Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

RECONSTRUCTING SPECIES-BASED DYNAMICS FROM REDUCED STOCHASTIC RULE-BASED MODELS

Tatjana Petrov

ETH Zurich
SWITZERLAND

Jerome Feret

DI-ENS (INRIA, ENS, CNRS)
Paris, FRANCE

Heinz Koeppl

ETH Zurich
SWITZERLAND

ABSTRACT

Many bio-molecular reactions inside the cell are characterized by complex-formation and mutual modification of a few constituent molecules that give rise to a combinatorial number of reachable complexes or species. For such cases rule-based models (or site-graph-rewrite rules), offer a compact model description, by enumerating only the necessary context of interacting molecules. Such a model specification induces symmetries in the underlying Markov chain, which we have recently exploited for model reduction, based on a backward Markovian bisimulation. Interestingly, the method showed a theoretical possibility of reconstructing the high-dimensional species-based dynamics from the aggregate state. In this paper, we present a procedure for reconstructing the high-dimensional species-based dynamics from the aggregate state, and we provide an algorithm for computing such de-aggregation functions explicitly. The algorithm involves counting the automorphisms of a connected site-graph, and has a quadratic time complexity in the number of molecules which constitute the site-graphs of interest. We provide illustrating case studies.

1 INTRODUCTION

A biochemical reaction system involves several molecular species and multiple chemical reactions. Recent advances in real-time single cell imaging, micro-fluidic techniques and synthetic biology have testified to the random nature of gene expression and protein abundance in single cells (Yu et al. 2006; Friedman et al. 2010). Thus, a stochastic description of chemical reactions is often mandatory to analyze the behavior of the system. The dynamics of the system is typically modeled by a continuous-time Markov chain (CTMC) with the state being the number of molecules of each species. A good survey of the Markov process techniques used in analyzing such reaction systems is provided by Anderson and Kurtz (2011).

During induction of signal transduction pathways, transient complex formation of proteins and their post-translational modification give rise to a combinatorial number of distinct molecular species (Hlavacek et al. 2006). By requiring the enumeration of all reachable molecular species, standard chemical kinetics is impractical to describe such combinatorial assembly processes and alternative modeling paradigms need to be sought. One such alternative are rule-based languages such as Kappa (Danos and Laneve 2004) or BioNetGen (Blinov et al. 2004). In these languages, proteins take center stage instead of the molecular species; their modification as well as their binding configuration is explicitly traced. The state of the system is described as a site-graph: nodes represent proteins and each node is assigned a set of sites, which represent protein domains. Some sites can bear a binding state: they can be free, or connected to exactly one other site in the graph, modeling in that way the linking structure between the proteins; other

978-1-4673-4782-2/12/$31.00 ©2012 IEEE

sites can bear an internal state, a string used to encode a level of energy (for example, phosphorilation or ubiquitination).

Recently, attempts have been made to reduce the complexity of rule-based models (Borisov et al. 2005; Feret et al. 2009; Harmer et al. 2010; Feret et al. 2012), by exploiting the limited context on which most binding and modification events are conditioned. The idea of these reductions is to detect a set of *fragments*, such that the observed dynamics (property) of the system can be described self-consistently as a function of those fragments' abundances. Formally, a fragment is a partial molecular species, the one in which parts of proteins' domains may be omitted. The meaning of a fragment can be interpreted from various perspectives. From an intentional point of view, a fragment is a particular subset of properties of a chemical species: if one is able to show that the correlation between the states of sites in different fragments of a species has no impact on the dynamics of the model, then the species can accordingly be cut into fragments. From an extensional perspective, a fragment is understood as a multi-set of chemical species: a fragment is identified with the multi-set of those species in which the fragment occurs. Finally, along the signaling pathway, a fragment represents an intrinsic information carrier, as explained in Harmer et al. (2010).

Fragments can be used in model reduction. In Feret et al. (2009), Harmer et al. (2010), the reduction is performed yielding a self-consistent set of ordinary differential equations of significantly reduced dimension. In Feret et al. (2012), the stochastic dynamics given by the CTMC underlying chemical kinetics is studied with respect to such a reduction. Moreover, in Petrov et al. (2012), the exponential reduction of the size of the underlying state space is demonstrated. Interestingly, in such a reduction, the probability distribution over the species-based states can be recovered from the probability distribution over the fragment-based states. The proven possibility of such a reconstruction, since being characterized only theoretically, opens the question: (how) can one effectively compute such de-aggregating functions in general? More concretely, let $X_t \in \mathcal{X} \subseteq \mathbb{N}^n_{\geq 0}$ be the CTMC of the original rule-set, taking values in the set of species' multisets, let $Y_t \in \mathcal{Y} \subseteq \mathbb{N}^m_{\geq 0}$ with $m < n$ be the CTMC of the transformed rule-set, taking values in the set of fragments' multisets, and assume that $\mathbf{x} \in \mathcal{X}$ is the observable state of interest. From the reduced model, one is able to estimate the probability $\mathsf{P}(Y_t = \mathbf{y})$. However, it is also possible to reconstruct the probability $\mathsf{P}(X_t = \mathbf{x})$, since the conditional probability $\mathsf{P}(X_t = \mathbf{x} \mid Y_t = \mathbf{y})$ is invariant of time; it depends only on the values $\mathbf{x}$, $\mathbf{y}$, and the structure of the site-graphs which represent a species or a fragment. We here provide a general expression for computing those conditional probabilities explicitly. In particular, the computation involves counting the number of automorphisms (symmetries) of a connected site-graph. Suitably enough, the graph isomorphism problem for connected site-graphs is linear in the size of the graph, because site-graphs enjoy the rigidity property: a potential isomorphism is completely determined by a map between two nodes. Hence, a straight-forward procedure is quadratic in the size of the site-graph. In addition, we show a general heuristic approach which provides a radical improvement, and is based on decomposing a site-graph into strongly connected components.

The remaining part of the paper is organized as follows. In Section 2 the basics of stochastic chemical kinetics and the mathematical framework of site-graphs and their modifications in terms of site-graph rewrite rules are introduced. The model reduction and translation of a rule-set to its reduction are defined in Section 3. The expression for inverting the probability of species-based states from the reduced model is given in Section 4. A detailed discussion on counting site-graph automorphisms, accompanied by an illustrative case study is given in Section 5.

2 PRELIMINARIES

2.1 Chemical Reaction Networks

For a well-mixed reaction system with molecular species $S_1,\dots,S_s$ the state of a system can be represented as a multiset of species. We denote their multiplicities in the multiset as $\mathbf{x} = (x_1,\dots,x_s) \in \mathcal{X} \subseteq \mathbb{N}^s_{\geq 0}$. The dynamics of such a system is determined by a set of r reactions. The k-th reaction reads $a_{1k}S_1,\dots,a_{sk}S_s \overset{c_k}{\to} a'_{1k}S_1,\dots,a'_{sk}S_s$, where $a_{ik} \in \mathbb{N}_{\geq 0}$ and $a'_{ik} \in \mathbb{N}_{\geq 0}$ denote the substrate and product stoichiometric coefficients of species i, respectively. If the k-th reaction occurs, after being in the state $\mathbf{x}$, the next state will be $\mathbf{x} + (\mathbf{a}'_k - \mathbf{a}_k) = \mathbf{x} + \mathbf{b}_k$, where $\mathbf{b}_k$ is referred to as the stoichiometric change vector. Under the above physical assumption the species multiplicities follow a continuous-time Markov chain and we denote the state of the system as the t-indexed random vector $X(t) = (X_1(t), \ldots, X_s(t))$. Hence, the probability of moving to the state $\mathbf{x} + \mathbf{b}_k$ from $\mathbf{x}$ after time Δ is

$$\mathsf{P}(X(t+\Delta) = \mathbf{x} + \mathbf{b}_k \mid X(t) = \mathbf{x}) = \lambda_k(\mathbf{x})\Delta + o(\Delta), \tag{1}$$

with λ_k the propensity of reaction k, the functional form of which is often assumed to follow the principle of mass-action (Gillespie 2007) $\lambda_k(\mathbf{x}) = c_k \prod_{i=1}^{s} \binom{x_i}{a_{ik}}$.

2.2 Site-graphs

A molecular species can be a protein, its post-translationally modified form or a protein complex that consists of proteins bound together. In order to reflect this internal structure of molecular species we represent them by *site-graphs*, in which modifications of protein residues and bonds are explicitly encoded. Such representation of species was first introduced in a rule-based modeling language Kappa (Danos and Laneve 2004). However, the formalism which we present is not as expressive as Kappa, it is sometimes referred to as a kernel of Kappa. For example, the release of a dangling bond (in Kappa syntax, written as an expression $A(x!\_) \to A(x)$) cannot be expressed in our framework.

