}$. This makes in total $(1 + 4 + 4) \cdot 4 = 36$.

Definition 3.8 *(Agent-view)* Given a rule-based system $\mathcal{B} = (\mathcal{V}, \mathcal{E}, n, \mathcal{R}, p_0)$ defined over the set of agent types $\mathcal{A}$ and set of sites $\mathcal{S}$, let $\mathcal{R}_A$ be the subset of rules $\mathcal{R}$, such that for all $R \in \mathcal{R}_A$, it holds that $Var_R \cap Var_A \neq \emptyset$. The full ILTS over the subset of rules $\mathcal{R}_A$ we call the *agent-view* of agent A.

We acknowledge that, due to the fact that the rule-set is closed under permuting the identifiers of the agents, we may define a *population-based* ILTS which models the rule-based system.

Definition 3.9 *(Population-based ILTS which models the rule-based system)* Given a rule-based system $\mathcal{B} = (\mathcal{V}, \mathcal{E}, n, \mathcal{R}, p_0)$ defined over the set of agent types $\mathcal{A}$ and set of sites $\mathcal{S}$. We construct the ILTS $M_{\mathcal{L}}$ that has as many states as many valuations there are in the set $Val_{(\mathcal{V}',\mathcal{E}')}$ partitioned by the equivalence relation $\sim \subseteq Val_{(\mathcal{V}',\mathcal{E}')} \times Val_{(\mathcal{V}',\mathcal{E}')}$, which identifies all the states up to the permutation over the identifiers

[4] note that p_0 is defined over the set of variables $Var_{\mathcal{V},\mathcal{E}}$, whereas the valuations assigned to states are over the variables $Var_{\mathcal{V}',\mathcal{E}'}$. We think of it along the lines of how the propositions which appear in rules are instantiated when agent multiplicities are given;

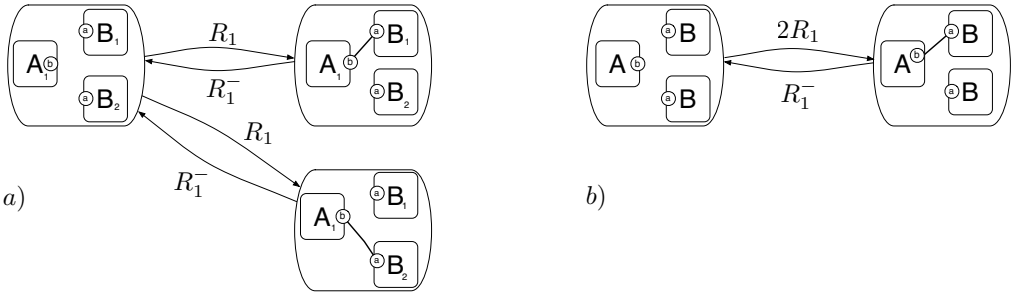


Fig. 5. (a) Representation of a view of agent A for the set of rules $\mathcal{R} = \{R_1, R_2\}$; (b) Population-based ILTS which models rule R_1 in Ex.2.

of agents of the same type:

$$\mathbf{x} \sim \mathbf{x}' \text{ if there exists a permutation } \sigma : \{1,\dots,n(A)\}_{A\in\mathcal{A}} \to \{1,\dots,n(A)\}_{A\in\mathcal{A}} \text{ such that for all } i,\ \mathbf{x}(a_{(A_i,s)}) = \mathbf{x}'(a_{(\sigma(A_i),s)}).$$

We set $S \equiv Val/_\sim$. Each state is assigned a set of valuations which belong to this equivalence class. Let us denote by $[\mathbf{x}, \alpha]$ the number of different instantiations of identifiers of variables in Var_α, such that $\mathbf{x} \models \alpha^{id}$ [5]. The set of labels is the set of rules without identifiers, ie $L = \mathcal{R}$. Two states $\mathbf{x}_{1\sim}, \mathbf{x}_{2\sim} \in Val/_\sim$ are connected by a label R that corresponds to the rule $R \equiv (\alpha, \alpha_d; k)$, if and only if the representative of the class $\mathbf{x}_{1\sim}$ satisfies the left-hand-side condition of the rule R, and the rate assigned to the label $R \in \mathcal{R}$ is equal to $k(R) \cdot [\mathbf{x}_1, \alpha]$.

