Forward and Backward Bisimulations for Chemical Reaction Networks

Luca Cardelli[1], Mirco Tribastone[2], Max Tschaikowski[3], and Andrea Vandin[4]

1   **Microsoft Research & University of Oxford, UK**
     `luca@microsoft.com`
2-4 **University of Southampton, UK**
     `{m.tribastone,m.tschaikowski,a.vandin}@soton.ac.uk`

―――― **Abstract** ――――

We present two quantitative behavioral equivalences over species of a chemical reaction network (CRN) with semantics based on ordinary differential equations. *Forward CRN bisimulation* identifies a partition where each equivalence class represents the exact sum of the concentrations of the species belonging to that class. *Backward CRN bisimulation* relates species that have the identical solutions at all time points when starting from the same initial conditions. Both notions can be checked using only CRN syntactical information, i.e., by inspection of the set of reactions. We provide a unified algorithm that computes the coarsest refinement up to our bisimulations in polynomial time. Further, we give algorithms to compute quotient CRNs induced by a bisimulation. As an application, we find significant reductions in a number of models of biological processes from the literature. In two cases we allow the analysis of benchmark models which would be otherwise intractable due to their memory requirements.

1   Introduction

At the interface between computer science and systems biology is the idea that biological systems can be interpreted as computational processes [20, 11], leading to a number of formal methods applied to study biomolecular systems [5, 16, 22]. In this context, chemical reaction networks (CRNs), a popular mathematical model of interaction in natural sciences, can also be seen as a kernel concurrent language for natural programming.

In this paper we present, for the first time to our knowledge, quantitative bisimulation equivalences for CRNs with the well-known interpretation based on ordinary differential equations (ODEs). (To make the paper self-contained, all background is given in Section 2.) In this semantics, each species is associated with an ODE giving the deterministic evolution of its concentration starting from an initial condition. Our bisimulations are equivalences over species that induce a reduced CRN that exactly preserves the dynamics of the original one. This is an important goal, especially in order to cope with the potentially very large number of species and reactions in many biological networks [14, 15].

We study two equivalences, developed in the Larsen-Skou style of probabilistic bisimulation [26], that are based on two distinct ideas of observable behavior. *Forward CRN*

bisimulation yields an aggregated ODE where the solution gives the exact sum of the concentrations of the species belonging to each equivalence class. In *backward CRN bisimulation*, instead, equivalent species have the same solution *at all time points*; in other words, backward CRN bisimulation relates species whose ODE solutions are equal whenever they start from identical initial conditions. The use of "forward" and "backward" has a long tradition in models of computation based on labelled transition systems [17]. In the case of quantitative variants, for instance those defined for process algebra with a continuous-time Markov chain (CTMC) semantics [23, 24, 8, 4], forward bisimulations are equivalences that induce a CTMC aggregation in the sense of ordinary lumpability [7], where the probability of an equivalence class is equal to the sum of the probabilities of the states belonging to that class. This is found by checking conditions on the *outgoing transitions* of related states in the transition diagram. A backward bisimulation induces a CTMC aggregation in the sense of weak lumpability [19], where all states in the same equivalence class have a time-invariant conditional probability distribution; exact lumpability is a special case where the conditional probability distribution is uniform, in the sense that any two states of each equivalence class have the same probability at any point in time whenever they have the same initial probabilities. It is found by relating states according to conditions on their *predecessor states* [19, 31, 7].

Despite being similar in spirit, technically our bisimulations are fundamentally different for two reasons. First, they concern a continuous-state semantics based on ODEs instead of a discrete-state CTMC. Second, they operate at the structural, syntactical level, because they are defined with quantities that can be computed by only inspecting the reactions of a CRN. Still, the repercussions of our bisimulations on the semantics are explained according to certain theories of aggregation. In particular, forward CRN bisimulation yields an aggregated system in the sense of ODE lumpability [33, 27]. This theory covers linear transformations of the original state variables in general; here we consider an instance, which we call *ordinary fluid lumpability*, where the transformation is induced by a partition of state variables. (Forward bisimulation is presented in Section 3.1.) Backward bisimulation (presented in Section 3.2) is related to *exact fluid lumpability*, introduced in the context of process algebra with fluid semantics [34], identifying process terms with the same ODE solution when initialized equally. The disadvantage of forward CRN bisimulation is that it is lossy (yet exact) because, similarly to the forward stochastic analogues, from the aggregated ODE system in general it is not possible to recover the solutions for the individual species within the same equivalence class. On the other hand, it does not impose restrictions on the initial conditions, which instead are present in our backward variant. As a further important difference, forward CRN bisimulation (again, like its stochastic analogues) turns out to be a sufficient condition for ODE lumpability. Instead, backward CRN bisimulation enjoys a full characterization, in the sense that there exists a backward CRN bisimulation between two species if and only if they have the same ODE solutions (provided that they start from equal initial conditions). More in general, by means of a number of examples we will show that the two equivalences are complementary because not comparable. In other words, there exist models that can be reduced up to forward CRN bisimulation but not by the backward variant, and vice versa; at the same time, there are models that can be reduced by both.

To enhance the usefulness of these notions, we present (in Section 5) a *template* partition-refinement algorithm that is parametric with respect to the equivalence of interest, computing the coarsest refinement up to either variant in polynomial time. To use our equivalences as an automatic model reduction tool, we further give two algorithms (in Section 4) that provide the quotient CRN induced by either bisimulation. With a prototype implementation available at `http://sysma.imtlucca.it/crnreducer/`, we show (in Section 6) that we are able to

reduce a number of case studies taken from the literature. Our bisimulations yielded quotient CRNs with number of reactions and species up to four orders of magnitude smaller than the original CRNs, leading to speed-ups in the ODE solution runtimes of up to five orders of magnitude. In two cases, it was possible to analyze models that were otherwise intractable directly within our experimental environment due to excessive memory requirements.

Related work. Behavioral equivalences have been recently proposed in [29] for comparing CRNs; however, the analysis is carried out at the qualitative level, i.e., ignoring the dynamical evolution. In [34] is introduced the notion of *label equivalence* for process algebra with fluid semantics, which captures exact fluid lumpability (processes are equivalent whenever their ODE solutions are equal at all time points). However, unlike backward CRN bisimulation, label equivalence is only a sufficient condition for ODE reduction. Indeed, it works at a coarser level of granularity as it relates *sets* of ODE variables, each corresponding to the behavior of a sequential process. Instead, backward CRN bisimulation relates individual ODE variables. Further, the conditions for equivalence, specific to the process algebra, are difficult to check automatically because of the universal quantifiers over the ODE variables. More important, no algorithm for computing the coarsest partition was developed. Similar considerations apply to the process-algebra specific ordinary fluid lumpability in [35].

Cardelli's notion of *emulation* between two CRNs is a (structural) mapping of species and reactions that, like backward CRN bisimulation, guarantees the equality between the ODE solutions at all time points [10]. An emulation requires a source and a target CRN — the modeler is intended to have the suspicion that, for some given CRN, another CRN might be related to it. But emulation cannot be used when one wants to discover equivalences between species *within the same given CRN*. Thus, emulation is not useful for model reduction because a-priori information about the structure of a quotient CRN is not available. Furthermore, no algorithm is provided in [10] to find emulations automatically. Since backward CRN bisimulation fully characterizes exact fluid lumpability, it is not difficult to show that backward CRN bisimulation generalizes emulation in the sense that any emulation between two CRNs can be understood in terms of a backward CRN bisimulation over the species of a "union CRN" that contains all the reactions of the two CRNs of interest.

Model reductions for the ODE semantics of CRNs have been studied in related models for biomolecular networks, most notably for rule-based systems such as BioNetGen [5] and the κ calculus [16]. These offer an intensional modeling approach, by providing graph-rewrite rules of interaction instead of a complete enumeration of all chemical reactions involved. *Differential fragments* for κ are self-consistent aggregates found by a static analysis on the model, identifying sums of chemical species for which an ODE system can be explicitly written [15]. In this sense, they are analogous to our CRN bisimulations, but with notable differences. First, fragmentation works directly at the rule-based level. This has the advantage that the analysis is performed on a set of rewrite rules, which is typically much more compact than the fully enumerated CRN. However, fragmentation is domain-specific, hence the model must be conveniently expressed as a biomolecular system (e.g., with complex formation or internal state modification). On the other hand, CRN bisimulations work for a generic language-independent CRN, which however must be explicitly given. Further, unlike CRN bisimulations, fragmentation is performed on a "static" view of the model, without information on the reaction rates. The ODE aggregations of both forward CRN bisimulation and fragmentation introduce loss of information (in contrast to backward CRN bisimulation). But, unlike our forward variant, In fragmentation the same species may be present in more than one fragment. Thus, fragmentation can be seen as a form of *improper lumping* that is not necessarily induced by a partition of the original state-space variables [27]. Overall,

because of these differences, it is not difficult to find models that can be reduced by our CRN bisimulations and not by fragmentation, and vice versa. However, an empirical evaluation suggests that CRN bisimulations and fragmentation may complement each other when they can be both applied, i.e., in the case of rule-based models of biomolecular networks for which an underlying CRN can be fully enumerated. This is presented in detail in Section 6.

2 Background

Notation. We write $A \to B$ and B^A for the functions from A to B. When $f \in A \to B$ and $a \in A$, we set $f_a := f(a)$. Moreover, for any $X \subseteq A$ and $b \in B$, we define $f(X) := \{b \in B \mid \exists a \in X.(f(a) = b)\}$. Sets and multisets are denoted by $\{\ldots\}$ and $\{|\ldots|\}$, respectively. Also, we shall not distinguish among an equivalence relation and the partition induced by it, and shall use the symbol $\sim_{\mathcal{H}}$ to denote the equivalence relation with $\mathcal{H} = S/\sim_{\mathcal{H}}$. Finally, given two partitions $\mathcal{H}_1$ and $\mathcal{H}_2$ of a given set S, we say that $\mathcal{H}_1$ is a *refinement* of $\mathcal{H}_2$ if for any $H_1 \in \mathcal{H}_1$ there exists a (unique) $H_2 \in \mathcal{H}_2$ such that $H_1 \subseteq H_2$.

2.1 Chemical Reaction Networks

Formally, a CRN (S, R) is a pair consisting of a finite set of species S and a finite set of chemical reactions R. A reaction is a triple written in the form $\rho \xrightarrow{\alpha} \pi$, where ρ and π are the multisets of species *reactants* and *products*, respectively, and $\alpha > 0$ is the reaction rate. In particular, we focus on basic chemistry where only *elementary reactions* are considered, where at most two reactants (possibly of the same species) interact. No restrictions are instead imposed on products. Several models found in the literature (including those discussed in Section 6) belong to this class. Also, this is consistent with the physical considerations which stipulate that reactions with more than two reactants are very unlikely to occur in nature [21]. We denote by $\rho(X)$ the multiplicity of species X in the multiset ρ, and by $\mathcal{MS}(S)$ the set of finite multisets of species in S. To adhere to standard chemical notation, we shall use the operator $+$ to denote multiset union, e.g., $X + Y + Y$ (or just $X + 2Y$) denotes the multiset $\{|X, Y, Y|\}$. We may also use X to denote either the species X or the singleton $\{|X|\}$.

The (autonomous) ODE system $\dot{V} = F(V)$ underlying a CRN (S, R) is $F : \mathbb{R}_{\geq 0}^S \to \mathbb{R}^S$, where each component F_X, with $X \in S$ is defined as:

$$F_X(V) := \sum_{\rho \xrightarrow{\alpha} \pi \in R} (\pi(X) - \rho(X)) \cdot \alpha \cdot \prod_{Y \in S} V_Y^{\rho(Y)} \ .$$

This represents the well-known *mass-action* kinetics, where the reaction rate is proportional to the concentrations of the reactants involved. Since the ODE system of a CRN is given by polynomials, the vector field F is locally Lipschitz. Hence, the theorem of Picard-Lindelöf ensures that for any $V(0) \in \mathbb{R}_{\geq 0}^S$ there exists a unique non-continuable solution of $\dot{V} = F(V)$.

▶ **Example 1.** We now provide a simple CRN, (S_e, R_e), with $S_e = \{A, B, C, D, E\}$ and $R_e = \{A \xrightarrow{6} E, B \xrightarrow{6} D, A + B \xrightarrow{2} C, C + D \xrightarrow{5} 2C + D, E + D \xrightarrow{5} 2E + D\}$, which will be used as a running example throughout the paper. Its ODE system is given by

$$\dot{V}_A = -6\,V_A - 2\,V_A\,V_B \qquad \dot{V}_B = -6\,V_B - 2\,V_A\,V_B \qquad \dot{V}_C = 2\,V_A\,V_B + 5\,V_C\,V_D$$
$$\dot{V}_D = 6\,V_B \qquad\qquad\qquad \dot{V}_E = 6\,V_A + 5\,V_E\,V_D$$

In the following, we shall assume that the countable infinite universe of all species is well-ordered with respect to $\sqsubseteq$. We then say that a function $\mu : S \to S$ is a *choice function* of a partition $\mathcal{H}$ of S, if $\mu(X) = \min_{\sqsubseteq} H$ for all $H \in \mathcal{H}$ and $X \in H$. Also, choice functions can be trivially lifted to multisets applying them element-wise, e.g., $\mu(X + Y) = \mu(X) + \mu(Y)$.

2.2 Fluid Lumpability

Ordinary Fluid Lumpability. We start by defining the notion of ordinary fluid lumpability, which is an instance of *ordinary lumpability* for ODEs [33] specialized to CRNs.

▶ **Definition 2** (Ordinary fluid lumpability). Let (S, R) be a CRN, F its vector field, and $\mathcal{H} = \{H_1, \ldots, H_m\}$ a partition of S. Then, $\mathcal{H}$ is *ordinary fluid lumpable* if for all $H \in \mathcal{H}$ there exists a polynomial $\wp_H$ in $|\mathcal{H}|$ variables such that $\sum_{X \in H} F_X(V) = \wp_H(\sum_{X \in H_1} V_X, \ldots, \sum_{X \in H_m} V_X)$ for all $V \in \mathbb{R}^S_{\geq 0}$.