A site-graph is an undirected graph where typed nodes have sites, and edges are partial matchings on sites. For the sake of simplicity, we assume that sites are partitioned into two kinds: the ones bearing a binding state and the ones bearing an internal state (thus, any site with an internal state is free). Internal states range in a predefined set. The nodes of the site-graph can be interpreted as proteins, and sites of a node stand for protein binding domains. Internal states are used to encode post-translational modifications.

Let $\mathcal{S}$ denote the set of site labels, and $\mathcal{I}$ the set of internal values that can be assigned to sites. The function $I : \mathcal{S} \to \wp(\mathcal{I})$ denotes the set of internal values that a site s can take. A site s can be evaluated only to a predefined set of values, $I(s)$. If the site is used for creating a bond, its set of predefined internal values is empty. Moreover, let $\mathcal{A}$ be the set of node types, and each node type is being equipped with a set of sites, defined by a signature map $\Sigma : \mathcal{A} \to \wp(\mathcal{S})$. Finally, let $\mathcal{E} \subseteq \{((A,s),(A',s')) \mid s \in \Sigma(A), s' \in \Sigma(A'), I(s) = I(s') = \varnothing\}$ be a set of predefined edge types.

The standard rule-based models will be defined on the set of node types $\mathcal{A}$ with a signature map Σ, and the model reductions will be defined over a different set of node types and corresponding signatures, denoted by $\tilde{\mathcal{A}}$ and $\tilde{\Sigma}$.

Definition 1 (Site-graph) An $\mathcal{A}$-site-graph is a tuple $G = (V, \text{Type}, S, E, \psi)$ with

- a set of nodes V,
- a node type function $\text{Type} : V \to \mathcal{A}$,
- a node interface function $S : V \to \wp(\mathcal{S})$, such that for $v \in V$, $S(v) \subseteq \Sigma(\text{Type}(v))$,
- a set of edges $E \subseteq \mathcal{E}$, which is
  - symmetric: $((v,s),(v',s')) \in E$ iff $((v',s'),(v,s)) \in E$,
  - injective: if $((v,s),(v',s')) \in E$, $((v,s),(v'',s'')) \in E$, then $(v',s') = (v'',s'')$,
  - irreflexive: for all $v \in V$, $s \in \mathcal{S}$, $((v,s),(v,s)) \notin E$,
- a site evaluation function $\psi : \{(v,s) \mid s \in S(v), I(s) \neq \varnothing\} \to \mathcal{I}$, which assigns an internal value to a node-site combination, so that $\psi(v,s) \in I(s)$.

The set $P = \{(v,s) \mid s \in S(v)\} \subseteq V \times \mathcal{S}$ is called the set of *ports*.

Given an edge $e = (p,p') \in E$, we denote by $\overline{e}$ the symmetric edge (p',p). If we do not specify explicitly the set of node types for a site-graph, we assume that it is an $\mathcal{A}$-site-graph. The restriction of valuation function to a node v is denoted by $\psi_v : S(v) \to \mathcal{I}$. The function Σ in the above definition tracks the sites assigned to a particular node of a site-graph. The site-graphs which model physically existing complexes are those whose interfaces are complete, in the sense that all the sites of node's signature are listed in its interface. We call such site-graphs reaction mixtures. Omitting parts of interfaces of a node will play a role in defining rules and fragments.

Definition 2 (Reaction mixture) An $\mathcal{A}$-site-graph $G = (V, \text{Type}, S, E, \psi)$, such that for all $v \in V$, $S(v) = \Sigma(\text{Type}(v))$ is called an $\mathcal{A}$-reaction mixture.

Definition 3 (Path) Given a site-graph $G = (V, \text{Type}, S, E, \psi)$, a sequence of edges $(e_1, \dots e_k) \in E^k$, $e_i = ((v_i, s_i), (v'_i, s'_i))$, such that $v'_i = v_{i+1}$ and $s'_i \neq s_{i+1}$, for $i = 1, \dots k-1$, is called a *path* between nodes v_1 and v_k.

Definition 4 (Connected site-graph) A site-graph G is connected if there exists a path between every two vertices v and v'.

Definition 5 (Species) A connected $\mathcal{A}$-reaction mixture is called an $\mathcal{A}$-species.

Note that the set of all $\mathcal{A}$-species may be infinite. For example, consider a set of sites $\mathcal{S} = \{x, y\}$, such that $I(x) = I(y) = \varnothing$. Hence, both x and y serve as binding sites. Moreover, let $\mathcal{A} = \{A, B\}$, and $\Sigma(A) = \Sigma(B) = \{x, y\}$, and a set of edge types is $\mathcal{E} = \{((A, y), (B, x)), ((A, x), (B, y))\}$. In such a model, there can be infinitely many connected reaction mixtures, each of them being a polymer with an arbitrary number of nodes of alternating types A and B.

Two site-graphs can be related by an embedding function, which is important for assigning the stochastic process to a rule-based model. The symmetry of a site-graph is formalized as a bijective embedding of a site-graph to itself, called automorphism.

Definition 6 (Embedding, isomorphism, automorphism) The *embedding* σ between site-graphs $G = (V, \text{Type}, S, E, \psi)$ and $G' = (V', \text{Type}', S', E', \psi')$, is induced by a support function $\sigma^* : V \to V'$, if

- σ^* is injective: for all $v, v' \in V$, $[\sigma^*(v) = \sigma^*(v') \implies v = v']$;
- for all $v \in V$, $\text{Type}(v) = \text{Type}'(\sigma^*(v))$ and for all $s \in \mathcal{S}$ such that $I(s) \neq \varnothing$, $\psi'(\sigma^*(v), s) = \psi(v, s)$;
- for all $v \in V$, $[s \in S(v) \implies s \in S(\sigma^*(v))]$;
- for all $((v, s), (v', s')) \in (V \times \mathcal{S})^2$, if $s \in S(v)$ and $s' \in S(v')$, then $[((v, s), (v', s')) \in E$ iff $((\sigma^*(v), s), (\sigma^*(v'), s')) \in E']$.

If σ^* is bijective, then σ is an *isomorphism*. An isomorphism between G and itself is called *automorphism*. The set of embeddings between the site-graph G and G' is denoted by $Emb(G, G')$. The set of automorphisms of a site-graph G is denoted by $Aut(G)$. The set cardinality will be denoted by $|\cdot|$.

2.3 Rule-based Models

Definition 7 (Rule) Let G, G' be site-graphs, and $c \in \mathbb{R}_{\geq 0}$ a non-negative real number. The triple (G, G', c), also denoted by $G \xrightarrow{c} G'$, is called a rule. A rule is *well-defined*, if $G' = (V', \text{Type}', S', E', \psi')$ can be derived from $G = (V, \text{Type}, S, E, \psi)$ by a finite number of applications of five elementary site-graph transformations:

1. adding an edge: $\delta_{ae}(G, e) = (V, \text{Type}, S, E \cup \{e, \bar{e}\}, \psi)$,
2. deleting an edge: $\delta_{de}(G, e) = (V, \text{Type}, S, E \setminus \{e, \bar{e}\}, \psi)$,
3. changing the state value: $\delta_{ci}(G, v', s', i') = (V, \text{Type}, S, E, \psi')$, $\psi'(v, s) = \begin{cases} i', & \text{if } v = v' \text{ and } s = s', \\ \psi(v, s), & \text{otherwise;} \end{cases}$
4. adding a node: $\delta_{an}(G, v', \psi_{v'}) = (V', \text{Type}, S, E, \psi')$, such that $V' = V \cup \{v'\}$, $S(v') = \Sigma(v')$, and $\psi'(v, s) = \begin{cases} \psi_{v'}(s), & \text{if } v' = v, \\ \psi(v, s), & \text{otherwise;} \end{cases}$
5. deleting a node: $\delta_{dn}(G, v) = (V', \text{Type}, S, E', \psi')$, such that $V' = V \setminus \{v\}$, $\psi' = \psi|_{V'}$ (the restriction of function ψ to the set of nodes V'), and $E' = E \setminus \{e \mid \text{there is a site } s \text{ and a port } p, \{e, \bar{e}\} \cap \{(v, s), p)\} \neq \varnothing\}$.