Example 3.10 (Fig.2 revisited) Let us observe the valuations $\mathbf{x} = \begin{pmatrix} a_{(A_1,x)} & b_{((A_1,b),(B_1,a))} & b_{((B_2,c),(C_1,b))} & a_{(C_1,y)} \\ 0 & 1 & 1 & 1 \end{pmatrix}$, and $\mathbf{x}' = \begin{pmatrix} a_{(A_1,x)} & b_{((A_1,b),(B_2,a))} & b_{((B_1,c),(C_1,b))} & a_{(C_1,y)} \\ 0 & 1 & 1 & 1 \end{pmatrix}$ [6]. It holds that $\mathbf{x} \sim \mathbf{x}'$, because we have a permutation $\sigma \begin{pmatrix} A_1 & B_1 & B_2 & C_1 \\ A_1 & B_2 & B_1 & C_1 \end{pmatrix}$, such that $\mathbf{x}(a_{(A_i,s)}) = \mathbf{x}(a_{(\sigma(A_i),s)})$ for all $A \in \mathcal{A}$, and $i = 1,\dots,n(A)$. The equivalence class whose representative is $\mathbf{x}$ and $\mathbf{x}'$ can be described as 'one dimer consisting of agents A and B and one dimer consisting of agents B and C'. If this state is named s, then we assign it the interpretation sets $\mathcal{L}(s) = \{\mathbf{x}, \mathbf{x}'\}$. There are 20 states in the population-based ILTS which models the system in the example – there are 5 ways to set the bonds: one where there are no bonds, two different valuations which encode for a mixture with one bond (either a complex is formed between agents of type A

[5] Think of having a rule were $\alpha \equiv \neg a_{(A,x)}$ and agents A_1 and A_2; then the cardinality $[\mathbf{x}, \alpha]$ may be 0, 1 or 2, depending on how many A's are free.

[6] we do not mention each agent's bond variables, since it is clear from context

and B, or between agents of type B and C), and two different valuations which encode for mixtures with two bonds (either one trimer with A, B and C, and one B is free, or two dimers are formed and no B is free). This observation leads to the 'population-based' semantics of the agent ensemble, which is the standard description.

4 Model decomposition

The state space of a full ILTS which models the rule-based system grows proportionally to the number of variables over its full contact map, which grows combinatorialy in the number of agents and the complexity of their interfaces. We propose to define it as a composition of smaller ILTS. We start by an ILTS which models each rule separately, and then we define a *composition* operator over them. In other words, we decompose the ILTS as a standard product of the set of smaler ILTS.

Definition 4.1 *(Cross-product of two ILTS)* Given two ILTS: $M_{1,\mathcal{L}_1} = (S_1, L_1, \delta_1, S_{0_1})$, with a set of variables Var_1, valuations Val_1 and an interpretation over states $\mathcal{L}_1$, and $M_{2,\mathcal{L}_2} = (S_2, L_2, \delta_2, S_{0_2})$, with Var_2 and Val_2 and $\mathcal{L}_2$, such that $L_1 \cap L_2 = \emptyset$, and $Var_1 \cap Var_2 = \emptyset$. We define the product $M_{\mathcal{L}} = (S, L, \delta, S_0)$, written $M_{\mathcal{L}} = M_{1,\mathcal{L}_1} \times M_{2,\mathcal{L}_2}$ in the following way:

- $S = S_1 \times S_2$,
- $L = L_1 \cup L_2$,
- $\delta((s_1, s_2), l) = (\delta_1(s_1, l), \delta_2(s_2, l))$, for any $l \in L$,
- $(s_1, s_2) \in S_0$ iff $s_1 \in S_{0_1}$ and $s_2 \in S_{0_2}$ (i.e. $S_0 = S_{0_1} \times S_{0_2}$).

Moreover, we set $Var = Var_1 \cup Var_2$, and we interpret the pair of states by the intersection of valuation sets of each of them:

$$\mathcal{L}((s_1, s_2)) = \mathcal{L}_1(s_1) \cap \mathcal{L}_2(s_2).$$

We can also see the ILTS $M_{1,\mathcal{L}_1}$ (resp. $M_{1,\mathcal{L}_1}$) as a *projection of the ILTS* $M_{\mathcal{L}}$ to the set of variables Var_1 (resp. Var_2), and we may write $M_{1,\mathcal{L}_1} = M_{\mathcal{L}}|_{Var_1}$.