Informally, a partition $\mathcal{H}$ is ordinary fluid lumpable if, for each $H \in \mathcal{H}$, the polynomial $\sum_{X \in H} F_X(V)$ in the variables $\{V_X \mid X \in S\}$ can be rewritten into a polynomial $\wp_H$ in the variables $\{\sum_{X \in H} V_X \mid H \in \mathcal{H}\}$. In particular, if $\mathcal{H}$ is known to be an ordinary fluid lumpable partition of (S, R) and V denotes the solution of $\dot{V} = F(V)$ subject to $V(0) \in \mathbb{R}^S_{\geq 0}$, the solution of the aggregated ODE system $(\dot{W}_{H_1}, \ldots, \dot{W}_{H_m}) = (\wp_{H_1}(W), \ldots, \wp_{H_m}(W))$ with $W_H(0) = \sum_{X \in H} V_X(0)$ is such that $W_H(t) = \sum_{X \in H} V_X(t)$ for all $t \in \mathrm{domain}(V)$.

▶ **Example 3.** Consider the ODEs of (S_e, R_e) of Example 1, and let $\mathcal{H}_O = \{\{A\}, \{B\}, \{C, E\}, \{D\}\}$. By applying a variable renaming consistent with the blocks of $\mathcal{H}_O$, i.e., $V_{CE} = V_C + V_E$, and by exploiting the linearity of the differential operator we get

$$\dot{V}_A = -6V_A - 2V_A V_B \quad \dot{V}_B = -6V_B - 2V_A V_B \quad \dot{V}_{CE} = 2V_A V_B + 6V_A + 5V_D V_{CE} \quad \dot{V}_D = 6V_B$$

That is, we obtained an ODE system in terms of block variables only. ◀

Exact Fluid Lumpability. We extend to CRNs the notion of exact fluid lumpability in [34].

▶ **Definition 4** (Exact fluid lumpability). Let (S, R) be a CRN, F its vector field, and $\mathcal{H}$ a partition of S. We call $V \in \mathbb{R}^S$ *constant on* $\mathcal{H}$ if $V_{X_i} = V_{X_j}$ for all $H \in \mathcal{H}$, and all $X_i, X_j \in H$. Then, $\mathcal{H}$ is *exactly fluid lumpable* if $F(V)$ is constant on $\mathcal{H}$ whenever V is constant on $\mathcal{H}$.

▶ **Example 5.** Consider the ODEs of (S_e, R_e) of Example 1, and let $\mathcal{H}_E = \{\{A, B\}, \{C\}, \{D\}, \{E\}\}$. It is easy to see that A and B have same concentrations at all time points if initialized equally. In these cases, we can replace the ODEs of (S_e, R_e) with the ones aggregated according to $\mathcal{H}_E$, obtained by removing $\dot{V}_B$ and replacing all occurrences of V_B with V_A:

$$\dot{V}_A = -6\,V_A - 2\,V_A\,V_A \quad \dot{V}_C = 2\,V_A\,V_A + 5\,V_C\,V_D \quad \dot{V}_D = 6\,V_A \quad \dot{V}_E = 6\,V_A + 5\,V_E\,V_D$$

That is, we obtained a (lossless) aggregated ODE system written in terms of a variable per block, chosen according to $\sqsubseteq$. ◀

We remark that the above definition expresses exact fluid lumpability in terms of properties of the ODE vector field of a CRN. Instead, in [34] exactly fluid lumpability was defined directly in terms of the desired dynamical property, i.e., that the ODE solutions within any equivalence class be equal at all time points. The following result is a new contribution showing that this dynamical property is fully characterized by the vector-field based definition.

▶ **Theorem 6.** *Let (S, R) be a CRN and F its vector field. A partition $\mathcal{H}$ of S is exactly fluid lumpable if and only if, for any $V(0) \in \mathbb{R}^S_{\geq 0}$ that is constant on $\mathcal{H}$, the underlying solution of $\dot{V} = F(V)$ is such that $V(t)$ is constant on $\mathcal{H}$ for all $t \in domain(V)$.* [1]

---

[1] *Note to reviewers.* The proofs are in the appendix that will be published in the case of acceptance.

3   CRN Bisimulations

Both notions of fluid lumpability given in Section 2 are not convenient to be used directly because they involve a universal quantifier over the (uncountable) state space. We address this problem by providing structural conditions that concern only the reactions of a CRN.

3.1   Forward CRN Bisimulation

We now introduce forward CRN bisimulation, an equivalence on species that will turn out to induce ordinary fluid lumpability. We start with the notions of *reaction* and *production rate*. The former collects the rates at which the concentration of a species X decreases when reacting with a given partner. The latter collects the positive contribution that X exerts to the concentration of a species Y, again when reacting with a certain partner.

▶ **Definition 7** (Reaction rate and production rate)**.** Let (S, R) be a CRN, $X, Y \in S$, and $\rho \in \mathcal{MS}(S)$. The ρ-*reaction rate* of X, and the ρ-*production rate* of Y-elements by X are defined respectively as

$$\mathbf{crr}[X, \rho] := (\rho(X) + 1) \sum_{X+\rho \xrightarrow{\alpha} \pi \in R} \alpha, \qquad \mathbf{pr}[X, \rho, Y] := (\rho(X) + 1) \sum_{X+\rho \xrightarrow{\alpha} \pi \in R} \alpha \cdot \pi(Y)$$

Finally, for $H \subseteq S$ we define $\mathbf{pr}[X, \rho, H] := \sum_{Y \in H} \mathbf{pr}[X, \rho, Y]$.

▶ **Definition 8** (Forward CRN Bisimulation)**.** Let (S, R) be a CRN, $\mathcal{R}$ an equivalence relation over S and $\mathcal{H} = S/\mathcal{R}$. Then, $\mathcal{R}$ is a forward CRN bisimulation (abbreviated FB) if for all $(X, Y) \in \mathcal{R}$, all $\rho \in \mathcal{MS}(S)$, and all $H \in \mathcal{H}$ it holds that

$$\mathbf{crr}[X, \rho] = \mathbf{crr}[Y, \rho] \quad \text{and} \quad \mathbf{pr}[X, \rho, H] = \mathbf{pr}[Y, \rho, H] \tag{1}$$

▶ **Example 9.** Consider $\mathcal{H}_O = \{\{A\}, \{B\}, \{C, E\}, \{D\}\}$ of Example 3. It can be shown that $\mathcal{H}_O$ is an FB, as, e.g., $\mathbf{crr}[C, D] = \mathbf{crr}[E, D] = 5$, and $\mathbf{pr}[C, D, \{C, E\}] = \mathbf{pr}[E, D, \{C, E\}] = 10$.

  We are interested in the coarsest FB, as well as in the coarsest one refining a given initial partition of species.

▶ **Proposition** 10. Let (S, R) be a CRN, I a set of indices, and $\mathcal{R}_i$ an FB for (S, R), for all $i \in I$. The transitive closure of their union $\mathcal{R} = (\bigcup_{i \in I} \mathcal{R}_i)^*$ is an FB for (S, R). In particular, if each R_i is such that $S/\mathcal{R}_i$ refines some partition $\mathcal{G}$ of S, then so does $S/\mathcal{R}$.

▶ **Theorem 11** (Forward bisimulation implies ordinary fluid lumpability)**.** *Let (S, R) be a CRN. Then, $\mathcal{H}$ is an ordinarily fluid lumpable partition of S if $\mathcal{H}$ is an FB of S.*

FB is only a sufficient condition for lumpability, as discussed in the next example. (However, Section 6 shows that FB can be effectively applied to interesting existing models.)

▶ **Example 12.** Consider the CRN $(\{F, G\}, \{F \xrightarrow{\alpha_1} G, G \xrightarrow{\alpha_2} F\})$, having ODEs

$$\dot{V}_F = -\alpha_1 V_F + \alpha_2 V_G \qquad \dot{V}_G = -\alpha_2 V_G + \alpha_1 V_F$$

If $\alpha_1 \neq \alpha_2$, $\mathcal{H}_c = \{\{G, F\}\}$ is not an FB, as $\mathbf{crr}[F, \emptyset] = \alpha_1$ and $\mathbf{crr}[G, \emptyset] = \alpha_2$. Nevertheless, the above ODE system is lumpable. Indeed, by applying the variable renaming consistent with $\mathcal{H}_c$, i.e., $V_{FG} = V_F + V_G$, we get a single ODE for V_{FG}, i.e., $\dot{V}_{FG} = 0$.   ◀

3.2   Backward CRN Bisimulation

We now introduce backward CRN bisimulation, an equivalence on species that will turn out to characterize exact fluid lumpability. We start with the notion of cumulative flux rate, which collects the overall contribution that reactions with a given multiset of reactants ρ exert to the concentration of a species X.

▶ **Definition 13** (Cumulative flux rate). Let (S, R) be a CRN, $X \in S$, $\rho \in \mathcal{MS}(S)$, and $\mathcal{M} \subseteq \mathcal{MS}(S)$. Then, we define

$$\mathbf{fr}(X, \rho) := \sum_{\rho \xrightarrow{\alpha} \pi \in R} (\pi(X) - \rho(X)) \cdot \alpha, \qquad \mathbf{fr}[X, \mathcal{M}] := \sum_{\rho \in \mathcal{M}} \mathbf{fr}(X, \rho).$$

We call $\mathbf{fr}(X, \rho)$ and $\mathbf{fr}[X, \mathcal{M}]$ ρ-*flux rate* and *cumulative $\mathcal{M}$-flux rate* of X, respectively.

▶ **Definition 14** (Backward CRN bisimulation). Let (S, R) be a CRN, $\mathcal{R}$ an equivalence relation over S, $\mathcal{H} = S/\mathcal{R}$ and μ the choice function of $\mathcal{H}$. Then, $\mathcal{R}$ is a backward CRN bisimulation (BB) if for any $(X, Y) \in \mathcal{R}$ it holds that

$$\mathbf{fr}[X, \mathcal{M}] = \mathbf{fr}[Y, \mathcal{M}] \quad \text{for all} \quad \mathcal{M} \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/ \approx_{\mathcal{H}}, \tag{2}$$

where any two $\rho, \sigma \in \mathcal{MS}(S)$ satisfy $\rho \approx_{\mathcal{H}} \sigma$ if $\mu(\rho) = \mu(\sigma)$.

▶ **Example 15.** Consider $\mathcal{H}_E = \{\{A, B\}, \{C\}, \{D\}, \{E\}\}$ of Example 5. We first note that $\{|A|\} \approx_{\mathcal{H}_E} \{|B|\}$, as $\approx_{\mathcal{H}_E}$ relates multisets with same number of $\mathcal{H}_E$-equivalent species. Also, it can be shown that $\mathcal{H}_E$ is a BB, as, e.g., $\mathbf{fr}[A, \mathcal{M}] = \mathbf{fr}[B, \mathcal{M}] = -6$ for $\mathcal{M} = \{\{|A|\}, \{|B|\}\}$.
◀

As for FB, there exists a coarsest BB (that refines a given partition of S).

▶ **Proposition** 16. Let (S, R) be a CRN, I a set of indices, and $\mathcal{R}_i$ a BB for (S, R), for all $i \in I$. The transitive closure of their union $\mathcal{R} = (\bigcup_{i \in I} \mathcal{R}_i)^*$ is a BB for (S, R). In particular, if each R_i is such that $S/\mathcal{R}_i$ refines some partition $\mathcal{G}$ of S, then so does $S/\mathcal{R}$.

We now state the mentioned characterization of exact fluid lumpability in terms of BB.

▶ **Theorem 17** (Backward bisimulation characterizes exact fluid lumpability). *Let (S, R) be a CRN. Then, $\mathcal{H}$ is an exactly fluid lumpable partition of S if and only if $\mathcal{H}$ is a BB of S.*
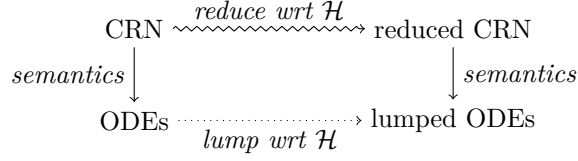
▶ Remark. We wish to stress that FB and BB are not comparable: First, $\mathcal{H}_O$ is not a BB, as $\mathbf{fr}[C, \{A+B\}] = 2$ and $\mathbf{fr}[E, \{A+B\}] = 0$; Second, $\mathcal{H}_E$ is not an FB, as $\mathbf{crr}(A, B) = 2$ and $\mathbf{crr}(B, B) = 0$; Third, for the same reasons, $\{\{A, B\}, \{C, E\}, \{D\}\}$ is neither an FB nor a BB. Similar examples on models of biological relevance are provided in Section 6.   ◀

4   Reduced Chemical Reaction Networks up to CRN Bisimulations

We have shown that, given a CRN and a CRN bisimulation $\mathcal{R}$, we can analyze the aggregated ODE system according to $\mathcal{R}$. We now provide the notion of reduced CRN from which the aggregated ODEs can be directly generated, as depicted in Figure 1.