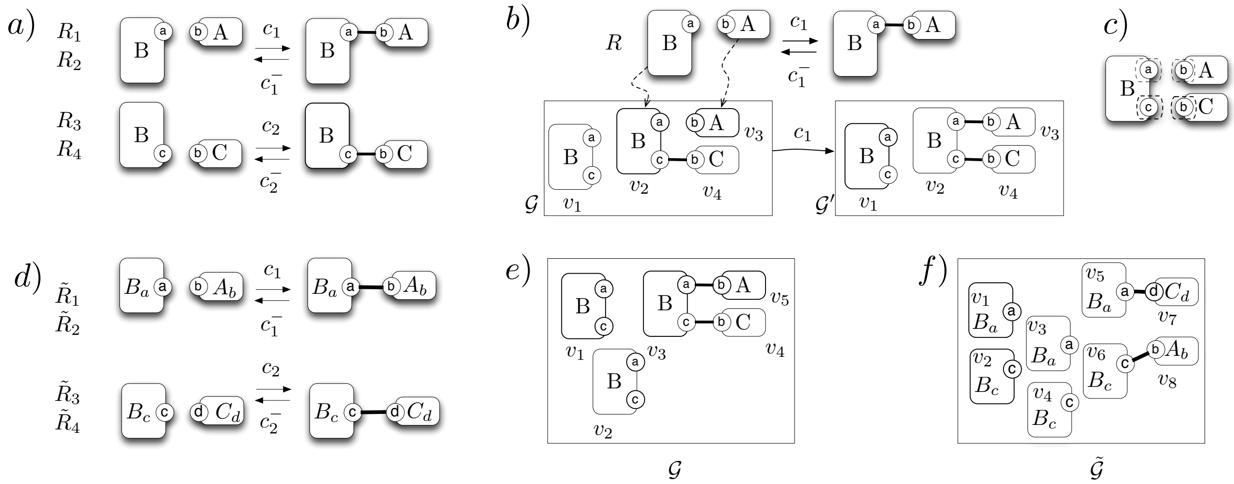

Figure 1: Illustration. a) A rule set over the set of node types $\mathcal{A}$; b) Rule application: Rule R can be applied to a reaction mixture $\mathcal{G}$ via the embedding indicated by the dotted arrows. The result of the rule application is the reaction mixture $\mathcal{G}'$, and the corresponding rate is c. c) Annotation of the node types; d) The translation to a new set of node types -$\tilde{\mathcal{A}}$; e) An $\mathcal{A}$-reaction mixture; f) An $\tilde{\mathcal{A}}$-reaction mixture.

In a rule (G_i, G_i', c_i), the site-graphs G_i and G_i' are such that parts of the nodes' signatures may be omitted. That is the crucial advantage of using site-graphs for modeling molecular interactions, because it enables to implicitly group species which share a set of properties. The interface function S is unaltered under any of the transformations - a site cannot be added or deleted from node's interface by a rule. Adding a node needs an evaluation of state of the sites which can bear an internal state.

Definition 8 (Rule-based model) A rule-based model $\mathcal{R} = \{R_1, \ldots, R_n\}$ is a set of well-defined $\mathcal{A}$-site-graph rewrite rules.

The continuous-time Markov chain (CTMC) assigned to a rule-based model takes values in the set of $\mathcal{A}$-reaction mixtures reachable from the initial one. A rule $R_i = (G_i, G_i', c_i) \in \mathcal{R}$ can be applied to a reaction mixture $\mathcal{G}$ if there exists an embedding between G_i and $\mathcal{G}$. The result of application of a rule to a reaction mixture is unique for a chosen embedding, rigorously defined in Danos et al. (2010). Intuitively, the part of the mixture to which the lhs of the rule is embedded, is transformed by the rule, whereas the rest of it remains unchanged. We here omit the formal definition, due to the space limitations. The illustration is provided in Figure 1.

Given an initial reaction mixture $\mathcal{G}_0$, the set of reachable reaction mixtures is denoted by $\mathbb{G}$: $\mathbb{G} = \{\mathcal{G} \mid \mathcal{G}$ is reachable by a finite number of applications of rules from $\mathcal{R}$ to $\mathcal{G}_0\}$. Let $\mathcal{G}'_{\sigma,i}$ be the reaction mixture obtained after the application of the rule R_i via embedding with support σ. Then, the stochastic behavior of a rule-based model is a CTMC $\{\hat{X}_t\}$, with a state-space $\mathbb{G}$, such that

$$\mathsf{P}(\hat{X}(t+\Delta) = \mathcal{G}'_{\sigma,i} \mid \hat{X}(t) = \mathcal{G}) = c_i\Delta + o(\Delta). \tag{2}$$

3 MODEL REDUCTION

In the following, we show how model reduction is performed. Given a rule-based model $\mathcal{R}$ and its CTMC, $\hat{X}$, both operating over the set of $\mathcal{A}$-reaction mixtures, we derive a different rule-based model — $\tilde{\mathcal{R}}$, such that its CTMC, denoted by $\hat{Y}$, has a smaller state space. In order to do so, a new set of node types, $\tilde{\mathcal{A}}$, is introduced for $\tilde{\mathcal{R}}$. The set $\tilde{\mathcal{A}}$ is induced from an equivalence relation over the signatures of node types from $\mathcal{A}$. For each node type's equivalence class, a new node type arises. We define how to translate any $\mathcal{A}$-site-graph to a $\tilde{\mathcal{A}}$-site-graph. Following that, different $\mathcal{A}$-reaction mixtures can be translated to

$\tilde{\mathcal{A}}$-reaction mixtures. The more equivalence classes per node type there are, the more $\mathcal{A}$-reaction mixtures are translated to a same $\tilde{\mathcal{A}}$-reaction mixture, and the better the reduction is. The projection of the CTMC $\hat{X}$ to the new set of $\tilde{\mathcal{A}}$-reaction mixtures is denoted by $\hat{X}|_{\tilde{\mathcal{A}}}$. Finally, a reduction is called *sound*, if the projection $\hat{X}|_{\tilde{\mathcal{A}}}$ coincides with the process $\hat{Y}$, and it is *complete*, if the transient probabilities of the process $\hat{X}$ can be computed from $\hat{Y}$ only.

Along this section, we use as a running example the rule set consisting of two pairs of reversible rules, given in Figure 1a. In the first pair of rules, two nodes of type B and A can bind/unbind on their sites a and b, while in the second pair of rules, two nodes of type B and C can bind/unbind on their sites c and d. The signature of the model is defined as follows: the set of node types is $\mathcal{A} = \{A,B,C\}$, the set of site types is $\mathcal{S} = \{a,b,c,d\}$, all internal site values are predefined to be empty (since they are all serving as binding sites), the signature map is such that $\Sigma(A) = \{b\}$, $\Sigma(B) = \{a,c\}$, $\Sigma(C) = \{d\}$, and the set of edge types is $\mathcal{E} = \{((A,b),(B,a)),((B,c),(C,d))\}$.

3.1 Projection to a New Set of Node Types

Assume given an equivalence relation $\sim_A$ over the set of sites $\Sigma(A)$, for each agent type $A \in \mathcal{A}$. The new set $\tilde{\mathcal{A}}$ of agent types and a new signature function $\tilde{\Sigma}$ are defined as follows (the set of site labels remains the same):

$$\tilde{\mathcal{A}} = \{A_{\mathcal{C}} \mid A \in \mathcal{A}, \mathcal{C} \in \Sigma(A)_{/\sim_A}\}, \text{ and } \tilde{\Sigma}(A_{\mathcal{C}}) = \mathcal{C}. \tag{3}$$

For each equivalence class $\mathcal{C} \in \Sigma(A)_{/\sim_A}$ assigned to a node type $A \in \mathcal{A}$, a new node type is created. The signature of the new node type is exactly the set of sites which belong to the class $\mathcal{C}$.

In our running example, we take the following equivalence relations: $\sim_A$ contains one equivalence class: a singleton $\{b\}$, $\sim_B$ contains two equivalence classes: $\{a\}$ and $\{c\}$, and $\sim_C$ contains one equivalence class: a singleton $\{d\}$. This way, each instance of nodes of type B is split into two parts in the corresponding reduced model (we will show in the next subsection that these equivalence relations define a sound model reduction) Hence the set of agent types in the reduced model is $\tilde{\mathcal{A}} = \{A_b, B_a, B_c, C_d\}$, as depicted in Figure 1c.