The only constraint for two ILTS to be composed by a cross-product is that they are defined over the mutually disjoint sets of variables and mutally disjoint sets of labels.

Proposition 4.2 *(Decomposing ILTS)* Given a rule-based system $\mathcal{B} = (\mathcal{V}, \mathcal{E}, n, \mathcal{R}, p_0)$ defined over the set of agent types $\mathcal{A}$ and set of sites $\mathcal{S}$. Let $M_{\mathcal{L}}$ be the full ILTS which models $\mathcal{R}^{id}$. If we can partition the set of rules into classes $\mathcal{R}_1,\ldots,\mathcal{R}_m$, such that $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_m$, and each two classes have mutually disjoint sets of variables, then $M_{\mathcal{L}}$ can be decomposed in the following form:

$$M_{\mathcal{L}} = \prod_{i=1}^{m} M_{i\mathcal{L}_i},$$

where for all $i = 1, \ldots m$, the ILTS $M_{i,\mathcal{L}_i}$ models $\mathcal{R}_i^{id}$.

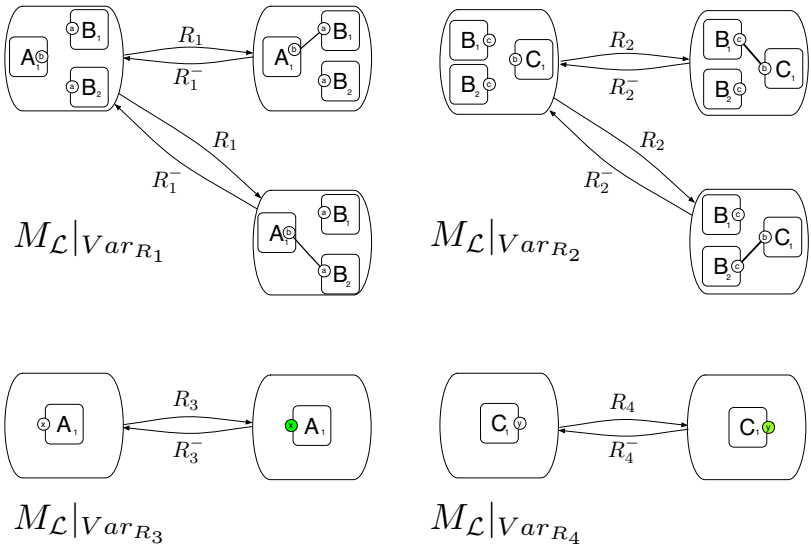


Fig. 6. Decomposition: $M_{\mathcal{L}} = M_{\mathcal{L}}|_{Var_{R_1}} \times M_{\mathcal{L}}|_{Var_{R_2}} \times M_{\mathcal{L}}|_{Var_{R_3}} \times M_{\mathcal{L}}|_{Var_{R_4}}$.

Example 4.3 (Fig.2 revisited) The sets of variables which appear in each of the rules are $Var_{R_1^{id_A=1,id_B=1}} = \{a_{(A_1,b)}, a_{(B_1,a)}, b_{((A_1,b),(B_1,a))}\}$, $Var_{R_1^{id_A=1,id_B=2}} = \{a_{(A_1,b)}, a_{(B_2,a)}, b_{((A_1,b),(B_2,a))}\}$, $Var_{R_2^{id_B=1,id_C=1}} = \{a_{(B_1,c)}, a_{(C_1,b)}, b_{((B_1,c),(C_1,b))}\}$, $Var_{R_2^{id_B=1,id_C=1}} = \{a_{(B_1,c)}, a_{(C_1,b)}, b_{((B_1,c),(C_1,b))}\}$, $Var_{R_3^{id_A=1}} = \{a_{(A_1,x)}\}$, $Var_{R_4^{id_C=1}} = \{a_{C_1,y}\}$. Not all of them are mutually disjoint, but we can group the sets of variables into the following disjoint classes: $Var_{R_1} = Var_{R_1^{id_A=1,id_B=1}} \cup Var_{R_1^{id_A=1,id_B=2}}$, $Var_{R_2} = Var_{R_2^{id_B=1,id_C=1}} \cup Var_{R_1^{id_B=2,id_C=1}}$, $Var_{R_3} = Var_{R_3^{id_A=1}}$, $Var_{R_4} = Var_{R_4^{id_C=1}}$.