▶ **Definition 18** (Forward reduction). Let (S, R) be a CRN, $\mathcal{H}$ an FB, and μ its choice function. The $(\mathcal{H}, F)$-*reduction of* (S, R) is given by $(S, R)^{(\mathcal{H}, F)} = (S^{(\mathcal{H}, F)}, R^{(\mathcal{H}, F)})$, where $S^{(\mathcal{H}, F)} = \mu(S)$ and $R^{(\mathcal{H}, F)}$ is defined as follows: (F1) Discard all reactions $\rho \xrightarrow{\alpha} \pi$ such that $\rho \neq \mu(\rho)$; (F2) Replace all remaining reactions $\rho \xrightarrow{\alpha} \pi$ with $\rho \xrightarrow{\alpha} \mu(\pi)$; (F3) Fuse all reactions that have the same reactants and products by summing their rates.

$$\text{CRN} \xrightsquigarrow{\textit{reduce wrt } \mathcal{H}} \text{reduced CRN}$$

$$\left.\textit{semantics}\right\downarrow \qquad\qquad\qquad \left.\downarrow\right.\textit{semantics}$$

$$\text{ODEs} \xdashrightarrow{\textit{lump wrt } \mathcal{H}} \text{lumped ODEs}$$

■ **Figure 1** Relation among ($\mathcal{H}$-reduced) CRNs and ($\mathcal{H}$-lumped) semantics, with $\mathcal{H}$ a bisimulation.

The idea underlying forward reduction is to discard all reactions having non-representative reagents, and to replace the products of the remaining reactions with their representatives.

▶ **Example 19.** Consider the FB $\mathcal{H}_O = \{\{A\}, \{B\}, \{C, E\}, \{D\}\}$ used in Example 3. The $(\mathcal{H}_O, F)$-reduction of (S_e, R_e) is (with C being the representative of its block) $S_e^{(\mathcal{H}_O, F)} = \{A, B, C, D\}$, $R_e^{(\mathcal{H}_O, F)} = \{A \xrightarrow{6} C, B \xrightarrow{6} D, A{+}B \xrightarrow{2} C, C{+}D \xrightarrow{5} 2C{+}D\}$. Note that the reaction $E{+}D \xrightarrow{5} 2E{+}D$ is discarded, as E is not a representative species. ◀

We now state that the $(\mathcal{H}, F)$-reduction of an FB $\mathcal{H}$ induces the ODEs aggregated according to $\mathcal{H}$. For example, the $(\mathcal{H}_O, F)$-reduction of (S_e, R_e) induces the ODEs shown in Example 3, if applying the renaming $V_C = V_{CE}$.

▶ **Theorem 20** (Forward reduction induces aggregation). *Let (S, R) be a CRN, $\mathcal{H}$ an FB and μ its choice function. Then, $(S, R)^{(\mathcal{H}, F)}$ is computed in at most $\mathcal{O}\big(|R|\cdot|S|\cdot(\log(|R|)+\log(|S|))\big)$ steps. Crucially, if F is the vector field of (S, R) and $\hat{F}$ the one of $(S, R)^{(\mathcal{H}, F)}$, then $\sum_{X \in H} F_X(V) = \hat{F}_{\mu(Y)}(\sum_{X \in H_1} V_X, \ldots, \sum_{X \in H_m} V_X)$ for all $V \in \mathbb{R}_{\geq 0}^S$, $H \in \mathcal{H}$ and $Y \in H$.*

For the backward reduction, the underlying idea is to keep track only of differential contributions that affect the representative species $\mu(S)$.

▶ **Definition 21** (Backward reduction). Let (S, R) be a CRN, $\mathcal{H}$ a BB, and μ its choice function. The $(\mathcal{H}, B)$-*reduction of* (S, R) is given by $(S, R)^{(\mathcal{H}, B)} = (S^{(\mathcal{H}, B)}, R^{(\mathcal{H}, B)})$, where $S^{(\mathcal{H}, B)} = \mu(S)$ and $R^{(\mathcal{H}, B)}$ is obtained as follows: (B1) Replace all reactions $\rho \xrightarrow{\alpha} \pi$ with $\rho \xrightarrow{\alpha} \tilde{\pi}$ where $\tilde{\pi}(X_i) := \pi(X_i)$ if $X_i \in \mu(S)$ and $\tilde{\pi}(X_i) := \rho(X_i)$ otherwise; (B2) Replace all such obtained reactions $\rho \xrightarrow{\alpha} \pi$ with $\mu(\rho) \xrightarrow{\alpha} \mu(\pi)$; (B3) Fuse all reactions that have the same reactants and products by summing their rates.

▶ **Example 22.** Considering the CRN (S_e, R_e) and the BB $\mathcal{H}_E$, (B1) changes $B \xrightarrow{6} D$ in $B \xrightarrow{6} D{+}B$, and $A{+}B \xrightarrow{2} C$ in $A{+}B \xrightarrow{2} C{+}B$, while (B2) yields $\{A \xrightarrow{6} E, A \xrightarrow{6} D{+}A, A{+}A \xrightarrow{2} C{+}A, C{+}D \xrightarrow{5} 2C{+}D, E{+}D \xrightarrow{5} 2E{+}D\}$. Finally, (B3) does not introduce any change. ◀

▶ **Theorem 23** (Backward reduction induces aggregation). *Let (S, R) be a CRN, $\mathcal{H}$ a BB and μ its choice function. Then, $(S, R)^{(\mathcal{H}, B)}$ is computed in at most $\mathcal{O}\big(|R|\cdot|S|\cdot(\log(|R|)+\log(|S|))\big)$ steps. Crucially, if $\hat{F}$ denotes the vector field induced by $(S, R)^{(\mathcal{H}, B)}$, it holds that $F_X(V) = \hat{F}_X(V)$ for all $X \in \mu(S)$ and $V \in \mathbb{R}_{\geq 0}^S$ that are constant on $\mathcal{H}$.*

5 Partition Refinement Algorithms for CRN Bisimulations

We study a polynomial-time algorithm for the computation of the coarsest bisimulations that refine an arbitrary input partition. We start introducing two auxiliary equivalence relations.

▶ **Definition 24** (Splitter equivalences). Let (S, R) be a CRN and $\mathcal{H}$ a partition over S. Then, we write $X \sim_{\mathcal{H}}^F Y$ if (1) is fulfilled by (X, Y). Similarly, write $X \sim_{\mathcal{H}}^B Y$ if (X, Y) satisfies (2).

Algorithm 1 Template partition refinement algorithm for the construction of the coarsest CRN bisimulations that refine some given initial partition $\mathcal{G}$.

Require: A CRN (S, R), a partition $\mathcal{G}$ of S and $\chi \in \{F, B\}$.

  $\mathcal{H} \longleftarrow \mathcal{G}$
  while true do
    $\mathcal{H}' \longleftarrow S/(\sim_{\mathcal{H}}^{\chi} \cap \sim_{\mathcal{H}})$
    if $\mathcal{H}' = \mathcal{H}$ **then**
      return $\mathcal{H}$
    else
      $\mathcal{H} \longleftarrow \mathcal{H}'$
    end if
  end while

Algorithm 1 iteratively computes the coarsest forward or backward bisimulation (when $\chi = F$ or $\chi = B$, respectively) that refines a given input partition of species of a CRN. Note that, contrary to CRN reduction algorithms, one (parametric) algorithm suffices for both bisimulations. Resorting to the above splitter equivalences, at each iteration the blocks of the current partition $S/\sim_{\mathcal{H}}$ are split in sub-blocks of $\sim_{\mathcal{H}}^{\chi}$-equivalent species $S/(\sim_{\mathcal{H}}^{\chi} \cap \sim_{\mathcal{H}})$. The algorithm terminates when no refinement is performed.

The freedom in choosing the initial partition $\mathcal{G}$ is useful in both bisimulations. For FB it allows to single out species that are the "observables" of the CRN. These are the species for which the modeler is interested in obtaining distinct ODE solutions, information which would otherwise be lost if such species are found in larger equivalence classes. BB is lossless, hence this issue does not arise. However BB requires the same initial conditions for equivalent species. In this case, an appropriate input partition may tell apart species for which it is known that the initial conditions are different.

▶ **Theorem 25** (Correctness). *Given a CRN (S, R) and a partition $\mathcal{G}$ of S, Algorithm 1 calculates the coarsest forward and backward bisimulation that refines $\mathcal{G}$. In both cases, the number of steps needed is polynomial in the number of species and reactions.*

Note that, due to space constraints, we only focussed on the existence of a polynomial-time algorithm, and in the next section we provide numerical evidence of its scalability. The proof of this theorem gives a bound of $\mathcal{O}(|R|^2 \cdot |S|^5)$ on the number of steps. Tighter bounds could be obtained by extending classical partition refinement approaches available for labeled transitions systems [28, 1] to CRNs, which is however the subject of future work.

6   Evaluation

We now evaluate FB and BB. We first study their effectiveness in reducing the ODEs of a number of biochemical models from the literature given in the `.net` format of BioNetGen [5], version 2.2.5-stable. Using selected models we discuss how FB and BB relate with each other, and provide a biological interpretation of the aggregatations. Finally, we compare them against κ's fragmentation. All experiments are replicable using a prototype available at `http://sysma.imtlucca.it/crnreducer/`.

Numerical results. Table 1 lists our case studies: four synthetic benchmarks to obtain combinatorially larger CRNs by varying the number of phosphorylation sites (M1–M4) [30]; a model of pheromone signaling (M5, [32]); two signaling pathways through the Fcε complex (M6–M7, [18, 30]); two models of enzyme activation (M8–M9, [2]); a model of a tumor

	Original model		Forward reduction				Backward reduction			
Id	$|R|$	$|S|$	Red.(s)	$|R|$	$|S|$	Speed-up	Red.(s)	$|R|$	$|S|$	Speed-up
M1	3538944	262146	4.61E+4	990	222	—	7.65E+4	2708	222	—
M2	786432	65538	1.92E+3	720	167	—	3.68E+3	1950	167	—
M3	172032	16386	8.15E+1	504	122	1.16E+3	1.77E+2	1348	122	5.34E+2
M4	48	18	1.00E–3	24	12	1.00E+0	2.00E–3	45	12	1.00E+0
M5	194054	14531	3.72E+1	142165	10855	1.03E+0	1.32E+3	93033	6634	1.03E+0
M6	187468	10734	3.07E+1	57508	3744	1.92E+1	2.71E+2	144473	5575	3.53E+0
M7	32776	2506	1.26E+0	16481	1281	6.23E+0	1.66E+1	32776	2506	x
M8	41233	2562	1.12E+0	33075	1897	1.12E+0	1.89E+1	41233	2562	x
M9	5033	471	1.91E–1	4068	345	1.04E+0	4.35E–1	5033	471	x
M10	5797	796	1.61E–1	4210	503	1.47E+0	7.37E–1	5797	796	x
M11	5832	730	3.89E–1	1296	217	1.32E+1	6.00E–1	2434	217	7.55E+0
M12	487	85	2.00E–3	264	56	1.88E+0	6.00E–3	426	56	1.31E+0
M13	24	18	1.20E–2	24	18	x	7.00E–3	6	3	1.00E+0

Table 1 Forward and backward reductions and corresponding speed-ups in ODE analysis. Speed-up entries "—" indicate that the original model could not be solved; entries "x" indicate that the coarsest bisimulation did not reduce the original model.

suppressor protein (M10, [3]); a model of tyrosine phosphorylation and adaptor protein binding (M11, [12, 13]); a MAPK model (M12, [25]); and an *influence network* (M13, [10]).

Headers $|R|$ and $|S|$ give the number of reactions and species of the CRN (and of its reductions), respectively. The reduction times (*Red.*) account also for the computation of the quotient CRNs. The speed-up is the ratio between the time to solve the ODEs of the original CRN and that of the reduced one including the time to reduce the CRN. Measurements were taken on a 2.6 GHz Intel Core i5 with 4 GB of RAM. The time interval of the ODE solution was taken from the original papers; for M1–M4, where this data was not available, time point 50.0 was used as an estimate of steady state. The initial conditions for the ODEs were also taken from the original papers. The initial partition for FB was chosen to be the trivial one containing the singleton block $\{S\}$ (i.e., no species was singled out). Instead, the initial partition for BB was chosen consistently with the ODE initial conditions; that is, two species may be equivalent only if they have the same initial conditions in the original CRN. This ensured that the backward reduced CRN was a lossless aggregation of the original CRN.

We make three main observations: (i) FB and BB can reduce a significant number of models. In the two largest models of our case studies, M1 and M2, the bisimulations were able to provide a compact aggregated ODE system which could be straightforwardly analyzed, while the solutions of the original models did not terminate due to out-of-memory errors, consistently with [30]. (ii) FB and BB are not comparable in general. For instance, both reduce M5 to 10855 and 6634 species, respectively, while M6 is reduced to 3744 species by FB, and to 5574 by BB. Also, FB was able to reduce M7–M10, while BB did not aggregate. The influence network M13 shows the opposite; in fact, none of the influence networks presented in [10] can be reduced up to FB (here we showed M13, which is the largest one from [10]). (iii) Models M1–M4 and M12 show that the intersection between FB and BB is nonempty.

Biological interpretation. Models M1 and M2 enjoy significant reductions and ODE analysis speed-ups. Here we use them to explain that FB and BB are effective at aggregating species representing symmetric sites in a complex. For this, let us consider M4, chosen for space reasons. A typical equivalence class is for instance $\{E(s!1).S(p1{\sim}P, p2{\sim}U!1), E(s!1).S(p1{\sim}U!1, p2{\sim}P)\}$. According to the syntax of the BioNetGen language, the CRN species are formed from basic *molecules* S and E. Molecule S has two binding sites ($p1$, and $p2$) which

can be either in *phosphorylated* state (P) or not (U); E has one stateless binding site (s) which can bind to those of S to form a complex. The two sites of S have equivalent capabilities in terms of binding with other species or changing state. For instance, the above equivalence class contains two species composed by S and E, with E bound to the unphosphorylated site of S (here the exclamation mark links the binding sites used to form the species). Models M1 and M2 exhibit a fast growth of the number of species due to a larger number of symmetric sites, requiring distinct species to track exactly which site exhibits a particular phosphorylation state. This form of symmetry has also been studied in [9] where the authors propose a fragmentation approach to detect it directly at the κ level. However, an experimental comparison could not be performed because [9] is not yet implemented. Although both bisimulations give the same equivalence classes in these cases, the reduced CRNs have different reactions, since FB provides the dynamics of the sums of equivalent species, while BB considers the distinct dynamics of representative species.