We now define how to project an $\mathcal{A}$-site-graph to a $\tilde{\mathcal{A}}$-site-graph. Let $\mathbf{G}$ be the set of all $\mathcal{A}$-site-graphs, and $\tilde{\mathbf{G}}$ the set of all $\tilde{\mathcal{A}}$-site-graphs. The translation function $\tau : \mathbf{G} \to \tilde{\mathbf{G}}$ maps a site-graph $G = (V, \text{Type}, S, E, \psi) \in \mathbf{G}$ to a site-graph $\tilde{G} = (\tilde{V}, \tilde{\text{Type}}, \tilde{S}, \tilde{E}, \tilde{\psi}) \in \tilde{\mathbf{G}}$, that is, $\tilde{G} = \tau(G)$, if there exists a map $\tau^* : V \to \wp(\tilde{V})$, such that:

- $\tilde{v} \in \tau^*(v)$ iff
  - Type$(v) = A$, and there exists $\mathcal{C} \in \Sigma(A)_{/\sim_A}$ such that $\tilde{\text{Type}}(\tilde{v}) = A_{\mathcal{C}}$ and $S(v) \cap \mathcal{C} \neq \varnothing$, $S(v) = \tilde{S}(\tilde{v})$;
  - $S(v) = \cup_{\tilde{v} \in \tau^*(v)} \tilde{S}(\tilde{v})$;
- $((\tilde{v},s),(\tilde{v}',s')) \in \tilde{E}$ iff $s \in \tilde{\Sigma}(\tilde{v})$, $s' \in \tilde{\Sigma}(\tilde{v}')$ and $((v,s),(v',s')) \in E$, $\tilde{v} \in \tau^*(v)$, $\tilde{v}' \in \tau^*(v')$,
- $\tilde{\psi}(\tilde{v},s) = \psi(v,s)$ iff $I(s) \neq \varnothing$, $\tilde{v} \in \tau^*(v)$, and $s \in \tilde{\Sigma}(\tilde{v})$.

In the running example, applying the function τ consists in splitting each node of type B into two parts. For instance, applying the function τ to the $\mathcal{A}$-reaction mixture $\mathcal{G}$, depicted in Figure 1e, gives the $\tilde{\mathcal{A}}$-reaction mixture $\tilde{\mathcal{G}}$, depicted in Figure 1f. Clearly, the information of how exactly the parts of interfaces were combined in the original mixture is lost. Consequently, different $\mathcal{A}$-reaction mixtures may be mapped to the same $\tilde{\mathcal{A}}$-reaction mixture. We denote the corresponding inverse transformation by $\tau^{-1} : \tilde{\mathbf{G}} \to \wp(\mathbf{G})$, defined by $\tau^{-1}(\tilde{G}) = \{G \mid \tau(G) = \tilde{G}\}$. Moreover, we denote by $\hat{X}|_{\mathcal{A}}$ the corresponding lumped stochastic process, a $\tilde{\mathcal{A}}$-projection of $\hat{X}$: $\hat{X}|_{\tilde{\mathcal{A}}} = \tilde{\mathcal{G}}$ iff $\hat{X} \in \tau^{-1}(\tilde{\mathcal{G}})$.

3.2 Sound and Complete Reduction

For some choices of splitting the nodes' interfaces, the process $\hat{X}|_{\tilde{\mathcal{A}}}$ is not necessarily Markov, nor homogeneous. It is therefore not always possible to define a CTMC directly over the new state space $\tilde{\mathbf{G}}$. Conditions can be imposed on the transition matrix of the Markov chain $\hat{X}$ to ensure that the lumped

Algorithm 1: Procedure for annotating the node type's signatures

```
Input  : A rule-based model $\mathcal{R} \equiv \{R_1,\dots,R_n\}$ over the set of node types $\mathcal{A}$ and the signature $\Sigma$, where for any $i$ between 1 and $n$,
         $R_i \equiv (G_i, G_i', c_i)$, $G_i = (V_i, \mathrm{Type}_i, S_i, E_i, \psi_i)$, $G_i' = (V_i', \mathrm{Type}_i', S_i', E_i', \psi_i')$.
Output:  Equivalence relations of the set of sites of each node type.

for each $A \in \mathcal{A}$ do $\sim_A = \{\{s\} \mid s \in \Sigma(A)\}$;
for each $R_i \in \mathcal{R}$ do
    for each $v, v' \in V_i \cup V_i'$ such that $Type(v) = Type(v')$ do
        $A = \mathrm{Type}(v)$;
        for each $s \in S_i(v) \cup S_i'(v)$ and each $s' \in S_i(v') \cup S_i'(v')$ do  $\sim_A = \textsc{addrelation}(\sim_A, s, s')$;
    end
end
return $(\sim_A)_{A \in \mathcal{A}}$;
\⋆ For any node type $A \in \mathcal{A}$, $\sim_A$ is a equivalence relation that is encoded by a forest as in the Union-Find algorithm (Cormen
et al. 2001), the primitive $\textsc{addrelation}(\sim, a, b)$ fuses the two $\sim$-equivalence classes $[a]_\sim$ and $[b]_\sim$. ⋆\
```

Algorithm 2: Translating a rule-based model.

```
Input  :  A rule $R \equiv (G, G', c)$, over the set of node types $\mathcal{A}$, the set of site labels $\mathcal{S}$, and with the interface $\Sigma$, and the annotation
          classes given by the equivalence relations $\{\sim_A\}_{A \in \mathcal{A}}$, such that $\sim_A \subseteq \Sigma(A)^2$.
Output:  A rule $\tilde{R} = (\tilde{G}, \tilde{G}', \tilde{c})$, over the set of nodes $\tilde{\mathcal{A}}$, the set of site label $\mathcal{S}$, and with the interface $\tilde{\Sigma}$.

$\tilde{G} = \tau(G)$; $\tilde{G}' = \tau(G')$; $\tilde{c} = c$; $(\tilde{V}, \tilde{\mathrm{Type}}, \tilde{S}, \tilde{E}, \tilde{\psi}) = \tilde{G}$; $(\tilde{V}', \tilde{\mathrm{Type}}', \tilde{S}', \tilde{E}', \tilde{\psi}') = \tilde{G}'$;
for all $v \in V \cap V'$ such that $S(v) = \varnothing$ do
    \⋆ we deal with node types which are tested without testing the state of any site ⋆\
    let $A = \mathrm{Type}(v)$ and $\mathcal{C}$ be an arbitrary equivalence class in $\sim_A$;
    $\tilde{V} = \tilde{V} \cup v$; $\tilde{\mathrm{Type}}(v) = A_{\mathcal{C}}$; $\tilde{S} = \tilde{S}[v' \mapsto \varnothing]$; $\tilde{V}' = \tilde{V}' \cup v$; $\tilde{\mathrm{Type}}'(v) = A_{\mathcal{C}}$; $\tilde{S}' = \tilde{S}'[v' \mapsto \varnothing]$;

for all $v \in V \setminus V'$ do
    \⋆ we deal with the agents which are removed in the rule ⋆\
    A=Type(v);
    for all classes $\mathcal{C} \in \mathcal{C}_A$ do
        if $\tilde{Type}(v) \neq A_{\mathcal{C}}$ then
            let $v'$ be a fresh node label not in $V \cup V'$;
            $\tilde{V} = \tilde{V} \cup v'$; $\tilde{\mathrm{Type}}(v') = \mathrm{Type}(v)_{\mathcal{C}}$; $\tilde{S} = \tilde{S}[v' \mapsto \varnothing]$;
            $\tilde{c} = \tilde{c}/[A_{\mathcal{C}}]$;
```

process $\hat{X}|_{\tilde{\mathcal{A}}}$ is also Markov (see for example Buchholz (1994), Rubino and Sericola (1993), Rubino and Sericola (1991), and references therein).

In Feret et al. (2012), it was shown that, if every two sites in the signature of a node type which are tested simultaneously in some of the rules are in the same equivalence class, then the projected process is Markov homogeneous. One can easily inspect on the running example, that the choice of equivalence relations $\sim_A$, $\sim_B$, and $\sim_C$ meets such a criterion. For any rule set, the general construction of the best set of equivalence relations which meet the criterion is outlined in Algorithm 1.

The following Lemma states that, for every class of A's interface, $\mathcal{C} \in \Sigma(A)_{/\sim_A}$, the interface of the node of type A which appears in some rule, is either contained fully in $\mathcal{C}$, or not at all. Moreover, unless the interface of the node is empty, there is a unique class which contains it.