We build the four ILTS which models each of these classes of variables: $M_{\mathcal{L}_1} \models \{R_1^{id_A=1,id_B=1}, R_1^{id_A=1,id_B=2}\}$, and similarly M_{L_2}, M_{L_3}, M_{L_4}. The ILTS $M_{\mathcal{L}}$ which models the rules $\mathcal{R}^{id}$ is the following composition:

$$M_{\mathcal{L}} = M_{\mathcal{L}_1} \times M_{\mathcal{L}_2} \times M_{\mathcal{L}_3} \times M_{\mathcal{L}_4}.$$

$M_{\mathcal{L}}$ is well-defined ILTS, and its projections are $M_{\mathcal{L}_1} = M_{\mathcal{L}}|_{Var_{R_1}}$, $M_{\mathcal{L}_2} = M_{\mathcal{L}}|_{Var_{R_2}}$, $M_{\mathcal{L}_3} = M_{\mathcal{L}}|_{Var_{R_3}}$, and $M_{\mathcal{L}_4} = M_{\mathcal{L}}|_{Var_{R_4}}$.

5 Constructing the generator

If we equip the ILTS that models a rule-based system with a stochastic semantics according to a continuous-time Markov chain, each ILTS that models a single rule can be thought of as a stochastic automaton and the composition thereof as a stochastic automata network. We introduce the construction of the generator by revisiting the example discussed in Fig. 2. Analyzing the rule-set and considering Proposition 4.2 we conclude that the variable sets are disjoint and we showed the network of four ILTS projections in Fig. 6 for the case $n(A) = n(C) = 1$, and $n(B) =$

2. We compose the generator matrix $\mathbf{Q}$ of the network of stochastic automata out of elementary matrices that are derived from the individual automata. Consider a network composed of ILTS $\{M_1, \ldots, M_m\}$. Each ILTS M_i is characterized by a set of transitions labeled from the set L^i. For instance, for the network in Fig. 6 we have $L^1 = \{R_1^{id_A=1,id_B=1}, R_1^{id_A=1,id_B=2}, R_1^{-id_A=1,id_B=1}, R_1^{-id_A=1,id_B=2}\}$. For each automaton M_i and label $l \in L^i$, we define an elementary rate matrix $\mathbf{E}_l^i$, the element $E_l^i(j,k)$ of which denotes the rate of exiting state j to state k by transition l in automaton M_i. Finally, to ensure zero row-sum of the generator we design a matrix $\mathbf{D}_l^i = \mathrm{diag}(\mathbf{E}_l^i \mathbf{e})$, with unit vector $\mathbf{e}$. According to [16,3] the generator can then be expressed as

$$\mathbf{Q} = \bigoplus_{i=1}^{m} \sum_{l \in L^i} \mathbf{E}_l^i - \bigoplus_{i=1}^{m} \sum_{l \in L^i} \mathbf{D}_l^i$$

where we use the symbol $\oplus$ to denote the Kronecker sum [19]. The composition includes only the Kronecker sum, which is known to correspond to the classical composition of independent continuous-time Markov chains. We restrict to the case of independent ILTS (there are no synchronized transitions between the small automata), so the part which involves the Kronecker product operator does not appear in the expression.

Example 5.1 (Fig.2 revisited) Going back to Fig. 6 and exemplify the construction for the projection $M_{\mathcal{L}}|_{Var_{R_1}}$ we have $L^1 = \{R_1^{id_A=1,id_B=1}, R_1^{id_A=1,id_B=2}, R_1^{-id_A=1,id_B=1}, R_1^{-id_A=1,id_B=2}\}$ with state space $S = \{s_1, s_2, s_3\}$. The elementary matrices then become

$$\mathbf{E}^{M_1}_{R_1^{id_A=1,id_B=1}} = \begin{bmatrix} 0 & k_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{E}^{M_1}_{R_1^{-id_A=1,id_B=1}} = \begin{bmatrix} 0 & 0 & k_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and

$$\mathbf{E}^{M_1}_{R_1^{-id_A=1,id_B=2}} = \begin{bmatrix} 0 & 0 & 0 \\ k_1^- & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{E}^{M_1}_{R_1^{-id_A=1,id_B=2}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ k_1^- & 0 & 0 \end{bmatrix},$$

with $\mathbf{E}_l^i = \mathbf{I}$ for $l \notin L^i$.