Symmetric sites are not the only property captured by our bisimulations. For instance in both M8 and M9 one of the FB equivalence classes is given by:

$$\big\{ J(k!1).R(x!1, i{\sim}\text{on}, l), \ J(k!1).L(r_1!2, r_2).R(x!1, i{\sim}\text{on}, l!2),$$
$$J(k!1).L(r_1, r_2!2).R(x!1, i{\sim}\text{on}, l!2), \ J(k!1).L(r_1!2, r_2!3).R(x!1, i{\sim}\text{on}, l!3).R(x, i{\sim}\text{on}, l!2)\big\}.$$

A biological interpretation is that a species containing the molecule J behaves in the same way as long as it is bound to a molecule R having binding site i in state "on". This is independent of whether R is further complexed with other molecules via its binding site l; For instance, the first species models that R is only bound to J, while in the second and third species it is also bound to L. Finally, in M5, one of the BB equivalence classes is

$$\big\{ \text{Dig2}(p!1).\text{Ste12}(\text{dig1},\text{dig2},\text{dna}!1,\text{mapk}), \ \text{Fus3}(p!1).\text{Ste12}(\text{dig1},\text{dig2},\text{dna}!1,\text{mapk}),$$
$$\text{Msg5}(p!1).\text{Ste12}(\text{dig1},\text{dig2},\text{dna}!1,\text{mapk}), \ \text{Sst2}(p!1).\text{Ste12}(\text{dig1},\text{dig2},\text{dna}!1,\text{mapk}),$$
$$\text{Ste12}(p!1).\text{Ste12}(\text{dig1},\text{dig2},\text{dna}!1,\text{mapk}), \ \text{Ste2}(p!1).\text{Ste12}(\text{dig1},\text{dig2},\text{dna}!1,\text{mapk})\big\}.$$

It captures that genes Dig2, Fus3, Msg5, Sst2, Ste12, and Ste2, bind to the protein Ste12 with equal rates. This yields equivalent dynamics for these Ste12-gene complexes, and all those formed by them which are equal up to the gene bound to Ste12.

Experimental comparison with κ-based reduction techniques. We now experimentally compare our CRN bisimulations and fragmentation in the case of rule-based biochemical models for which the underlying CRN can be fully enumerated. All models in Table 1 belong to this class; however, none of them was originally available in κ, the only language that supports fragmentation. Thus, we performed a manual translation of a selection of the case studies from the BioNetGen language into κ.[2] We found: (i) Models that can be reduced by our bisimulations but not by fragmentation: For instance, the κ encoding of M12 returned 85 fragments, equal to the size of the CRN, while both FB and BB reduced to 56 species. In addition, fragmentation returned more fragments than species for M6 and M7 (58040 and 10930 fragments, respectively), while the two models are reduced by our bisimulations. (ii) Models that can be reduced by fragmentation but not by our bisimulations: The κ model of early events of the EGF pathway in [6] is reduced from 356 species to 38 fragments [15], while no aggregation is obtained with either FB or BB. (iii) Models that can be reduced by both our bisimulations and fragmentation: The κ encodings of models M1–M4 present different

---

[2] All discussed κ-encodings are shown in Appendix A.4 and are available for download.

reductions than using either bisimulation, specifically 38, 34, 30 and 10 fragments (versus 222, 167, 122, and 12 FB and BB equivalence classes, respectively). It can be shown that, in the latter examples, the reductions are complementary, in the sense that no two bisimilar species are included in the same fragment. While our bisimulations captured symmetric sites, fragments explain that the sites of S are *independent*, i.e., the state of a site does not affect the dynamics of the other. For instance, one of the fragments for model M4 is

$$\{S(p1{\sim}P, p2{\sim}P), S(p1{\sim}P, p2{\sim}U), E(s!1).S(p1{\sim}P, p2{\sim}U!1), F(s!1).S(p1{\sim}P, p2{\sim}P!1)\}$$

which essentially collects all species where the $p1$ site of molecule S is phosphorylated.

7   Conclusion

Forward and backward bisimulations are equivalence relations over the species of a chemical reaction network inducing a partition of the underlying mass-action system of ordinary differential equations. An experimental evaluation has demonstrated their usefulness by showing their complementarity as well as significant model reductions in a number of biochemical models available in the literature. This has been supported by a prototype, which currently allows a ready-to-use tool-chain with BioNetGen, a state-of-the-art tool.

Ongoing work is studying stochastic counterparts of both forward and backward bisimulations, to obtain model reductions when the semantics of chemical reaction networks based on continuous-time Markov chains is considered. Also, we plan to investigate the applicability of our bisimulations in other model repositories, e.g., those using the well-known SBML interchange format (`http://sbml.org`).

——— **References** ———

1   C. Baier, B. Engelen, and M. E. Majster-Cederbaum. Deciding bisimilarity and similarity for probabilistic processes. *J. Comput. Syst. Sci.*, 60(1):187–231, 2000.

2   D. Barua, J. R. Faeder, and J. M. Haugh. A bipolar clamp mechanism for activation of jak-family protein tyrosine kinases. *PLoS Computational Biology*, 5(4), 2009.

3   D. Barua and W. S. Hlavacek. Modeling the effect of apc truncation on destruction complex function in colorectal cancer cells. *PLoS Comput Biol*, 9(9):e1003217, 09 2013.

4   M. Bernardo. A survey of Markovian behavioral equivalences. In *Formal Methods for Perf. Eval.*, volume 4486 of *LNCS*, pages 180–219. Springer Berlin Heidelberg, 2007.

5   M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004.

6   M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. *Biosystems*, 83:136–151, 2006.

7   P. Buchholz. Exact and Ordinary Lumpability in Finite Markov Chains. *Journal of Applied Probability*, 31(1):59–75, 1994.

8   P. Buchholz. Markovian Process Algebra: Composition and Equivalence. In *Proc. 2nd Workshop on Process Algebra and Performance Modelling*, Erlangen, Germany, 1994.

9   F. Camporesi and J. Feret. Formal reduction for rule-based models. *Electronic Notes in Theoretical Computer Science*, 276:29–59, 2011. MFPS XXVII.

10 L. Cardelli. Morphisms of reaction networks that couple structure to function. *BMC Systems Biology*, 8(1):84, 2014.

11 L. Cardelli and A. Csikász-Nagy. The cell cycle switch computes approximate majority. *Sci. Rep.*, 2, 2012.

12 J. Colvin, M. I. Monine, J. R. Faeder, W. S. Hlavacek, D. D. Von Hoff, and R. G. Posner. Simulation of large-scale rule-based models. *Bioinformatics*, 25(7):910–917, 2009.

13 J. Colvin, M. I. Monine, R. N. Gutenkunst, W. S. Hlavacek, D. D. Von Hoff, and R. G. Posner. Rulemonkey: software for stochastic simulation of rule-based models. *BMC Bioinformatics*, 11:404, 2010.

14 H. Conzelmann, J. Saez-Rodriguez, T. Sauter, B. Kholodenko, and E. Gilles. A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. *BMC Bioinformatics*, 7(1):34, 2006.

15 V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Abstracting the differential semantics of rule-based models: Exact and automated model reduction. In *LICS*, pages 362–381, 2010.

16 V. Danos and C. Laneve. Formal molecular biology. *TCS*, 325(1):69–110, 2004.

17 R. De Nicola, U. Montanari, and F. Vaandrager. Back and forth bisimulations. In *CONCUR*, volume 458 of *LNCS*, pages 152–165. Springer, 1990.

18 J. R. Faeder, W. S. Hlavacek, I. Reischl, M. L. Blinov, H. Metzger, A. Redondo, C. Wofsy, and B. Goldstein. Investigation of early events in FcεRI-mediated signaling using a detailed mathematical model. *The Journal of Immunology*, 170(7):3769–3781, 2003.

19 J. Feret, T. Henzinger, H. Koeppl, and T. Petrov. Lumpability abstractions of rule-based systems. *TCS*, 431:137–164, 2012.

20 J. Fisher and T.A. Henzinger. Executable cell biology. *Nature Biotechnology*, 25(11):1239–1249, 2007. See also correspondence in Nature Biotechnology 26(7):737-8;738-9, 2008.

21 D. Gillespie. The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

22 J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. *TCS*, 391(3):239–257, 2008.

23 H. Hermanns and M. Rettelbach. Syntax, semantics, equivalences, and axioms for MTIPP. In *Proceedings of Process Algebra and Probabilistic Methods*, pages 71–87, Erlangen, 1994.

24 J. Hillston. *A Compositional Approach to Performance Modelling*. CUP, 1996.

25 P. Kocieniewski, J. R. Faeder, and T. Lipniacki. The interplay of double phosphorylation and scaffolding in MAPK pathways. *Journal of Theoretical Biology*, 295:116–124, 2012.

26 K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

27 M. S. Okino and M. L. Mavrovouniotis. Simplification of mathematical models of chemical reaction systems. *Chemical Reviews*, 2(98):391–408, 1998.

28 R. Paige and R. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

29 S. W. Shin, C. Thachuk, and E. Winfree. Verifying chemical reaction network implementations: A pathway decomposition approach. In *VEMPD*, Vienna Summer of Logic, 2014.

30 M. W. Sneddon, J. R. Faeder, and T. Emonet. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nature Methods*, 8(2):177–183, 2011.

31 J. Sproston and S. Donatelli. Backward Bisimulation in Markov Chain Model Checking. *IEEE Trans. Software Eng.*, 32(8):531–546, 2006.

32 R. Suderman and E. J. Deeds. Machines vs. ensembles: Effective MAPK signaling through heterogeneous sets of protein complexes. *PLoS Comput Biol*, 9(10):e1003278, 10 2013.

33 J. Toth, G. Li, H. Rabitz, and A. S. Tomlin. The effect of lumping and expanding on kinetic differential equations. *SIAM Journal on Applied Mathematics*, 57(6):1531–1556, 1997.

34 M. Tschaikowski and M. Tribastone. Exact fluid lumpability for Markovian process algebra. In *CONCUR*, LNCS, pages 380–394, 2012.

35 M. Tschaikowski and M. Tribastone. A unified framework for differential aggregations in Markovian process algebra. *JLAMP*, 84(2):238–258, 2015.

APPENDIX

A.1 Forward CRN Bisimulation

The following auxiliary lemma will be needed in the proofs of Propositions 10 and 16.

▶ **Lemma 26.** *Let I be an index set and let $\mathcal{R}_i$ denote equivalence relations on S such that $S/\mathcal{R}_i$ is a refinement of some partition $\mathcal{G}$ of S. Then, $\mathcal{R} = (\bigcup_{i \in I} \mathcal{R}_i)^*$ is such that $S/\mathcal{R}$ is a refinement of $\mathcal{G}$.*

Proof. We first note that $\mathcal{R}$ is an equivalence relation over S, as it is the transitive closure of the union of equivalence relations over S. For $i \in I$, set $\mathcal{H}_i = S/\mathcal{R}_i$ and $\mathcal{H} = S/\mathcal{R}$. Then, for any $(y_0, y_1) \in \mathcal{R}$, there exist $x_0, \dots, x_k \in S$ such that $x_0 R_{i_0} x_1 R_{i_1} \dots R_{i_{k-1}} x_k$ with $y_0 = x_0$, $y_1 = x_k$ and $i_j \in I$ for all $0 \le j \le k-1$. Moreover, $x_0 \in G$ for some (unique) $G \in \mathcal{G}$. We show that $x_0, \dots, x_k \in G$ by induction. Since the base case $j = 0$ is trivial, let us consider the induction step $j - 1 \to j$. Then, $x_{j-1} \mathcal{R}_{i_{j-1}} x_j$ implies the existence of some $H \in \mathcal{H}_{i_{j-1}}$ such that $x_{j-1}, x_j \in H$. Let $G_{j-1} \in \mathcal{G}$ be such that $H \subseteq G_{j-1}$. Since $x_{j-1} \in G$ by induction hypothesis and $x_{j-1} \in G_{j-1}$, it holds that $G \cap G_{j-1} \ne \emptyset$. Since $\mathcal{G}$ is a partition, this implies that $G = G_{j-1}$, yielding in turn $x_j \in G$. ◀

Proof of Proposition 10. We first note that $\mathcal{R}$ is an equivalence relation over S, as it is the transitive closure of the union of equivalence relations over S. For $i \in I$, set $\mathcal{H}_i = S/\mathcal{R}_i$ and $\mathcal{H} = S/\mathcal{R}$. Note that, for all $i \in I$, any $H \in \mathcal{H}$ is a union of blocks of $\mathcal{H}_i$. For any $(y_0, y_1) \in \mathcal{R}$, there exist $x_0, \dots, x_k \in S$ such that $x_0 R_{i_0} x_1 R_{i_1} \dots R_{i_{k-1}} x_k$ with $y_0 = x_0$, $y_1 = x_k$ and $i_j \in I$ for all $0 \le j \le k-1$. Noting that $x_j R_{i_j} x_{j+1}$ for all $0 \le j \le k-1$, we infer that $\mathbf{crr}[x_j, \rho] = \mathbf{crr}[x_{j+1}, \rho]$ and $\mathbf{pr}[x_j, \rho, H] = \mathbf{pr}[x_{j+1}, \rho, H]$ for all $\rho \in \mathcal{MS}(S)$ and $H \in \mathcal{H}$. Thus, we infer that $\mathbf{crr}[x_0, \rho] = \mathbf{crr}[x_1, \rho] = \dots = \mathbf{crr}[x_{k-1}, \rho] = \mathbf{crr}[x_k, \rho]$ and $\mathbf{pr}[x_0, \rho, H] = \mathbf{pr}[x_1, \rho, H] = \dots = \mathbf{pr}[x_{k-1}, \rho, H] = \mathbf{pr}[x_k, \rho, H]$. The remainder of the claim follows then from Lemma 26. ◀

We now provide a sort of commutative property for $\mathbf{crr}$ and $\mathbf{pr}$ for the case of singleton ρ, used to prove Theorem 11.