Lemma 1 Let $\{\sim_A\}_{A \in \mathcal{A}}$ be the relations derived by the Algorithm 1. Then, if a node v of type $\mathrm{Type}(v) = A$ appears in the rule $R \in \mathcal{R}$, its interface $S(v)$ is either empty, or there exists exactly one class $\mathcal{C} \in \Sigma(A)_{/\sim_A}$, such that $S(v) \cap \mathcal{C} \neq \varnothing$, and in that case $S(v) \subseteq \mathcal{C}$.

We can check that Lemma 1 holds in our running example. In rules R_1 and R_2, the interface of the node of type A is $\{b\}$, an equivalence class of $\sim_A$; the interface of the node of type B is $\{a\}$, an equivalence class of $\sim_B$. In rules R_3 and R_4, the interface of the node of type B is $\{c\}$, an equivalence class of $\sim_B$; the interface of the node of type C is $\{d\}$; an equivalence class of $\sim_C$.

Every rule-based model $\mathcal{R}$ over the set of node types $\mathcal{A}$ and a CTMC $\hat{X}$ can be translated to a rule-based model $\tilde{\mathcal{R}}$ over a set of node types $\tilde{\mathcal{A}}$ and a CTMC $\hat{Y}$, as shown in Algorithm 2. The following Theorem states that the desired properties of soundness and completeness hold between $\hat{X}$ and $\hat{Y}$. Soundness states that the projection $\hat{X}|_{\tilde{\mathcal{A}}}$ is Markov homogeneous, and that it coincides with the CTMC of the reduced model $\tilde{\mathcal{R}}$, denoted by $\hat{Y}_t$. Completeness guarantees that one can reconstruct the original process by having only the reduced process available.

Theorem 1 ((Feret et al. 2012), (Petrov et al. 2012)) Given a rule-based model $\mathcal{R}$, let $\{\sim_A\}_{A\in\mathcal{A}}$ be the relations derived by the Algorithm 1, $\tilde{\mathcal{R}}$ the corresponding translation of $\mathcal{R}$ (Algorithm 2), and $\hat{Y}$ its CTMC. Then, $\hat{X}|_{\tilde{\mathcal{A}}} \equiv \hat{Y}$ (soundness), and if $\mathsf{P}(\hat{X}_0 = \mathcal{G} \mid \tilde{X}_0 = \tau(\mathcal{G}) = \tilde{\mathcal{G}}) = \frac{1}{|\tau^{-1}(\tilde{\mathcal{G}})|}$, then for all $t > 0$ $\mathsf{P}(\hat{X}_t = \mathcal{G} \mid \tilde{X}_t = \tilde{\mathcal{G}}) = \frac{1}{|\tau^{-1}(\tilde{\mathcal{G}})|}$ (completeness).

The Algorithm 2 works as follows: every node appearing on the lhs of the rule simply renames the node type with a corresponding node type from $\tilde{\mathcal{A}}$. Due to Lemma 1, if the interface of the node is not empty, there will be a unique partition class which covers it, and thus, simple renaming is sufficient. Thus, applying the Algorithm 2 to the $\mathcal{A}$-rules in Figure 1.a gives the $\tilde{\mathcal{A}}$-rules in Figure 1d.

The case when the interface of the observed node is empty, or if a node is deleted by the rule need to be treated separately. If the node's interface is empty, it is then translated to a node with an arbitrary equivalence class (the choice has no impact on the dynamics of the reduced system). The case of deletion is more subtle. Deleting a node in an original rule-set should reflect in deleting one node in *each* equivalence class of that node's signature. Indeed, if a node is consumed by a (new, translated) rule, one node per all other partition classes must be picked uniformly at random and then removed. Such a situation can be resolved by adding fictitious nodes in the left hand side of reduced rules, each fictitious node having an empty interface. Yet, the rate of rules must be corrected by dividing the rate constant by the number of nodes of this type in the mixture. Thus the reduced system does not strictly follow the mass action law, but it uses the functional rates that depend on the state of the system). Such rates are enabled in the KaSim simulator (Krivine, J. 2012).

4 RECONSTRUCTING THE SPECIES-BASED DYNAMICS

In this section, we introduce the species-based dynamics and the fragment-based dynamics of a rule-based model. Both dynamics are illustrated on a running example, introduced in the previous section. Species-based dynamics, denoted by $\hat{X}|_{\mathcal{S}}$, is a CTMC which takes values in multi-sets of species, that is, $\mathcal{X} \subset \mathbb{N}^s_{\geq 0}$. It is worth mentioning that the species-based dynamics of a given rule-based model over the set of node types $\mathcal{A}$ coincides with the CTMC underlying the chemical reaction network with mass-action kinetics, defined in (1): $\hat{X}|_{\mathcal{S}} \equiv X$. Fragment-based dynamics, denoted by $Y \equiv \hat{Y}|_{\tilde{\mathcal{S}}}$, is a CTMC which takes values in multi-sets of fragments, that is, $\tilde{\mathcal{S}} \subset \mathbb{N}^m_{\geq 0}$. It is defined as the species-based dynamics of the translated rule-based model to a corresponding set of $\tilde{\mathcal{A}}$-species. In the following, we show how the two dynamics relate.

Definition 9 (Species-based dynamics) Let $\mathcal{R}$ be a rule-based model with a CTMC $\hat{X}$, and $\mathcal{S} = \{S_1, \dots, S_s\}$ a set of $\mathcal{A}$-species. The projection function $\varphi : \mathbb{G} \to \mathcal{X} \subset \mathbb{N}^s_{\geq 0}$ maps a reaction mixture to a multi-set of $\mathcal{A}$-species:

$$\varphi(\mathcal{G}) := (x_1, \dots, x_s), \text{ with } x_i = |Emb(S_i, \mathcal{G})|/|Aut(S_i)|. \tag{4}$$

The function φ maps more reaction mixtures to a same multi-set of species, and thus partitions the state space $\mathbb{G}$. Conversely, let $\mathbf{x} \in \mathcal{X}$, and let a_i denote the number of nodes of type A_i in some site-graph $\mathcal{G}$, such that $\varphi(\mathcal{G}) = \mathbf{x}$:

$$a_i := \sum_{j=1}^{s} \{k \cdot x_j \mid S_i \text{ contains } k \text{ nodes of type } A_j \ , i = 1, \dots, N\}. \tag{5}$$

Let $M = \sum_{i=1}^{s} a_i$, and fix a set of node labels $V = \{v_1, \dots, v_M\}$. Denote by $\varphi^{-1}(\mathbf{x})$ the set of reaction mixtures with set of node labels V, projected to $\mathbf{x}$: $\varphi^{-1}(\mathbf{x}) := \{\mathcal{G} \mid \varphi(\mathcal{G}) = \mathbf{x} \text{ and the set of node labels of } \mathcal{G} \text{ is } V\}$.

The species-based dynamics of $\hat{X}$ is a stochastic process $\hat{X}|_{\mathcal{S}}$ over the state space $\mathcal{X} \subset \mathbb{N}^{s}_{\geq 0}$, defined by: $[(\hat{X}|_{\mathcal{S}} = \mathbf{x})$ iff $(\hat{X} \in \varphi^{-1}(\mathbf{x}))]$.

In the running example of the Figure 1, the set of species for the rule-based model is given by $\mathcal{S} = \{S_A, S_B, S_C, S_{AB}, S_{BC}, S_{ABC}\}$. For example, a reaction mixture $\mathcal{G}$ presented in Figure 1e is projected to a multi-set $\varphi(\mathcal{G}) = (1,2,0,0,0,1)$.

Definition 10 (Fragment-based dynamics) Let $\mathcal{R}$ be a rule-based model with a CTMC and $\tilde{\mathcal{R}}$ its translation by Algorithm 2. Let $\tilde{\mathcal{S}} = \{\tilde{S}_1, \tilde{S}_2, \ldots\}$ denote the set of $\tilde{\mathcal{A}}$-species. The projection of $\hat{Y}$ to the set of $\tilde{\mathcal{A}}$-species by a function $\tilde{\varphi} : \mathbb{G} \to \mathcal{Y} \subseteq \mathbb{N}^{m}_{\geq 0}$ is denoted by $Y \equiv \tilde{Y}|_{\tilde{\mathcal{S}}}$, and we refer to it as fragment-based dynamics of $\mathcal{R}$.