Let us discuss another example, where we review the construction proposed in Section 4. Consider a kinase K can that bind a substrate S and phosphorylate its two modification sites $\mathtt{s_2}$ and $\mathtt{s_3}$ independently. In Kappa syntax that is to say

$$\begin{aligned} R_1: &\quad \mathtt{K(k_1), S(s_1)} \underset{c_1^-}{\overset{c_1^+}{\rightleftharpoons}} \mathtt{K(k_1^1), S(s_1^1)} \\ R_2: &\quad \mathtt{K(k_1^1), S(s_1^1, s_2^u)} \underset{c_2^-}{\overset{c_2^+}{\rightleftharpoons}} \mathtt{K(k_1^1), S(s_1^1, s_2^p)} \\ R_3: &\quad \mathtt{K(k_1^1), S(s_1^1, s_3^u)} \underset{c_3^-}{\overset{c_3^+}{\rightleftharpoons}} \mathtt{K(k_1^1), S(s_1^1, s_3^p)}. \end{aligned} \tag{1}$$

Starting out with considering the sets of variables which appear in each of the rules R_1, R_2 and R_3, since they are all having non-empty mutual intersection, we cannot apply the decomposition of the ILTS which models all the rules, proposed in Prop. 4.2. Consequently, the agent-views of each of the agents expose and determine the complete interface of every agent, which equivalent to saying that the views are fully specified species. The effective degrees of freedom of this system, taking aside mass-conservation relations, is thus equal to the number of distinct reachable species.

6 Conclusions

The paper proposes a natural approach to describe the stochastic interactions of highly structured molecular agents. Each agent is associated a stochastic automaton, which describes the degrees of freedom, views of that agent. It allows for a bottom-up construction of the effective state-space. This modular representation comes at an expense: the product formulation overapproximates the reachable state space – however not the dimension of it. Naturally, the approach gets more appeal, the more local transitions per automaton there are, i.e., the richer the internal logic of an agent becomes.

We showed how to represent the generator matrix of the underlying Markov process of the whole rule-set as Kronecker sums of the rate matrices belonging to individual view-automata. The decomposition criterion is derived by analysing the set of variables that appear in each rule. One can understand the intuition behind the decomposition principle as taking advantage of the statistical independence of certain events. In reference to the rule-set of (1), we discuss in the following a future research direction, which promises further decomposition. Namely, to utilize the weaker notion of conditional independence to construct state spaces of reduced dimension. The two modification sites s_2 and s_3 in (1) are not statistically independent, i.e., the joint probability $\Pr(s_2, s_3)$ cannot be factorized. However, conditioning on the state s_1 the events become independent. Using the product rule for conditional probabilities we have the identity $\Pr(s_2, s_3|s_1) = \Pr(s_2|s_3, s_1)\Pr(s_3|s_1)$. Independence means that conditioning the state s_2 on s_3 and s_1 is the same as just conditioning on s_1 alone. Thus, we have $\Pr(s_2, s_3|s_1) = \Pr(s_2|s_1)\Pr(s_3|s_1)$. Let us consider a coarse-grained fragmentation that does not enumerate the site states s_2 and s_3 but just the site state s_1. Then the reconstruction problem is, whether given $\Pr(s_1)$ we can reconstruct the joint $\Pr(s_2, s_3, s_1)$. Accounting for the dependency structure at hand, we have $\Pr(s_2, s_3, s_1) = \Pr(s_2|s_1)\Pr(s_3|s_1)\Pr(s_1)$ and we are left with defining the two conditional distributions. The number of such modification events follow a Poission law. Thus, the modification states s_2 and s_3 can be recovered from knowing s_1. Clearly, in situation where the substrate `S` in (1) has multiple independent phosphorylation sites, the presented decomposition would be exponentially larger than the one which exploits conditional, and not full independence.