▶ **Proposition** 27. Let (S, R) be a CRN and $X, Y \in S$. Then it holds

$$\mathbf{crr}(X, Y) = \mathbf{crr}(Y, X) \tag{3}$$
$$\mathbf{pr}(X, Y, Z) = \mathbf{pr}(Y, X, Z) \tag{4}$$

Proof. The statement follows from Definition 7, since (3) and (4) can be rewritten, resp., as:

$$\{\!|X, Y|\!\}(X) \cdot \sum_{X+Y \xrightarrow{\alpha} \pi \in R} \alpha = \{\!|X, Y|\!\}(Y) \cdot \sum_{X+Y \xrightarrow{\alpha} \pi \in R} \alpha$$

$$\{\!|X, Y|\!\}(X) \cdot \sum_{X+Y \xrightarrow{\alpha} \pi \in R} \alpha \cdot \pi(Z) = \{\!|X, Y|\!\}(Y) \cdot \sum_{X+Y \xrightarrow{\alpha} \pi \in R} \alpha \cdot \pi(Z)$$

◀

Proof of Theorem 11

For (S, R) a CRN, $H \subseteq S$, and V a concentration function for (S, R), we use V_H for $\sum_{X \in H} V_X$, and X^H or Y^H to denote any element of H. We first separate the influence exerted by each reaction to the concentration of a species in a negative one (the *depletion* rate), and in a positive one (the *accretion rate*).

▶ Remark. Let (S, R) be a CRN, V a concentration function for (S, R), and $X \in S$. We denote the *depletion* and *accretion* rates of X due to a reaction of R as, respectively,

$$Depl(\rho \xrightarrow{\alpha} \pi, X, V) = \rho(X) \cdot \alpha \cdot \prod_{Y \in S} V_Y^{\rho(Y)}$$

$$Accr(\rho \xrightarrow{\alpha} \pi, X, V) = \pi(X) \cdot \alpha \cdot \prod_{Y \in S} V_Y^{\rho(Y)}$$

We can then reformulate the definition of vector field of (S, R) given in Section 2.1 as:

$$F_X(V) = \sum_{\rho \xrightarrow{\alpha} \pi \in R} \Big(Accr(\rho \xrightarrow{\alpha} \pi, X, V) - Depl(\rho \xrightarrow{\alpha} \pi, X, V) \Big) \qquad\qquad ◀$$

We now state that the aggregated accretion and depletion rates of an equivalence class of an FB can be written in terms of the aggregate concentrations.

▶ **Proposition** 28 (Aggregated depletion and accretion rate). Let (S, R) be a CRN, V a concentration function for (S, R), and $\mathcal{H}$ an FB. Then, for any $H \in \mathcal{H}$

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Depl(\rho \xrightarrow{\alpha} \pi, V, X) \quad \text{and} \quad \sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Accr(\rho \xrightarrow{\alpha} \pi, V, X)$$

can be written in terms of the aggregated concentrations $V_{H_1} \ldots V_{H_n}$.

Proof. We start addressing the depletion rate case. We have

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Depl(\rho \xrightarrow{\alpha} \pi, V, X) = \sum_{X \in H} \sum_{\rho \in \mathcal{MS}(S)} \sum_{\rho \xrightarrow{\alpha} \pi \in R} \rho(X) \cdot \alpha \cdot \prod_{Y \in S} V_Y^{\rho(Y)}$$

$$= \sum_{X \in H} \sum_{\rho \in \mathcal{MS}(S)} \prod_{Y \in S} V_Y^{\rho(Y)} \sum_{\rho \xrightarrow{\alpha} \pi \in R} \rho(X) \cdot \alpha \qquad\qquad (5)$$

Given that we have unary or binary reactions only, we can rewrite (5) as:

$$\sum_{X \in H} \sum_{Y \in S} V_Y \sum_{Y \xrightarrow{\alpha} \pi \in R} \{\!|Y|\!\}(X) \cdot \alpha + \qquad\qquad (6)$$

$$\sum_{X \in H} \sum_{Y + Y' \in \mathcal{MS}(S)} V_Y \cdot V_Y' \sum_{Y + Y' \xrightarrow{\alpha} \pi \in R} \{\!|Y, Y'|\!\}(X) \cdot \alpha \qquad\qquad (7)$$

Now, (6) can be easily rewritten in terms of the aggregated variables only:

$$\sum_{X \in H} \sum_{Y \in S} V_Y \sum_{Y \xrightarrow{\alpha} \pi \in R} \{\!|Y|\!\}(X) \cdot \alpha = \sum_{X \in H} V_X \sum_{X \xrightarrow{\alpha} \pi \in R} \alpha = \sum_{X \in H} V_X \cdot \mathbf{crr}[X, \emptyset],$$

from which by the condition on $\mathbf{crr}$ of Definition 8 we obtain $\mathbf{crr}[X^H, \emptyset] \cdot V_H$.

Instead, as regards (7) we have:

$$\sum_{X \in H} \sum_{Y+Y' \in \mathcal{MS}(S)} V_Y \cdot V'_Y \sum_{Y+Y' \xrightarrow{\alpha} \pi \in R} \{\!| Y, Y' |\!\}(X) \cdot \alpha =$$

$$\sum_{X \in H} \sum_{X+Y \in \mathcal{MS}(S)} V_X \cdot V_Y \sum_{X+Y \xrightarrow{\alpha} \pi \in R} \{\!| X, Y |\!\}(X) \cdot \alpha =$$

$$\sum_{X \in H} V_X \sum_{X+Y \in \mathcal{MS}(S)} V_Y \cdot \mathbf{crr}[X, Y] = \text{(by the condition on } \mathbf{crr} \text{ of Definition 8)}$$

$$\sum_{X \in H} V_X \sum_{X+Y \in \mathcal{MS}(S)} V_Y \cdot \mathbf{crr}[X^H, Y] =$$

$$\sum_{X \in H} V_X \sum_{\tilde{H} \in \mathcal{H}} \sum_{Y \in \tilde{H}} V_Y \cdot \mathbf{crr}[X^H, Y] = \text{(by Proposition 27)}$$

$$\sum_{X \in H} V_X \sum_{\tilde{H} \in \mathcal{H}} \sum_{Y \in \tilde{H}} V_Y \cdot \mathbf{crr}[Y, X^H] = \text{(by the condition on } \mathbf{crr} \text{ of Definition 8)}$$

$$\sum_{X \in H} V_X \sum_{\tilde{H} \in \mathcal{H}} \mathbf{crr}[Y^{\tilde{H}}, X^H] \sum_{Y \in \tilde{H}} V_Y =$$

$$\sum_{X \in H} V_X \sum_{\tilde{H} \in \mathcal{H}} \mathbf{crr}[Y^{\tilde{H}}, X^H] \cdot V_{\tilde{H}} =$$

$$\sum_{\tilde{H} \in \mathcal{H}} V_{\tilde{H}} \cdot \mathbf{crr}[Y^{\tilde{H}}, X^H] \sum_{X \in H} V_X =$$

$$\sum_{\tilde{H} \in \mathcal{H}} V_{\tilde{H}} \cdot \mathbf{crr}[Y^{\tilde{H}}, X^H] \cdot V_H = V_H \sum_{\tilde{H} \in \mathcal{H}} V_{\tilde{H}} \cdot \mathbf{crr}[X^H, Y^{\tilde{H}}]$$

closing the case.

We now address the accretion rate case. We have

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Accr(\rho \xrightarrow{\alpha} \pi, V, X) = \sum_{X \in H} \sum_{\rho \in \mathcal{MS}(S)} \sum_{\rho \xrightarrow{\alpha} \pi \in R} \pi(X) \cdot \alpha \cdot \prod_{Y \in S} V_Y^{\rho(Y)}$$

$$= \sum_{X \in H} \sum_{\rho \in \mathcal{MS}(S)} \prod_{Y \in S} V_Y^{\rho(Y)} \sum_{\rho \xrightarrow{\alpha} \pi \in R} \pi(X) \cdot \alpha \qquad (8)$$

Given that we have unary or binary reactions only, we can rewrite (8) as:

$$\sum_{X \in H} \sum_{Y \in S} V_Y \sum_{Y \xrightarrow{\alpha} \pi \in R} \pi(X) \cdot \alpha + \qquad (9)$$

$$+ \sum_{X \in H} \sum_{Y+Y' \in \mathcal{MS}(S)} V_Y \cdot V_{Y'} \sum_{Y+Y' \xrightarrow{\alpha} \pi \in R} \pi(X) \cdot \alpha \qquad (10)$$

Now, (9) can be easily rewritten in terms of the aggregated variables only:

$$\sum_{Y \in S} V_Y \sum_{X \in H} \sum_{Y \xrightarrow{\alpha} \pi \in R} \pi(X) \cdot \alpha = \sum_{Y \in S} V_Y \cdot \mathbf{pr}[Y, \emptyset, H] = \sum_{\tilde{H} \in \mathcal{H}} \sum_{Y \in \tilde{H}} V_Y \cdot \mathbf{pr}[Y, \emptyset, H],$$

from which, by the condition on $\mathbf{pr}$ of Definition 8, we obtain $\sum_{\tilde{H} \in \mathcal{H}} \mathbf{pr}[Y^{\tilde{H}}, \emptyset, H] \cdot V_{\tilde{H}}$.

Instead, as regards (10) we have:

$$\sum_{Y+Y'\in\mathcal{MS}(S)} V_Y \cdot V_{Y'} \sum_{X\in H} \sum_{Y+Y'\xrightarrow{\alpha}\pi\in R} \pi(X)\cdot\alpha =$$

$$\sum_{Y+Y\in\mathcal{MS}(S)} V_Y^2 \sum_{X\in H} \sum_{Y+Y\xrightarrow{\alpha}\pi\in R} \pi(X)\cdot\alpha +$$

$$\sum_{Y+Y'\in\mathcal{MS}(S)\ s.t.\ Y\neq Y'} V_Y \cdot V_{Y'} \sum_{X\in H} \sum_{Y+Y'\xrightarrow{\alpha}\pi\in R} \pi(X)\cdot\alpha = \quad \text{(see below)}$$

$$\frac{1}{2}\sum_{Y\in S} V_Y^2 \sum_{X\in H} 2 \sum_{Y+Y\xrightarrow{\alpha}\pi\in R} \pi(X)\cdot\alpha + \tag{11}$$

$$\frac{1}{2}\sum_{Y\in S} V_Y \sum_{Y'\in S\ s.t.\ Y\neq Y'} V_{Y'} \sum_{X\in H} \sum_{Y+Y'\xrightarrow{\alpha}\pi\in R} \pi(X)\cdot\alpha = \tag{12}$$

$$\frac{1}{2}\sum_{Y\in S} V_Y^2 \cdot \mathbf{pr}[Y,Y,H] + \frac{1}{2}\sum_{Y\in S} V_Y \sum_{Y'\in S\ s.t.\ Y\neq Y'} V_{Y'}\cdot\mathbf{pr}[Y,Y',H] =$$

$$\frac{1}{2}\sum_{Y\in S} V_Y \sum_{Y'\in S} V_{Y'}\cdot\mathbf{pr}[Y,Y',H] =$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}}\sum_{Y\in\tilde{H}} V_Y \sum_{Y'\in S} V_{Y'}\cdot\mathbf{pr}[Y,Y',H] = \quad \text{(by cond on } \mathbf{pr} \text{ of Definition 8)}$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{Y'\in S} V_{Y'}\cdot\mathbf{pr}[Y^{\tilde{H}},Y',H] =$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H}}\sum_{Y'\in\widehat{H}} V_{Y'}\cdot\mathbf{pr}[Y^{\tilde{H}},Y',H] = \quad \text{(by Proposition 27)}$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H}}\sum_{Y'\in\widehat{H}} V_{Y'}\cdot\mathbf{pr}[Y',Y^{\tilde{H}},H] = \quad \text{(by cond on } \mathbf{pr} \text{ of Definition 8)}$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H}} \mathbf{pr}[Y^{\widehat{H}},Y^{\tilde{H}},H] \sum_{Y'\in\widehat{H}} V_{Y'} = \frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H}} V_{\widehat{H}}\cdot\mathbf{pr}[Y^{\widehat{H}},Y^{\tilde{H}},H]$$

Where (11) is multiplied by 2/2, while (12) is divided by two because each multiset is considered twice (e.g., $Y + Y'$ and $Y' + Y$). ◀

Finally we can provide the proof of Theorem 11.

Proof of Theorem 11. The theorem can be proved by showing that, for each $H \in \mathcal{H}$, the sum of the ODEs of the species in H can be expressed in terms of the aggregated concentrations of the blocks of $\mathcal{H}$ only. Such sum is:

$$\sum_{X\in H} \sum_{\rho\xrightarrow{\alpha}\pi\in R} Accr(\rho\xrightarrow{\alpha}\pi,V,X) - \sum_{X\in H} \sum_{\rho\xrightarrow{\alpha}\pi\in R} Depl(\rho\xrightarrow{\alpha}\pi,V,X)$$

Thus, the claim follows from Proposition 28. ◀

Proof of Theorem 20

Given a CRN (S,R) and an FB $\mathcal{H}$, we hereby provide the technical results relating the $\mathcal{H}$-lumped ODEs of (S,R) and the ODEs of its $(\mathcal{H},F)$-reduction. In the following we use

$\mathcal{X}^H$ to denote the canonical representative of the species in a block $H \in \mathcal{H}$. In addition, for $X \in S$ and $\pi \in \mathcal{MS}(S)$, we may use μ_X for $\mu(X)$ and μ_π for $\mu(\pi)$. Finally, given a concentration function V for (S,R), we use $V^{(\mathcal{H},F)}$ for the corresponding concentration function for $(S,R)^{(\mathcal{H},F)}$, having a component $V^{(\mathcal{H},F)}_{\mathcal{X}^H} = V_H$ per block $H \in \mathcal{H}$.