In our running example, the set of fragments for the rule-based model if $\tilde{\mathcal{S}} = \{\tilde{S}_{Ba}, \tilde{S}_{Bc}, \tilde{S}_{AB?}, \tilde{S}_{CB?}\}$ (the question mark '?' in the species name $\tilde{S}_{ABa?}$ denotes the fact that we do not care whether the node of type B is bound to a node of type C, or not), and the reaction mixture $\tilde{\mathcal{G}}$, depicted in the Figure 1f, is projected to a multi-set $\tilde{\varphi}(\tilde{\mathcal{G}}) = (2,2,1,1)$.

In order to relate the species-based and fragment-based dynamics, we need to first answer the question: Given a multi-set of $\mathcal{A}$-species, how many $\mathcal{A}$-reaction mixtures are lumped to it?

Theorem 2 Let $\mathbf{x}$ be a multiset of $\mathcal{A}$-species: $\mathcal{S} = \{S_1, \ldots S_s\}$, where $\mathcal{A} = \{A_1, \ldots, A_N\}$. Then,

$$|\varphi^{-1}(\mathbf{x})| = \left(\frac{a_1! a_2! \ldots a_N!}{\prod_{i=1}^{s} (x_i! |Aut(S_i)|^{x_i})} \right), \tag{6}$$

where a_i denotes the number of nodes of type A_i in a site-graph $\mathcal{G} \in \varphi^{-1}(\mathbf{x})$, defined in (5).

The proof is available in the preprint version of this manuscript (Petrov, Feret, and Koeppl 2012).

The following result shows that the transient probability distribution of the process X can be deduced from the transient probability distribution of the rule-based model $\tilde{\mathcal{R}}$ and by adequately controlling the initial distributions (Feret et al. 2012).

Theorem 3 Let $f : \mathcal{Y} \times \mathcal{X} \to [0,1]$ be defined by:

$$f(\mathbf{y}, \mathbf{x}) = \begin{cases} \frac{|\varphi^{-1}(\mathbf{x})|}{|\tilde{\varphi}^{-1}(\mathbf{y})|}, & \text{if } \varphi^{-1}(\mathbf{x}) \cap \tilde{\varphi}^{-1}(\mathbf{y}) \neq \varnothing \\ 0, & \text{otherwise.} \end{cases}$$

Then, $\mathsf{P}(X_t = \mathbf{x} \mid Y_t = \mathbf{y}) = f(\mathbf{y}, \mathbf{x})$ holds for all $t \geq 0$ if it holds for $t = 0$.

The expression given in (6) can be used to compute $f(\mathbf{y}, \mathbf{x})$, by first applying it for a set of $\mathcal{A}$-species and a state $\mathbf{x}$, and then by applying it for the set of $\tilde{\mathcal{A}}$-species and a state $\mathbf{y}$. The proof relies on the concept of Markov chain lumpability (Kemeny and Snell 1960), whose extension to continuous time we present in (Petrov et al.). We omit the proof here due to the space limitations, and since its technical connotation is orthogonal to the one of this paper. The references (Feret et al. 2010), (Feret et al. 2012), or (Petrov et al. 2012) contain the proof of the Theorem 3 in the corresponding formalisms.

We may now illustrate the Theorem 3 on the running example. The number of automorphisms of $\mathcal{A}$-site-graphs $\mathcal{G}$ in Figure 1e is $|\tau^{-1}(\mathcal{G})| = \frac{3!1!1!}{2!1!} = 3$, and the number of automorphisms of $\tilde{\mathcal{A}}$-site-graph $\tilde{\mathcal{G}}$ is $|\tilde{\tau}^{-1}(\tilde{\mathcal{G}})| = \frac{3!3!1!1!}{2!2!1!1!} = 9$. The Theorem 3 tells us that the probability of being in state $\mathcal{G}$, conditioned on being in the state $\tilde{\mathcal{G}}$, is $1/3$, and it remains constant, if it is such at the initial state.

Related works on stochastic fragmentation contain more examples and case studies which demonstrate the usage of the Theorems 1 and 2, and the fragments methodology overall. At this point, it is worth re-mentioning the contribution of this paper in the context of the previous ones. In the previous works, the reconstruction quantities are computed specifically for the case studies presented therein, but for states with conveniently small copy numbers of species, where computing graph-automorphisms was possible to do by hand. In this work, we developed a general algorithm for computing the reconstruction quantities for arbitrarily complex fragmentation structures, and for arbitrarily large abundances of fragments and species. Accordingly, with a purpose to illustrate the efficiency of the algorithm, we construct another example, presented in Section 5.4.

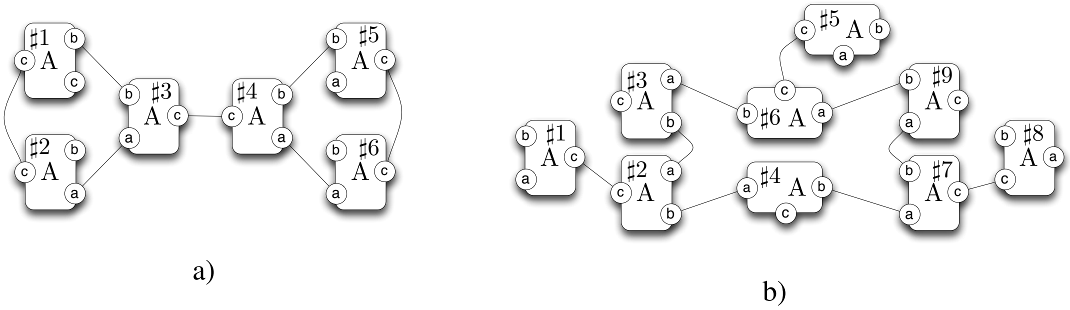

Figure 2: Examples of site-graphs. a) There are two automorphisms: the identity and the involution which maps the node #3 into the node #4; The edge between the node #3 and the node #4 is the gravity center (as defined in the Subsection 5.3.1). b) There are three automorphisms: the identity function, the automorphism which maps the node #2 into the node #7, and the automorphism which maps the node #2 into the node #6; The subgraph containing #2, #3, #4, #6, #7, and #9 is the gravity center (as defined in the Subsection 5.3.1).

5 COUNTING AUTOMORPHISMS IN A CONNECTED SITE-GRAPH

In this section, we address the problem of counting automorphisms in a connected site-graph. We will illustrate our approach on the two examples shown in Figure 2: there are two automorphisms in the site-graph in Figure 2a, and three automorphisms in the site-graph in Figure 2b.

5.1 Rigidity

It is much easier to count the embeddings between two connected site-graphs than between general graphs. For regular graphs, deciding whether there exists an embedding between two graphs (subgraph isomorphism problem), is known to be NP-complete. Indeed, site-graphs enjoy the so-called rigidity property which ensures that an embedding between two connected site-graphs is fully defined by the image of one node.

Theorem 4 (rigidity) Let $G = (V, \text{Type}, S, E, \psi)$ and $G' = (V', \text{Type}', S', E', \psi')$ be two connected site-graphs and let σ_1 and σ_2 be two embeddings between G and G'. Then, for any node $v \in V$, we have: $[\sigma_1^*(v) = \sigma_2^*(v) \implies \sigma_1 = \sigma_2]$.

The proof is available in the preprint version of this manuscript (Petrov, Feret, and Koeppl 2012).

5.2 Counting Embeddings

A direct consequence of Theorem 4 is that, given two connected sites-graphs $G = (V, \text{Type}, S, E, \psi)$ and $G' = (V', \text{Type}', S', E', \psi')$, the number of embeddings between G and G' is bounded by the number of nodes in G'. Moreover, for a fixed node $v \in V$, there is a bijection between the set of embeddings $Emb(G, G')$ between G and G' and the set of the images of this node by any embedding between G and $G' - \{\sigma^*(v) \mid \sigma \in Emb(G, G')\}$.

We can check that the automorphisms of the site-graph in the Figure 2a are obtained by mapping the node #3 either to the node #3 or to the node #4, whereas the automorphisms of the site-graph in the Figure 2b are obtained by mapping the node #2 into the node #2, to the node #6, or to the node #7.

Given a pair of nodes $(v, v') \in V \times V'$, deciding whether there exists an embedding $\sigma \in Emb(G, G')$ such that $\sigma^*(v) = v'$ can be done in a time that is linear with respect to the number of sites in $G = (V, \text{Type}, S, E, \psi)$. We describe a procedure in Algorithm 3. The algorithm consists in exploring the graphs G and G' in parallel, starting respectively from the node v and the node v'. At each step, we check if the node in v is less specific than v'. For this purpose we use a primitive $\leq$ to check whether a node v is less specific than a node v'. More formally, $\leq$ is a binary relation which relates any pair of nodes $(v, v') \in V \times V'$ such that:

Algorithm 3: *testEmbedding*: testing whether there exists an embedding which maps the node v to the node v' in a site-graph.