References

[1] Baier, C., B. Haverkort, H. Hermanns and J.-P. Katoen, *Model-checking algorithms for continuous-time Markov chains*, IEEE Transactions on Software Engineering **29** (2003), p. 2003.

[2] Borisov, N. M., N. I. Markevich, J. B. Hoek and B. N. Kholodenko, *Signaling through receptors and scaffolds: Independent interactions reduce combinatorial complexity*, Biophys J **89** (2005), pp. 951—966.

[3] Buchholz, P. and P. Kemper, *Kronecker based matrix representations for large Markov models*, in: *Validation of Stochastic Systems*, Lecture Notes in Computer Science **2925**, 2004, pp. 256—295.

[4] Changeux, J.-P. and S. J. Edelstein, *Allosteric mechanisms of signal transduction*, Science **308** (2005), pp. 1424–1428.

[5] Conzelmann, H., J. Saez-Rodriguez, T. Sauter, B. N. Kholodenko and E. D. Gilles, *A domain oriented approach to the reduction of combinatorial complexity in signal transduction networks*, BMC Syst Biol **7** (2006).

[6] Danos, V., J. Feret, W. Fontana and J. Krivine, *Abstract interpretation of cellular signalling networks*, , **4905**, 2008, pp. 42–58.

[7] Danos, V. and C. Laneve, *Core formal molecular biology*, Theoretical Computer Science **325** (2003), pp. 69–110.

[8] Feret, J., V. Danos, J. Krivine, R. Harmer and W. Fontana, *Internal coarse-graining of molecular systems*, Proc. Natl. Acad. Sci. USA **106** (2009), pp. 6453–6458.

[9] Feret, J., T. A. Henzinger, H. Koeppl and T. Petrov, *Lumpability abstractions of rule-based systems*, MeCBIC (2010), pp. 142–161.

[10] Feret, J., H. Koeppl and T. Petrov, *Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models*, International Journal of Software and Informatics **in press** (2010).

[11] Gillespie, D. T., *Exact stochastic simulation of coupled chemical reactions*, J Phys Chem **81** (1977), pp. 2340–2361.

[12] Kemper, P., *Numerical analysis of superposed GSPNs*, IEEE T Software Eng **9** (1996), pp. 615–628.

[13] Mori, T., D. R. Willams, M. O. Bryne, X. Qin, M. Egli, H. S. Mchaourab, P. L. Stewart and C. H. Johnson, *Elucidating the ticking of an in vitro circadian clockwork*, PLoS Biol **5** (2007), pp. 841–853.

[14] Nakajima, M., K. Imai, H. Ito, T. Nishiwaki, Y. Murayama, H. Iwasaki, T. Oyama and T. Kondo, *Reconstitution of circadian oscillation of cyanobacterial KaiC phosphorylation in vitro*, Science **308** (2005), pp. 414–415.

[15] Nishiwaki, T., Y. Satomi, Y. Kitayama, K. Terauchi, R. Kiyohara, T. Takao and T. Kondo, *A sequential program of dual phosphorylation of KaiC as a basis for circadian rhythm in cyanobacteria*, EMBO J **26** (2007), pp. 4029–4037.

[16] Plateau, B., *On the stochastic structure of parallelism and synchronization models for distributed algorithms*, , **13**, 1985, pp. 147–154.

[17] Salazar, C. and T. Höfer, *Multisite protein phosphorylation - from molecular mechanisms to kinetic models*, FEBS J **276** (2009), pp. 3177–3198.

[18] Smock, R. G. and L. M. Gierasch, *Sending signals dynamically*, Science **324** (2009), pp. 198–203.

[19] Van Loan, C. F., *The ubiquitous kronecker product*, Comput Appl Math **123** (2000), pp. 85–100.

[20] van Zon, J. S., D. K. Lubensky, P. R. H. Altena and P. Rein ten Wolde, *An allosteric model of circadian KaiC phosphorylation*, Proc Natl Acad Sci USA **104** (2007), pp. 7420–7425.

[21] Walsh, C. T., "Posttranslation Modification of Proteins: Expanding Nature's Inventory," Roberts and Co. Publisher, 2006.

[22] Wolf, V., *Modelling of biochemical reactions by stochastic automata networks*, Electron. Notes Theor. Comput. Sci. **171** (2007), pp. 197–208.