We start providing a proposition used in the proof of Theorem 20.

▶ **Proposition** 29. Let (S,R) be a CRN, $\mathcal{R}$ an FB, $\mathcal{H} = S/\mathcal{R}$, and μ its choice function. Let V be a concentration function for (S,R). Then, for any $H \in \mathcal{H}$ we have

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Accr(\rho \xrightarrow{\alpha} \pi, X, V) = \sum_{\substack{\rho \xrightarrow{\alpha} \pi \in R \\ \rho = \mu(\rho)}} Accr(\rho \xrightarrow{\alpha} \mu(\pi), \mathcal{X}^H, V^{(\mathcal{H},F)}) \tag{13}$$

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Depl(\rho \xrightarrow{\alpha} \pi, X, V) = \sum_{\substack{\rho \xrightarrow{\alpha} \pi \in R \\ \rho = \mu(\rho)}} Depl(\rho \xrightarrow{\alpha} \mu(\pi), \mathcal{X}^H, V^{(\mathcal{H},F)}) \tag{14}$$

Proof. We only address the *Accr* case, as the *Depl* one is similar (actually, the latter is simpler). By the proof of Proposition 28 we obtain that we can rewrite the left-hand side of Equation (13) as

$$\sum_{\tilde{H} \in \mathcal{H}} \mathbf{pr}[Y^{\tilde{H}}, \emptyset, H] \cdot V_{\tilde{H}} + \tag{15}$$

$$\frac{1}{2} \sum_{\tilde{H} \in \mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H} \in \mathcal{H}} V_{\widehat{H}} \cdot \mathbf{pr}[Y^{\widehat{H}}, Y^{\tilde{H}}, H] \tag{16}$$

Instead, we can rewrite the right-hand side of Equation (13) as

$$\sum_{\substack{Y \xrightarrow{\alpha} \pi \in R \\ Y = \mu(Y)}} \mu_\pi(\mathcal{X}^H) \cdot \alpha \cdot V_Y^{(\mathcal{H},F)} + \tag{17}$$

$$\sum_{\substack{Y_1 + Y_2 \xrightarrow{\alpha} \pi \in R \\ Y_1 = \mu(Y_1) \wedge Y_2 = \mu(Y_2)}} \mu_\pi(\mathcal{X}^H) \cdot \alpha \cdot V_{Y_1}^{(\mathcal{H},F)} \cdot V_{Y_2}^{(\mathcal{H},F)} \tag{18}$$

We close the proof showing that Equation (15) is equal to Equation (17) and that Equation (16) is equal to Equation (18).

We focus on the first equality, which easily follows noting that Equation (17) can be rewritten as

$$\sum_{\tilde{H} \in \mathcal{H}} V_{\mathcal{X}^{\tilde{H}}}^{(\mathcal{H},F)} \sum_{\substack{Y \xrightarrow{\alpha} \pi \in R \\ Y = \mathcal{X}^{\tilde{H}}}} \mu_\pi(\mathcal{X}^H) \cdot \alpha = \sum_{\tilde{H} \in \mathcal{H}} V_{\tilde{H}} \sum_{Z \in H} \sum_{\substack{Y \xrightarrow{\alpha} \pi \in R \\ Y = \mathcal{X}^{\tilde{H}}}} \pi(Z) \cdot \alpha = \sum_{\tilde{H} \in \mathcal{H}} V_{\tilde{H}} \cdot \mathbf{pr}[\mathcal{X}^{\tilde{H}}, \emptyset, H]$$

We now consider the case "Equation (16) equals Equation (18)". We can rewrite Equation (18) as follows:

$$\sum_{\substack{Y_1 + Y_1 \xrightarrow{\alpha} \pi \in R \\ Y_1 = \mu(Y_1)}} \mu_\pi(\mathcal{X}^H) \cdot \alpha \cdot V_{Y_1}^{(\mathcal{H},F)} \cdot V_{Y_1}^{(\mathcal{H},F)} + \tag{19}$$

$$\sum_{\substack{Y_1 + Y_2 \xrightarrow{\alpha} \pi \in R, Y_1 \neq Y_2 \\ Y_1 = \mu(Y_1) \wedge Y_2 = \mu(Y_2)}} \mu_\pi(\mathcal{X}^H) \cdot \alpha \cdot V_{Y_1}^{(\mathcal{H},F)} \cdot V_{Y_2}^{(\mathcal{H},F)} \tag{20}$$

Now, Equation (19) can be rewritten as

$$\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}}\cdot V_{\tilde{H}} \sum_{\substack{Y_1+Y_1\xrightarrow{\alpha}\pi\in R\\Y_1=\mathcal{X}^{\tilde{H}}}} \mu_\pi(\mathcal{X}^H)\cdot\alpha =$$

$$\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}}\cdot V_{\tilde{H}} \sum_{Z\in H}\sum_{\substack{Y_1+Y_1\xrightarrow{\alpha}\pi\in R\\Y_1=\mathcal{X}^{\tilde{H}}}} \pi(Z)\cdot\alpha = \textit{(multiplying by }\frac{2}{2}\textit{)}$$

$$\frac{1}{2}\sum_{\tilde{Y}\in\mathcal{H}} V_{\tilde{H}}\cdot V_{\tilde{H}} \sum_{Z\in H} 2 \sum_{\substack{Y_1+Y_1\xrightarrow{\alpha}\pi\in R\\Y_1=\mathcal{X}^{\tilde{H}}}} \pi(Z)\cdot\alpha =$$

$$\frac{1}{2}\sum_{\tilde{Y}\in\mathcal{H}} V_{\tilde{H}}\cdot V_{\tilde{H}}\cdot\mathbf{pr}[\mathcal{X}^{\tilde{H}},\mathcal{X}^{\tilde{H}},H] \tag{21}$$

Instead, Equation (20) can be rewritten as, where we divide Equation (22) by 2 because each multiset is considered twice (e.g., Y_1+Y_2 and Y_2+Y_1):

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H},\tilde{H}\neq\widehat{H}} V_{\widehat{H}} \sum_{\substack{Y_1+Y_2\xrightarrow{\alpha}\pi\in R\\Y_1=\mathcal{X}^{\tilde{H}}\wedge Y_2=\mathcal{X}^{\widehat{H}}}} \mu_\pi(\mathcal{X}^H)\cdot\alpha = \tag{22}$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H},\tilde{H}\neq\widehat{H}} V_{\widehat{H}} \sum_{Z\in H}\sum_{\substack{Y_1+Y_2\xrightarrow{\alpha}\pi\in R\\Y_1=\mathcal{X}^{\tilde{H}}\wedge Y_2=\mathcal{X}^{\widehat{H}}}} \pi(Z)\cdot\alpha =$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H},\tilde{H}\neq\widehat{H}} V_{\widehat{H}}\cdot\mathbf{pr}[\mathcal{X}^{\tilde{H}},\mathcal{X}^{\widehat{H}},H] \tag{23}$$

By summing Equation (21) and Equation (23), we finally rewrite Equation (18) as

$$\frac{1}{2}\sum_{\tilde{Y}\in\mathcal{H}} V_{\tilde{H}}\cdot V_{\tilde{H}}\cdot\mathbf{pr}[\mathcal{X}^{\tilde{H}},\mathcal{X}^{\tilde{H}},H] + \frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H},\tilde{H}\neq\widehat{H}} V_{\widehat{H}}\cdot\mathbf{pr}[\mathcal{X}^{\tilde{H}},\mathcal{X}^{\widehat{H}},H] =$$

$$\frac{1}{2}\sum_{\tilde{H}\in\mathcal{H}} V_{\tilde{H}} \sum_{\widehat{H}\in\mathcal{H}} V_{\widehat{H}}\cdot\mathbf{pr}[\mathcal{X}^{\tilde{H}},\mathcal{X}^{\widehat{H}},H]$$

This closes the case, and thus the proof is complete. ◀

We now provide the proof of Theorem 20.

Theorem 20. We first address the correctness of the reduction. For any $H\in\mathcal{H}$ we have

$$\sum_{X\in H} F_X(V) = \sum_{X\in H}\sum_{\rho\xrightarrow{\alpha}\pi\in R} Accr(\rho\xrightarrow{\alpha}\pi,X,V) - \sum_{X\in H}\sum_{\rho\xrightarrow{\alpha}\pi\in R} Depl(\rho\xrightarrow{\alpha}\pi,X,V)$$

As regards $\hat{F}$, by Definition 18, for any $H\in\mathcal{H}$ we obtain

$$\hat{F}_{\mathcal{X}^H}(V^{(\mathcal{H},F)}) = \sum_{\substack{\rho\xrightarrow{\alpha}\pi\in R\\\rho=\mu(\rho)}} Accr(\rho\xrightarrow{\alpha}\mu(\pi),\mathcal{X}^H,V^{(\mathcal{H},F)}) -$$

$$\sum_{\substack{\rho\xrightarrow{\alpha}\pi\in R\\\rho=\mu(\rho)}} Depl(\rho\xrightarrow{\alpha}\mu(\pi),\mathcal{X}^H,V^{(\mathcal{H},F)})$$

We close the proof by showing that

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Accr(\rho \xrightarrow{\alpha} \pi, X, V) = \sum_{\substack{\rho \xrightarrow{\alpha} \pi \in R \\ \rho = \mu(\rho)}} Accr(\rho \xrightarrow{\alpha} \mu(\pi), \mathcal{X}^H, V^{(\mathcal{H}, F)}) \tag{24}$$

$$\sum_{X \in H} \sum_{\rho \xrightarrow{\alpha} \pi \in R} Depl(\rho \xrightarrow{\alpha} \pi, X, V) = \sum_{\substack{\rho \xrightarrow{\alpha} \pi \in R \\ \rho = \mu(\rho)}} Depl(\rho \xrightarrow{\alpha} \mu(\pi), \mathcal{X}^H, V^{(\mathcal{H}, F)}) \tag{25}$$

Both Equations (24) and (25) follow from Proposition 29.

We now address the complexity of the reduction, showing that $(S, R)^{(\mathcal{H}, F)}$ can be performed in $\mathcal{O}\big(|R| \cdot |S| \cdot (\log(|S|) + \log(|R|))\big)$ time.

Steps (F1) and (F2) of Definition 18 require to iterate (once) the reactions, $(\mathcal{O}(|R|))$. In particular, for each reaction we have to in turn iterate its reagents and products to perform (F1) and (F2), respectively. This requires $\mathcal{O}(|S|)$ time. Finally, in order to efficiently perform (F3) we assume that the reagents and products are stored as an ordered list, and thus we have to sort the obtained canonized products $(\mathcal{O}(|S| \cdot \log(|S|)))$. To sum up, steps (F1) and (F2) require $\mathcal{O}\big(|R| \cdot |S| \cdot \log(|S|)\big)$ time.

Instead, step (F3) can be computed by first sorting the reactions obtained from (F1) and (F2), requiring $\mathcal{O}(|R| \cdot \log(|R|) \cdot |S|)$, where the $|S|$ factor comes from the fact that in order to compare two reactions it is necessary to scan (once) their reagents and products. Then, (F3) is completed by iterating (once) the reactions to actually collapse them $(\mathcal{O}(|R| \cdot |S|))$. ◀

A.2 Backward CRN Bisimulation

Proof of Theorem 6. We first prove the if direction. Let $\mu : S \to S$ be some choice function of $\mathcal{H}$, set $\hat{S} := \mu(S)$ and define $G_{\hat{X}}(\hat{V}) := F_{\hat{X}}(\hat{V} \circ \mu)$ for any $\hat{V} \in \mathbb{R}^{\hat{S}}$ and $\hat{X} \in \hat{S}$. Further, let $\hat{V}$ denote the unique ODE solution of $\frac{d}{dt}\hat{V}(t) = G\big(\hat{V}(t)\big)$ subject to some given initial condition $\hat{V}(0)$. Then, for all $X \in S$, it holds that

$$\frac{d}{dt}(\hat{V}(t) \circ \mu)_X = \left(\frac{d}{dt}\hat{V}(t)\right)_{\mu(X)} = G_{\mu(X)}\big(\hat{V}(t)\big) = F_{\mu(X)}(\hat{V}(t) \circ \mu) = F_X(\hat{V}(t) \circ \mu)$$

Thus, $t \mapsto \hat{V}(t) \circ \mu$ is the unique solution of the ODE system $\frac{d}{dt}V(t) = F(V(t))$ subject to $\hat{V}(0) \circ \mu$. Since $t \mapsto \hat{V}(t) \circ \mu$ is constant on $\mathcal{H}$, the proof of the if direction is complete. We now turn to the proof of the only-if direction. For this, let us assume that F is such that, for any $V(0) \in \mathbb{R}^S_{\geq 0}$ that is constant on $\mathcal{H}$, the underlying solution of $\dot{V} = F(V)$ is constant on $\mathcal{H}$ as well. Fix arbitrary $H \in \mathcal{H}$, $X, Y \in H$ and $V(0) \in \mathbb{R}^S_{\geq 0}$ with $V(0)$ constant on $\mathcal{H}$. Since the solution of $\dot{V} = F(V)$ is smooth, we can apply Taylor's theorem to infer that there exist $\delta > 0$ and functions r_X, r_Y such that $\lim_{h \searrow 0} \frac{1}{h} r_X(h) = \lim_{h \searrow 0} \frac{1}{h} r_Y(h) = 0$ and

$$V_X(h) = V_X(0) + F_X(V(0)) \cdot h + r_X(h) \qquad V_Y(h) = V_Y(0) + F_Y(V(0)) \cdot h + r_Y(h)$$

for all $0 \leq h < \delta$. Since $V(0)$ is constant on $\mathcal{H}$ and $X, Y \in H$ for some $H \in \mathcal{H}$, this yields $\lim_{h \searrow 0} \frac{1}{h}(V_X(h) - V_Y(h)) = F_X(V(0)) - F_Y(V(0))$. Noting that $V_X(h) = V_Y(h)$ for all $0 \leq h < \delta$ because $V(0)$ is constant on $\mathcal{H}$, we thus infer $F_X(V(0)) - F_Y(V(0)) = 0$. ◀