The algoirthm is omitted due to space limitations. It is available online in the preprint version of this article (T. Petrov and J. Feret and H. Koeppl 2012).

(i) v and v' have the same type ($\text{Type}(v) = \text{Type}'(v')$), (ii) each site documented in v is also documented in v' (for all $s \in S(v)$, $s \in S'(v')$), and (iii) documented sites are evaluated by the same internal states (if $\psi(v,s)$ is defined, then $\psi'(v',s)$ is defined and $\psi'(v',s) = \psi(v,s)$).

In the site-graph depicted in Figure 2a, the call of the Algorithm 3 to the pair of nodes #3 and #4 will succeed and form the support function $[\#3 \mapsto \#4, \#2 \mapsto \#6, \#1 \mapsto \#5, \#4 \mapsto \#3, \#6 \mapsto \#2, \#5 \mapsto \#1]$. Moreover, in the site-graph in Figure 2b, the call of the Algorithm 3 ensures that there is no automorphism mapping the node #2 to the node #4, since the node #4 cannot be less specific that the node #2 (one has the site c free, whereas the other does not); but the call of the Algorithm 3 to the pair of nodes #2 and #7 ensures that there is an automorphism mapping the node #2 to the node #7 with the following support function $[\#2 \mapsto \#7, \#3 \mapsto \#4, \#6 \mapsto \#2, \#9 \mapsto \#3, \#7 \mapsto \#6, \#4 \mapsto \#9, \#8 \mapsto \#5, \#5 \mapsto \#1, \#1 \mapsto \#8]$.

A drawback of the Algorithm 3 is that it may waste some time before discovering that there is no embedding mapping the node v to the node v'. Nevertheless, it is possible to tune the algorithm by refining the binary relation $\leq$, so that it checks whether the k-radius of the node v is a subgraph of the k-radius of the node v' instead (the k-radius of a node v in a subgraph G is the subgraph that contains all the nodes v' of G for which there exists a path between v and v' of length less than k). By increasing the radius k, the failure is detected earlier, but, each check $v \leq v'$ becomes more costly.

5.3 Counting Automorphisms

Now we consider counting the number of automorphisms in the connected site-graph G, that is, an embedding of G to itself. Curiously, for regular graphs, the graph isomorphism problem (deciding whether there exists an isomorphism between two graphs) is one of the only two problems, which are known to be in NP, but not known whether they are actually NP-hard or P-hard (Garey and Johnson 1990) (the second such problem is the integer factorization).

5.3.1 Gravity Center

Each site-graph G which has at least two automorphisms has a subgraph G', called the *gravity center* of G, which is preserved by any of its automorphisms (for any automorphism σ of G and any node v of G', $\sigma^*(v)$ is a node of G'). Such a subgraph is either a strongly connected component (strongly connected components are formally defined in Def. 14 as a connected subgraph which remains connected if we remove an edge of it, whichever edge we choose to remove) or an edge which separates the graph into two connected components. In the example of the Figure 2a, the gravity center is indeed the edge between the node #3 and the node #4, whereas in the example of the Figure 2b, the gravity center is the strongly connected component containing the nodes #2, #3, #4, #6, and #7. As a consequence, it is enough to detect the set of automorphisms of the gravity center of G and check which of them are also automorphisms of G. We provide an algorithm to compute the gravity center of a site-graph. Whenever the site-graph has only one automorphism, the algorithm returns a failure (in which case, we know that there is only one automorphism), or an arbitrary strongly connected component (which is also sound, since it is preserved by each automorphism of the initial graph).

Definition 11 (separating edge) Let $G = (V, \text{Type}, S, E, \psi)$ be a connected site graph and $e \in E$ be an edge in G. We say that e is separating the graph G if and only if the site-graph $(V, \text{Type}, S, E \setminus \{e, \overline{e}\}, \psi)$ is not connected. The set of separating edges of the graph G is denoted by $\mathcal{E}(G)$.

We notice that a site-graph G contains a cycle if and only if it contains an edge in G which is not separating.

Algorithm 4: *findGravityCenter*: compute the gravity center of a site-graph G if this one has more than two automorphisms, otherwise, it returns a failure or outputs a strongly connected component.

```
Input  :  A site-graph $G = (V, \mathrm{Type}, S, E, \psi)$.
Output: The gravity center if there $G$ has at least two automorphisms, return a failure or an arbitrary strongly connected
        component otherwise.

for any separating edge $((v,s),(v',s')) \in \mathcal{E}(G)$ do
    if Type$(v)$ = Type$(v')$ and $s = s'$ and testEmbedding$(G,G,v,v')$ then
        return $(\{v,v'\}, \mathrm{Type}, [v \mapsto \{s\}, v' \mapsto \{s\}], \{((v,s),(v',s'));((v',s'),(v,s))\}, \varnothing)$;

$G_0 = G$;
for any separating edge $e = ((v,s),(v',s)) \in \mathcal{E}(G)$ do
    $G_1, G_2 = (V, \mathrm{Type}, S, E \setminus \{e, \bar{e}\}, \psi)$;
    if $G_1$ and $G_2$ have the same number of nodes then return Failure;
    if $G_1$ has more nodes than $G_2$ then
        $G_0 = G_0 \cap G_1$;
    else
        $G_0 = G_0 \cap G_2$;

return $G_0$;
```

Definition 12 (strongly connected site-graph) We say that a site-graph is strongly connected if and only if it has no separating edges.

Definition 13 (subgraph) Let $G = (V, \mathrm{Type}, S, E, \psi)$ and $G' = (V', \mathrm{Type}', S', E', \psi')$ be two site-graphs. We say that G' is a subgraph of G if and only if the following properties are all satisfied: (i) $V' \subseteq V$, (ii) Type$'$ is a restriction of Type, (iii) $S' \subseteq S$, (iv) $E' \subseteq E$, and (v) ψ' is a restriction of ψ.

Definition 14 (strongly connected component) Let $G = (V, \mathrm{Type}, S, E, \psi)$ be a site-graph. The graph $G' = (V, \mathrm{Type}, \mathcal{S}, E \setminus \mathcal{E}(G), \psi)$ is called the decomposition of the site-graph G to strongly connected components.

Moreover, each connected component in G' is called a strongly connected component of G.

The site-graph in the Figure 2a has only one separating edge between the nodes #3 and #4, and hence contains two strongly connected components: one consists of the nodes #1, #2, and #3, and another one consists of the nodes #4, #5, and #6. The site-graph in the Figure 2b has three separating edges (between the nodes #1 and #2, the nodes #5 and #6, and #7 and #8, which gives four strongly connected components: three with a single node -#1, #5, #8, and one with the other nodes -#2, #3, #4, #6, and #7).

Theorem 5 (gravity center) Let $G = (V, \mathrm{Type}, S, E, \psi)$ be a site-graph. Then one of the following assertions is satisfied:

1. there is only one automorphism of G;
2. there are exactly two automorphisms of G and there is one separating edge $((v,s),(v',s)) \in E$ such that: one automorphism is the identity and the other one is an involution which maps the node v into the node v' (and conversely);
3. there is a strongly connected component in G, which is preserved by all the automorphisms of G.

The proof can be found in the preprint version of this manuscript (Petrov, Feret, and Koeppl 2012).

It follows that an acyclic graph has at most two automorphisms. Moreover, whenever a site-graph G (acyclic, or not) has at least two automorphisms and $e = ((v,s),(v',s)) \in \mathcal{E}(G)$is a separating edge, then e separates G either into two subgraphs with the same number of nodes, or into two subgraphs G_1 and G_2 such that G_1 has more nodes than G_2. In the first case, the subgraph $(\{v,v'\}, \mathrm{Type}, [v \mapsto \{s\}, v' \mapsto \{s\}], \{((v,s),(v',s'));((v',s'),(v,s))\}, \varnothing)$ is the gravity center of G, and in the second case, G_1 is the gravity center of G. This concludes the proof that the Algorithm 4 computes the gravity center of a site-graph whenever this graph has more than two automorphisms, and returns a failure or an arbitrary strongly connected component otherwise.

We can apply the Algorithm 4 to our two favorite site-graphs. The separating edge of the site-graph in the Figure 2a is indeed its gravity center, and hence the site-graph has exactly two automorphisms. The site-graph in the Figure 2b has a gravity center that is a strongly connected component containing the nodes #2, #3, #4, #6, #7, and #9.