Proof of Proposition 16. We first note that $\mathcal{R}$ is an equivalence relation over S, as it is the transitive closure of the union of equivalence relations over S. For $i \in I$, set $\mathcal{H}_i = S/\mathcal{R}_i$ and $\mathcal{H} = S/\mathcal{R}$. Note that, for all $i \in I$, any $H \in \mathcal{H}$ is a union of blocks of $\mathcal{H}_i$. From this, in turn, it is easy to see that any $\mathcal{M} \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}}$ can be written

as a union of blocks of $\{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}_i}$. Observe that, for any $(y_0, y_1) \in \mathcal{R}$, there exist $x_0, \ldots, x_k \in S$ such that $x_0 R_{i_0} x_1 R_{i_1} \ldots R_{i_{k-1}} x_k$ with $y_0 = x_0$, $y_1 = x_k$ and $i_j \in I$ for all $0 \leq j \leq k-1$. Noting that $x_j R_{i_j} x_{j+1}$ for all $0 \leq j \leq k-1$, we infer that $\mathbf{fr}[x_j, \mathcal{M}] = \mathbf{fr}[x_{j+1}, \mathcal{M}]$ for all $\mathcal{M} \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}_{i_j}}$. Thus, we infer that $\mathbf{fr}[x_0, \mathcal{M}] = \mathbf{fr}[x_1, \mathcal{M}] = \ldots = \mathbf{fr}[x_{k-1}, \mathcal{M}] = \mathbf{fr}[x_k, \mathcal{M}]$ for all $\mathcal{M} \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}}$. The remainder of the claim follows then from Lemma 26. ◄

Proof of Theorem 17. Define $[\![\rho]\!]_V := \prod_{X \in S} V_X^{\rho(X)}$ and set $Q := \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}}$. Fix some arbitrary $X_i, X_j \in H$ and $H \in \mathcal{H}$ and note that

$$F_{X_k}(V) = \sum_{\rho \xrightarrow{\alpha} \pi \in R} \alpha(\pi(X_k) - \rho(X_k))[\![\rho]\!]_V = \sum_{[\rho_0] \in Q} \sum_{\rho \in [\rho_0]} \mathbf{fr}(X_k, \rho)[\![\rho]\!]_V$$

$$= \sum_{[\rho_0] \in Q} \underbrace{\left(\sum_{\rho \in [\rho_0]} \mathbf{fr}(X_k, \rho) \right)}_{c(X_k, [\rho_0])} [\![\rho_0]\!]_V$$

whenever V is constant on $\mathcal{H}$. Observe also that the function $V \mapsto F_{X_k}(V)$, where V is constant on $\mathcal{H}$, defines a polynomial in $|Q|$ variables with the monomials $\{c(X_k, [\rho_0]) \cdot [\![\rho_0]\!]_V \mid [\rho_0] \in Q\}$. At last, recall that the multi dimensional version of Taylor's theorem implies that two real polynomials are equivalent if and only if they have the same monomials. Thus, $F_{X_i}(V) = F_{X_j}(V)$ for all V that are constant on $\mathcal{H}$ if and only if $c(X_i, [\rho_0]) = c(X_j, [\rho_0])$ for all $[\rho_0] \in Q$. ◄

Proof of Theorem 23. By encoding $\mu : S \to S$, ρ and π as arrays of length $|S|$, it is easy to see that the first operation needs at most $\mathcal{O}(|R| \cdot |S|)$ steps. For the second operation, note that the renaming of species according to μ can be done in again in $\mathcal{O}(|R| \cdot |S|)$. However, since elements of $\mathcal{MS}(S)$ are stored as ordered lists to allow for performant processing, the second operation needs $\mathcal{O}(|R| \cdot |S| \cdot \log(|S|))$. To accomplish the third operation, instead, we first sort the reactions with respect to the lexicographical order which takes $\mathcal{O}(|R| \cdot \log(|R|) \cdot |S|)$. Afterwards, the rates of the reactions that coincide in reactants and products can be summed in $\mathcal{O}(|R| \cdot |S|)$.

We now turn to the correctness of algorithm. Let $(X, V) \mapsto G_X^i(V)$ denote the vector field that arises from R after applying (B1), ..., (Bi), with G^0 being F itself. We next prove that $G_{X_k}^i(\hat{V} \circ \mu_S) = F_{X_k}(\hat{V} \circ \mu_S)$ for all $\hat{V} \in \mathbb{R}_{\geq 0}^{\hat{S}}$, $X_k \in \hat{S}$ and $i \in \{1, 2\}$. For this, let us first apply the reaction changes $\rho \to^\alpha \pi \mapsto \rho \to^\alpha \tilde{\pi}$ of (B1). Then, $G_{X_k}^0(\hat{V} \circ \mu) = G_{X_k}^1(\hat{V} \circ \mu)$ because of $\pi(X_k) - \rho(X_k) = \tilde{\pi}(X_k) - \rho(X_k)$. Let us now consider a reaction changes $\rho \to^\alpha \pi \mapsto \tilde{\rho} \to^\alpha \tilde{\pi}$ of (B2). Then, $G_{X_k}^1(\hat{V} \circ \mu) = G_{X_k}^2(\hat{V} \circ \mu)$ because $\prod_{X \in S}(\hat{V} \circ \mu)_X^{\rho(X)} = \prod_{X \in S}(\hat{V} \circ \mu)_X^{\tilde{\rho}(X)}$ and $\pi(X_k) - \rho(X_k) = \tilde{\pi}(X_k) - \tilde{\rho}(X_k)$. Since $G_{X_k}^2(\hat{V} \circ \mu) = G_{X_k}^3(\hat{V} \circ \mu)$ is trivially true, we infer the claim. ◄

A.3 Partition Refinement

The following auxiliary results will be needed to prove the correctness of Algorithm 1.

▶ **Lemma 30.** *Given a CRN (S, R), let $\mathcal{H}_1, \mathcal{H}_2$ be two partitions of S such that $\mathcal{H}_1$ is a refinement of $\mathcal{H}_2$. Then, the following holds true.*

1. $X_i \sim_{\mathcal{H}_1}^F X_j$ implies $X_i \sim_{\mathcal{H}_2}^F X_j$.
2. $X_i \sim_{\mathcal{H}_1}^B X_j$ implies $X_i \sim_{\mathcal{H}_2}^B X_j$.

Algorithm 2 Algorithm to calculate the quotient $A/\sim$. We assume that A is implemented as an array of objects $\{a[1], \ldots, a[n]\}$ where each object contains, among the actual data, a pointer that is initialized with zero at the beginning.

Require: A set A and an equivalence relation $\sim$ on A.
  for $i = 1$ to n **do**
    if $a[i].p$!= **null then**
      continue
    end if
    $a[i].p \leftarrow \&a[i]$
    for $j = i + 1$ to n **do**
      if $a[j].p ==$ **null** && $a[j] \sim a[i]$ **then**
        $a[j].p \leftarrow \&a[i]$
      end if
    end for
  end for

Proof. Let us assume that $X_i \sim^F_{\mathcal{H}_1} X_j$, i.e. it holds that $\mathbf{crr}[X_i, \rho] = \mathbf{crr}[X_j, \rho]$ and $\mathbf{pr}[X_i, \rho, H_1] = \mathbf{pr}[X_j, \rho, H_1]$ for all $\rho \in \mathcal{MS}(S)$ and $H_1 \in \mathcal{H}_1$. Since any $H_2 \in \mathcal{H}_2$ can be written as a disjoint union of blocks from $\mathcal{H}_1$, we thus infer that $X_i \sim^F_{\mathcal{H}_2} X_j$. Let us now assume that $X_i \sim^B_{\mathcal{H}_1} X_j$, i.e. it holds that $\mathbf{fr}[X_i, \mathcal{M}_1] = \mathbf{fr}[X_j, \mathcal{M}_1]$ for all $\mathcal{M}_1 \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}_1}$. Since any $\mathcal{M}_2 \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}_2}$ can be written as a disjoint union of blocks from $\{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}_1}$, we deduce that $X_i \sim^B_{\mathcal{H}_2} X_j$. ◄

▶ **Lemma 31.** *Let (S, R) be a CRN and $\mathcal{H}$ a partition of S. Then, the following holds.*

1. *$\mathcal{H}$ is an FB if and only if $X_i \sim^F_{\mathcal{H}} X_j$ for all $X_i, X_j \in H$ and $H \in \mathcal{H}$. Moreover, it holds that $\mathcal{H}$ is an FB if and only if $\mathcal{H} = S/(\sim^F_{\mathcal{H}} \cap \sim_{\mathcal{H}})$.*
2. *$\mathcal{H}$ is a BB if and only if $X_i \sim^B_{\mathcal{H}} X_j$ for all $X_i, X_j \in H$ and $H \in \mathcal{H}$. Moreover, it holds that $\mathcal{H}$ is a BB if and only if $\mathcal{H} = S/(\sim^B_{\mathcal{H}} \cap \sim_{\mathcal{H}})$.*

Proof. The first parts of 1. and 2. are straightforward. The second parts, instead, follow from the corresponding first parts by observing that $\mathcal{H} = S/(\sim^\chi_{\mathcal{H}} \cap \sim_{\mathcal{H}})$ if and only if $\mathcal{H}$ is a refinement of $S/\sim^\chi_{\mathcal{H}}$. ◄

The following auxiliary results will be needed to establish polynomial complexity of Algorithm 1. We implement $\rho \in \mathcal{MS}(S)$ as maps with keys in S. Moreover, we store a partition of a set A by means of a map that associates to each element of A a pointer to its representative. That is, each element of a partition block has a pointer to the representative of the block.

▶ **Lemma 32.** *Fix a CRN (S, R), pick $A \in \{S, R\}$ and assume that deciding $a_1 \sim a_2$ for some equivalence relation $\sim$ on A can be done in $\mathcal{O}(|R|^{e_1} \cdot |S|^{e_2})$ steps. Then, $A/\sim$ can be calculated in $\mathcal{O}(|A|^2 \cdot |R|^{e_1} \cdot |S|^{e_2})$ steps.*

Proof. It can be easily seen that Algorithm 2 calculates $A/\sim$ in $\mathcal{O}(|A|^2 \cdot |R|^{e_1} \cdot |S|^{e_2})$. ◄

▶ **Lemma 33.** *For a CRN (S, R) and a partition $\mathcal{H}$ of S, deciding $X \sim^F_{\mathcal{H}} Y$ can be done in $\mathcal{O}(|R|^2 \cdot |S|^2)$.*

Proof. We first note that, for a given $\rho \in \mathcal{MS}(S)$, deciding $\mathbf{crr}[X, \rho] = \mathbf{crr}[Y, \rho]$ can be done in $\mathcal{O}(|R| \cdot |S|)$ because the comparison of two $\rho_1, \rho_2 \in \mathcal{MS}(S)$ needs $\mathcal{O}(|S|)$. Similarly, for given $\rho \in \mathcal{MS}(S)$ and $Z \in S$, the calculation of $\mathbf{pr}[X, \rho, Z]$ can be done in $\mathcal{O}(|R| \cdot |S|)$. Note also that $\mathcal{D}(Z) := \{\rho \mid \exists Z + \rho \xrightarrow{\alpha} \pi \in R\}$ can be calculated in $\mathcal{O}(|R| \cdot |S|)$ steps. We are now in a position to infer the claim. For this, note that $\mathbf{crr}[X, \rho] = \mathbf{crr}[Y, \rho]$ for all $\rho \in \mathcal{MS}(S)$ if $\mathbf{crr}[X, \rho] = \mathbf{crr}[Y, \rho]$ for all $\rho \in \mathcal{D}(X) \cup \mathcal{D}(Y)$. Consequently, deciding whether $\mathbf{crr}[X, \rho] = \mathbf{crr}[Y, \rho]$ for all $\rho \in \mathcal{MS}(S)$ can be done in $\mathcal{O}(|R|^2 \cdot |S|)$ steps. Similarly, we note that $\mathbf{pr}[X, \rho, H] = \mathbf{pr}[Y, \rho, H]$ for all $\rho \in \mathcal{MS}(S)$ and $H \in \mathcal{H}$ if $\mathbf{pr}[X, \rho, H] = \mathbf{pr}[Y, \rho, H]$ for all $\rho \in \mathcal{D}(X) \cup \mathcal{D}(Y)$ and $H \in \mathcal{H}$. Thus, deciding whether $\mathbf{pr}[X, \rho, H] = \mathbf{pr}[Y, \rho, H]$ for all $\rho \in \mathcal{MS}(S)$ and $H \in \mathcal{H}$ holds true can be done in $\mathcal{O}(|R|^2 \cdot |S|^2)$. ◀

▶ **Lemma 34.** *For a CRN (S, R) and a partition $\mathcal{H}$ of S, deciding $X \sim_{\mathcal{H}}^{B} Y$ can be done in $\mathcal{O}(|R|^2 \cdot |S|)$.*

Proof. Note that we can decide whether $\rho \approx_{\mathcal{H}} \sigma$ in $\mathcal{O}(|S|)$. Thus, since $\mathcal{R} = \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}$ can be calculated in $\mathcal{O}(|R| \cdot |S|)$ steps, Lemma 32 implies that $Q = \mathcal{R}/\approx_{\mathcal{H}}$ can be calculated in $\mathcal{O}(|R|^2 \cdot |S|)$. Moreover, we note that, for a given $\rho \in \mathcal{MS}(S)$, deciding whether $\mathbf{fr}[X, \rho] = \mathbf{fr}[Y, \rho]$ can be done in $\mathcal{O}(|R| \cdot |S|)$. Consequently, deciding whether $X \sim_{\mathcal{H}}^{B} Y$ can be done in $\mathcal{O}(|R|^2 \cdot |S|)$. ◀

We now are in a position to prove Theorem 25.