5.3.2 The Group of Automorphisms

For testing the existence of an automorphism, it is also possible to avoid checking all the pairs of nodes $v, v' \in V$ which satisfy $v \leq v'$, by using the algebraic structures of the set of automorphisms. Indeed, the set of all automorphisms of a graph forms a group with respect to the composition operator. Consequently, the following two properties hold.

Property 1 If σ_1 and σ_2 are two automorphisms of a site-graph G, then the automorphism $\sigma_2 \circ \sigma_1$ maps its node v to a node $[\sigma_2^* \circ \sigma_1^*](v)$.

Property 2 Let σ be an automorphism of a site-graph G. Let v, v' be its nodes, such that there is no automorphism mapping v to v'. Then, there is also no automorphism mapping v to $\sigma^*(v')$.

Getting back to the site-graph in Figure 2b, from the property 1, we conclude that there is an automorphism mapping the node #2 to the node #6. Thanks to the property 2, we can show that there is no automorphism mapping the node #2 to either the node #9 or the node #3. As a conclusion, there are exactly three automorphisms.

6 CONCLUSIONS

The work presented in this paper concludes the framework for exact reductions of stochastic rule-based models, following (Feret et al. 2012), (Camporesi et al. 2010), (Petrov et al. 2012), to name a few. The focus is on deriving efficient procedures for what was only implicitly defined in previous works. We here show (i) how to derive the reduced, fragment-based model, and, more interestingly, (ii) how to effectively reconstruct the species-based dynamics by simulating only the reduced model. For the latter, we encounter and successfully deal with the problem of counting site-graph automorphisms.

Using fragments is a powerful automated reduction technique, shown to detect sometimes exponential reduction of the size of the underlying state space (Petrov et al. 2012). One obvious future direction of this line of research is to examine how the reduction performs on real case studies. Another aspect is that we dealt here with the *exact* reductions, with the mechanistic rule-based model being the correct (reference) system. Since it is rarely the case that mutual conditioning of sites never occurs, the exact reduction may give no reduction at all. Moreover, every biological model is already an approximation of reality, and seeking for exact reductions with respect to them may thus be both inefficient and misleading. This all motivates to design *approximate* reduction techniques instead.

ACKNOWLEDGEMENTS

Jérôme Feret's contribution was partially supported by the AbstractCell ANR-Chair of Excellence. The work of Tatjana Petrov and Heinz Koeppl is supported by the Swiss National Science Foundation (SNF grant number PP00P2 128503/1) and by SystemsX.ch (the Swiss Initiative for Systems Biology) within an IPhD project. All authors thank to Marica Stojanov, for the enthusiastic and insightful discussions on the topic.

REFERENCES

Anderson, D. F., and T. G. Kurtz. 2011. "Continuous time Markov chain models for chemical reaction networks". In *Design and Analysis of Biomolecular Circuits*, edited by H. Koeppl, G. Setti, M. di Bernardo, and D. Densmore. Springer-Verlag.

Blinov, M. L., J. R. Faeder, and W. S. Hlavacek. 2004. "BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains". *Bioinformatics* 20:3289–3292.

Borisov, N. M., N. I. Markevich, B. N. Kholodenko, and E. D. Gilles. 2005. "Signaling through Receptors and Scaffolds: Independent Interactions Reduce Combinatorial Complexity". *Biophysical Journal* 89.

Buchholz, P. 1994. "Exact and Ordinary Lumpability in Finite Markov Chains". *Journal of Applied Probability* 31:59–75.

Camporesi, F., J. Feret, H. Koeppl, and T. Petrov. 2010, September. "Combining Model Reductions". In *the 26th Conference on the Mathematical Foundations of Programming Semantics - MFPS 2010*, edited by M. Mislove and P. Selinger, Volume 265 of *Electronic Notes in Theoretical Computer Science*, 73–96. Ottawa, Canada: Elsevier.

Cormen, T. H., C. Stein, R. L. Rivest, and C. E. Leiserson. 2001. *Introduction to Algorithms, Chapter 21: Data structures and Disjoint Sets*. 2nd ed. McGraw-Hill Higher Education.

Danos, V., J. Feret, W. Fontana, R. Harmer, and J. Krivine. 2010. "Abstracting the Differential Semantics of Rule-Based Models: Exact and Automated Model Reduction". In *Proceedings of the 2010 25th Annual IEEE Symposium on Logic in Computer Science*, LICS '10, 362–381. Washington, DC, USA: IEEE Computer Society.

Danos, V., and C. Laneve. 2004. "Formal molecular biology". *Theoretical Computer Science* 325 (1): 69–110.

Feret, J., V. Danos, J. Krivine, R. Harmer, and W. Fontana. 2009, April. "Internal coarse-graining of molecular systems". *Proceedings of the National Academy of Sciences* 106 (16): 6453–6458.

Feret, J., T. Henzinger, H. Koeppl, and T. Petrov. 2012. "Lumpability abstractions of rule-based systems". *Theoretical Computer Science* 431 (0): 137 – 164. Modelling and Analysis of Biological Systems Based on papers presented at the Workshop on Membrane Computing and Bio-logically Inspired Process Calculi (MeCBIC) held in 2008 (Iasi), 2009 (Bologna) and 2010 (Jena).

Feret, J., H. Koeppl, and T. Petrov. 2010. "Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models". *Int. Jour. of Software and Informatics*. to appear.

Friedman, N., L. Cai, and X. S. Xie. 2010. "Stochasticity in Gene Expression as Observed by Single-molecule Experiments in Live Cells". *Israel Journal of Chemistry* 49:333–342.

Garey, M. R., and D. S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Gillespie, D. T. 2007. "Stochastic simulation of chemical kinetics". *Annu Rev Phys Chem* 58 (1): 35–55.

Harmer, R., V. Danos, J. Feret, J. Krivine, and W. Fontana. 2010. "Intrinsic Information carriers in combinatorial dynamical systems". *Chaos* (20): 037108.

Hlavacek, W. S., J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana. 2006. "Rules for Modeling Signal-Transduction Systems". *Science's STKE* 2006 (344).

Kemeny, J., and J. L. Snell. 1960. *Finite Markov Chains*. Van Nostrand.

Krivine, J. 2008-2012. "KaSim: a simulator for Kappa". http://www.kappalanguage.org.

T. Petrov and J. Feret and H. Koeppl 2012. "Reconstructing species-based dynamics from reduced stochastic rule-based models". Preprint containing proofs and algorithms, available at https://edit.ethz.ch/bison/Publications/archive/Petrov_WSC_2012.

Petrov, T., A. Ganguly, and H. Koeppl. "Markov chain aggregation and its applications to combinatorial reaction networks". BISON group, ETH Zurich, in preparation.

Petrov, T., A. Ganguly, and H. Koeppl. 2012. "Model Decomposition and Stochastic Fragments". *Electronic Notes in Theoretical Computer Science* 284 (0): 105 – 124. ¡ce:title¿Proceedings of the 2nd International Workshop on Static Analysis and Systems Biology (SASB 2011)¡/ce:title¿.

Rubino, G., and B. Sericola. 1991. "A Finite Characterization of Weak Lumpable Markov Processes. Part II: The Continuous Time Case". *Stochastic processes and their applications* vol. 38, no2:195–204.

Rubino, G., and B. Sericola. 1993. "A Finite Characterization of Weak Lumpable Markov Processes. Part I: The Discrete Time Case". *Stochastic processes and their applications* vol. 45, no 1:115–125.

Yu, J., J. Xiao, X. Ren, K. Lao, and X. S. Xie. 2006. "Probing gene expression in live cells, one protein molecule at a time.". *Science* 311 (5767): 1600–3.

AUTHOR BIOGRAPHIES

TATJANA PETROV is a doctoral student at the Automatic Control Lab, Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland. Her research interests are in using formal methods in systems biology, in particular for modeling signaling pathways in cells. Her email address is tpetrov@ethz.ch.

JEROME FERET is a research fellow (CR1) at INRIA, in the ABSTRACTION project-team. He is mainly interested in the static analysis of critical embedded software and biological models by the means of abstract interpretation. His email address is feret@ens.fr.

HEINZ KOEPPL is an assistant professor at the Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland. He is interested in the application of control and system theoretical concepts to computational biology. In particular, he currently works on statistical inference and model reduction in the context of stochastic intra-cellular reaction dynamics. His email address is koeppl@ethz.ch.