Proof of Theorem 25. For the proof of correctness, let us assume that $\mathcal{H}$ denotes the coarsest bisimulation that refines $\mathcal{H}_0 := \mathcal{G}$ and define $\mathcal{H}_{k+1} := S/(\sim_{\mathcal{H}_k}^{\chi} \cap \sim_{\mathcal{H}_k})$ for all $k \geq 0$. Then, the sequence $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \dots$ is such that

1. $\mathcal{H}$ is a refinement of $\mathcal{H}_k$
2. $\mathcal{H}_k$ is a refinement of $\mathcal{H}_{k-1}$

for all $k \geq 1$. We prove this by induction on k.

- $k = 1$: Since $\mathcal{H}$ is a refinement of $\mathcal{H}_0$, Lemma 30 ensures the first claim. The second claim is trivial.
- $k \to k+1$: Thanks to the fact that $\mathcal{H}$ is a refinement of $\mathcal{H}_k$ by induction, Lemma 30 ensures the first claim. The second claim is trivial.

Since $\mathcal{H}$ is a refinement of any $\mathcal{H}_k$, it holds that $\mathcal{H} = \mathcal{H}_k$ whenever $\mathcal{H}_k$ is a bisimulation. Thanks to the fact that $\mathcal{H}_k$ is a refinement of $\mathcal{H}_{k-1}$ for all $k \geq 1$ and S is finite, we can fix the smallest $k \geq 1$ such that $\mathcal{H}_k = \mathcal{H}_{k-1}$. Since this implies $\mathcal{H}_{k-1} = \mathcal{H}_k = S/(\sim_{\mathcal{H}_{k-1}}^{\chi} \cap \sim_{\mathcal{H}_{k-1}})$, Lemma 31 yields the claim. We now turn to the complexity analysis. Note that since deciding $X \sim_{\mathcal{H}} Y$ needs a constant number of steps, Lemma 33 and Lemma 34 imply that $X \sim_{\mathcal{H}}^{\chi} Y$ can be decided in $\mathcal{O}(|R|^2 \cdot |S|^2)$ steps. Thus, Lemma 32 ensures that $S/\sim_{\mathcal{H}}^{\chi}$ can be calculated in $\mathcal{O}(|R|^2 \cdot |S|^4)$ steps. Noting that the algorithm makes at most $|S|$ iterations, we conclude that the algorithm needs at most $\mathcal{O}(|R|^2 \cdot |S|^5)$ steps. ◀

A.4 κ-encodings discussed in Section 6

M1-M4

We start providing the κ-encoding of model M4 of Table 1, whose original (BNGL) version has been taken from [30]. M4 is a special case in which almost no changes are necessary to convert it in κ. The encoding is given in Listing 1, where we omit unnecessary details. Each κ rule is preceded by the corresponding original BNGL rule (starting with #). The encodings for all other models of the same benchmark, M1-M3, are similar and thus we omit them here. They are available for download at `http://sysma.imtlucca.it/crnreducer/`.

🟨 **Listing 1** Encoding of M4 from BNGL into κ

```
1   ###  SITE 1 ###
2   #E(s) + S(p1~U) <-> E(s!1).S(p1~U!1)
3   E(s) , S(p1~U) <-> E(s!1),S(p1~U!1)
4
5   #E(s!1).S(p1~U!1) -> E(s) + S(p1~P)
6   E(s!1),S(p1~U!1) -> E(s) , S(p1~P)
7
8   #F(s) + S(p1~P) <-> F(s!1).S(p1~P!1)
9   F(s) , S(p1~P) <-> F(s!1),S(p1~P!1)
10
11  #F(s!1).S(p1~P!1) -> F(s) + S(p1~U)
12  F(s!1),S(p1~P!1) -> F(s) , S(p1~U)
13
14
15  ###  SITE 2 ###
16  #E(s) + S(p2~U) <-> E(s!1).S(p2~U!1)
17  E(s) , S(p2~U) <-> E(s!1),S(p2~U!1)
18
19  #E(s!1).S(p2~U!1) -> E(s) + S(p2~P)
20  E(s!1),S(p2~U!1) -> E(s) , S(p2~P)
21
22  #F(s) + S(p2~P) <-> F(s!1).S(p2~P!1)
23  F(s) , S(p2~P) <-> F(s!1),S(p2~P!1)
24
25  #F(s!1).S(p2~P!1) -> F(s) + S(p2~U)
26  F(s!1),S(p2~P!1) -> F(s) , S(p2~U)
```

M6-M7

Models M6 and M7 are not directly encodable in κ, as they contain the molecule $Lig(l, l)$ having two identical binding sites l, which is forbidden in κ. In fact, BioNetGen supports the aggregation of identical binding sites [3]. Intuitively, the rule

$$Rec(a) + Lig(l, l) \rightarrow Rec(a!1).Lig(l!1, l) \tag{26}$$

gives rise to only one chemical complex in the underlying CRN, $Rec(a!1).Lig(l!1, l)$. This represents the (forward and backward) canonical representative of a ligand bound to a single receptor $Rec(a)$. To see this, let us rename the two sites and *expand* the rule appropriately:

$$Rec(a) + Lig(l_1, l_2) \rightarrow Rec(a!1).Lig(l_1!1, l_2)$$
$$Rec(a) + Lig(l_1, l_2) \rightarrow Rec(a!1).Lig(l_1, l_2!1) \tag{27}$$
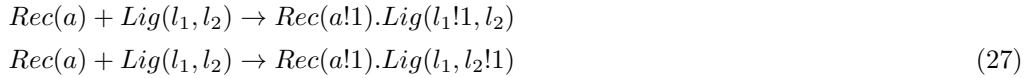
Then, this underlying CRN will distinguish the two sites. However, applying either of our bisimulations leads to the CRN for Equation (26).

We remark that the sizes of M6 and M7 given in Table 1 already account for the aggregations obtained with BioNetGen. Nevertheless, our bisimulations allow for further (significant) reductions. E.g., part of the reductions for M6 are due to the presence of $Rec(a, b, g_1, g_2)$, a molecule with symmetric sites g_1 and g_2, similarly to those of M4 discussed in Section 6.

In order to encode M6 and M7 in BNGL we thus first expanded the BNGL rules of the two models similarly to as done to that in Equation (26) to obtain the ones in Equation (27), and then we encoded the expanded BNGL rules in κ. This led to initial CRNs with 21174 and 4858 species, respectively. In these cases fragmentation returned more fragments than

---

[3] See Faeder, J.R., Blinov, M.L., Hlavacek, W.S.: Rule-Based Modeling of Biochemical Systems with BioNetGen. In: Maly, I.V. (ed.) Systems Biology, Methods in Molecular Biology, vol. 500, pp. 113–167. Humana Press (2009)

species (58040 and 10930 fragments, respectively). Instead, resorting to our bisimulations we obtained the same reductions as for the original models. For presentation reasons, we do not provide the κ-encodings of the expansions of M6 and M7, which however are available for download.

M12

We conclude this appendix providing in Listing 2 the κ-encoding of M12, whose BNGL version has been taken from [25]. As for M4, each κ rule is preceded by the corresponding original BNGL rule.

■ **Listing 2** Encoding of M12 from BNGL to κ

```
1   #MAP3K(s,S~I) -> MAP3K(s,S~A)
2   MAP3K(s,S~I) -> MAP3K(s,S~A)
3
4   #MAP3K(s,S~A)+Scaff(map3k) <-> MAP3K(s!1,S~A).Scaff(map3k!1)
5   MAP3K(s,S~A),Scaff(map3k) <-> MAP3K(s!1,S~A),Scaff(map3k!1)
6
7   #MAP3K(s!1,S~I).Scaff(map3k!1) -> MAP3K(s,S~I)+Scaff(map3k)
8   MAP3K(s!1,S~I),Scaff(map3k!1) -> MAP3K(s,S~I),Scaff(map3k)
9
10  #MAP2K(s,R1~Y,R2~Y)+Scaff(map2k) <-> MAP2K(s!1,R1~Y,R2~Y).Scaff(map2k!1)
11  MAP2K(s,R1~Y,R2~Y),Scaff(map2k) <-> MAP2K(s!1,R1~Y,R2~Y),Scaff(map2k!1)
12
13  #MAP2K(s,R1~Yp,R2~Y)+Scaff(map2k) <-> MAP2K(s!1,R1~Yp,R2~Y).Scaff(map2k!1)
14  MAP2K(s,R1~Yp,R2~Y),Scaff(map2k) <-> MAP2K(s!1,R1~Yp,R2~Y),Scaff(map2k!1)
15
16  #MAP2K(s,R1~Y,R2~Yp)+Scaff(map2k) <-> MAP2K(s!1,R1~Y,R2~Yp).Scaff(map2k!1)
17  MAP2K(s,R1~Y,R2~Yp),Scaff(map2k) <-> MAP2K(s!1,R1~Y,R2~Yp),Scaff(map2k!1)
18
19  #MAP2K(s,R1~Yp,R2~Yp)+Scaff(map2k) <-> MAP2K(s!1,R1~Yp,R2~Yp).Scaff(map2k!1)
20  MAP2K(s,R1~Yp,R2~Yp),Scaff(map2k) <-> MAP2K(s!1,R1~Yp,R2~Yp),Scaff(map2k!1)
21
22  #MAPK(s,R1~Y,R2~Y)+Scaff(mapk) <-> MAPK(s!1,R1~Y,R2~Y).Scaff(mapk!1)
23  MAPK(s,R1~Y,R2~Y),Scaff(mapk) <-> MAPK(s!1,R1~Y,R2~Y),Scaff(mapk!1)
24
25  #MAPK(s,R1~Yp,R2~Y)+Scaff(mapk) <-> MAPK(s!1,R1~Yp,R2~Y).Scaff(mapk!1)
26  MAPK(s,R1~Yp,R2~Y),Scaff(mapk) <-> MAPK(s!1,R1~Yp,R2~Y),Scaff(mapk!1)
27
28  #MAPK(s,R1~Y,R2~Yp)+Scaff(mapk) <-> MAPK(s!1,R1~Y,R2~Yp).Scaff(mapk!1)
29  MAPK(s,R1~Y,R2~Yp),Scaff(mapk) <-> MAPK(s!1,R1~Y,R2~Yp),Scaff(mapk!1)
30
31  #MAPK(s!1,R1~Yp,R2~Yp).Scaff(mapk!1) -> MAPK(s,R1~Yp,R2~Yp) + Scaff(mapk)
32  MAPK(s!1,R1~Yp,R2~Yp),Scaff(mapk!1) -> MAPK(s,R1~Yp,R2~Yp) , Scaff(mapk)
33
34  #MAP3K(s!1,S~A).Scaff(map3k!1,map2k!2).MAP2K(s!2,R1~Y) -> MAP3K(s!1,S~A).Scaff(map3k!1,
       map2k!2).MAP2K(s!2,R1~Yp)
35  MAP3K(s!1,S~A),Scaff(map3k!1,map2k!2),MAP2K(s!2,R1~Y) -> MAP3K(s!1,S~A),Scaff(map3k!1,
       map2k!2),MAP2K(s!2,R1~Yp)
36
37  #MAP3K(s!1,S~A).Scaff(map3k!1,map2k!2).MAP2K(s!2,R2~Y) -> MAP3K(s!1,S~A).Scaff(map3k!1,
       map2k!2).MAP2K(s!2,R2~Yp)
38  MAP3K(s!1,S~A),Scaff(map3k!1,map2k!2),MAP2K(s!2,R2~Y) -> MAP3K(s!1,S~A),Scaff(map3k!1,
       map2k!2),MAP2K(s!2,R2~Yp)
39
40  #MAPK(s!1,R1~Y).Scaff(mapk!1,map2k!2).MAP2K(s!2,R1~Yp,R2~Yp) -> MAPK(s!1,R1~Yp).Scaff(
       mapk!1,map2k!2).MAP2K(s!2,R1~Yp,R2~Yp)
41  MAPK(s!1,R1~Y),Scaff(mapk!1,map2k!2),MAP2K(s!2,R1~Yp,R2~Yp) -> MAPK(s!1,R1~Yp),Scaff(
       mapk!1,map2k!2),MAP2K(s!2,R1~Yp,R2~Yp)
42
43  #MAPK(s!1,R2~Y).Scaff(mapk!1,map2k!2).MAP2K(s!2,R1~Yp,R2~Yp) -> MAPK(s!1,R2~Yp).Scaff(
       mapk!1,map2k!2).MAP2K(s!2,R1~Yp,R2~Yp)
44  MAPK(s!1,R2~Y),Scaff(mapk!1,map2k!2),MAP2K(s!2,R1~Yp,R2~Yp) -> MAPK(s!1,R2~Yp),Scaff(
       mapk!1,map2k!2),MAP2K(s!2,R1~Yp,R2~Yp)
45
46  #MAP3K(S~A) -> MAP3K(S~I)
47  MAP3K(S~A) -> MAP3K(S~I)
48
49  #MAP2K(R1~Yp) -> MAP2K(R1~Y)
50  MAP2K(R1~Yp) -> MAP2K(R1~Y)
51
52  #MAP2K(R2~Yp) -> MAP2K(R2~Y)
53  MAP2K(R2~Yp) -> MAP2K(R2~Y)
54
55  #MAPK(R1~Yp) -> MAPK(R1~Y)
56  MAPK(R1~Yp) -> MAPK(R1~Y)
57
58  #MAPK(R2~Yp) -> MAPK(R2~Y)
59  MAPK(R2~Yp) -> MAPK(R2~Y)
```