Normalizing Chemical Reaction Networks by Confluent Structural Simplification

Guillaume Madelaine[1,2], Elisa Tonello[3], Cédric Lhoussaine[1,2], and Joachim Niehren[2,4]

[1] University of Lille, France
[2] Cristal, Cnrs Umr 9189, Lille, France
[3] University of Nottingham, United Kingdom
[4] Inria, Lille, France

Abstract. Reaction networks can be simplified by eliminating linear intermediate species in partial steady states. In this paper, we study the question whether this rewrite procedure is confluent, so that for any given reaction network, a unique normal form will be obtained independently of the elimination order. We first contribute a counter example which shows that different normal forms of the same network may indeed have different structures. The problem is that different "dependent reactions" may be introduced in different elimination orders. We then propose a rewrite rule that eliminates such dependent reactions and prove that the extended rewrite system is confluent up to kinetic rates, i.e., all normal forms of the same network will have the same structure. However, their kinetic rates may still not be unique, even modulo the usual axioms of arithmetics. This might seem surprising given that the ODEs of these normal forms are equal modulo these axioms.

1 Introduction

Chemical reaction networks are widely used in systems biology for modeling the dynamics of biochemical molecular systems [4,6,1,11]. A chemical reaction network has a graph structure that can be identified with a Petri net [2]. Beside of this, it assigns to each of its reactions a kinetic rate that models the reaction's speed. Chemical reaction networks can either be given a deterministic semantics in terms of ordinary differential equations (ODEs), which describes the evolution of the average concentrations of the species of the network over time, or a stochastic semantics in terms of continuous time Markov chains, which defines the evolution of molecule distributions of the different species over time. In this paper, we focus on the deterministic semantics.

Reaction networks may become very large when modeling molecular biological systems in sufficient detail, see e.g. the examples in the BioModels database [8]. Therefore much effort has been spent on their simplification (see [18] for an overview). The traditional approach is by reducing the ODEs of the network by symbolic rewriting techniques [9,10]. While clearly beneficial, such approaches have the disadvantage that the simplified ODEs cannot always be translated

back to a reaction network [3], so that these simplifications cannot be understood directly as simplifications of biological systems.

Another major problem with large biological reaction networks is that precise kinetic rates are rarely available [14,16]. In the worst case, no kinetic information is available, so that no ODEs can be derived. The only simplifications that are possible in this case rely purely on the graph structure of the reaction network [12,17]. In a less extreme setting, the kinetic rates are given by arithmetic expressions with unknown parameters. In this case, the purely structural methods must be lifted so that they can properly account for the kinetic rates.

The common objective of the structural simplification methods is to eliminate intermediate species that are irrelevant to the external behaviour of the system. This can be done in an exact manner – when assuming partial steady states – so that the solutions of the ODEs of reaction networks are preserved [19,13,22]. It should be noticed that any structural reduction algorithm preserving the ODE's solutions necessarily induces an exact reduction method on the underlying ODE level. Indeed the above methods are based on the same idea, which is to resolve the partial steady state equation of some intermediate species along its concentration variable, so that this variable can be eliminated from the ODEs. The restriction that makes this possible is that the kinetic rates of the network's reactions are linear in the concentration of the intermediate species.

The structural reduction method for intermediate elimination from [13] removes the intermediates stepwise one by one. The approach of [22] is similar with an extension to rapid equilibrium assumption. The alternative method of [19] removes several intermediates simultaneously. We verified that both methods perform the same reductions when restricted to a single intermediate, even though these are computed by quite differently algorithms. The yet independent method from [17,18] also performs simultaneous elimination of intermediates, but not necessarily in a unique manner. The intermediates are eliminated from the reaction graph by computing elementary modes in a first step, and in a second, appropriate kinetic rates are assigned to reduced graph. Their method can also be applied in the nonlinear case, but then with some approximations.

In this paper, we study the question of whether the stepwise elimination of linear intermediates is confluent, so that for any given reaction network, a unique normal form will be obtained independently of the elimination order. If confluence would hold, one could compare reaction networks for equivalence, by computing and comparing their normal forms. Furthermore, the unique normal form would be the natural target for simultaneous reduction methods such as [18,19]. Indeed, a confluence statement was claimed in Section 5 of [19] (for the case without conservation laws), but without proof.

We first contribute a counter example which shows that the elimination of linear intermediates on the same network may lead to normal forms with different graph structure. This example contradicts the confluence statement from [19]. The problem is that different "dependent reactions" may be introduced in different elimination orders. We then propose a rewrite rule that eliminates such dependent reactions and prove that the extended rewrite system is confluent up

to kinetic rates, so that all normal forms of a same network will have the same structure. This yields a method to eliminate linear intermediates from a reaction graph in a unique manner, while no uniqueness result was stated in [17,18]. However, the kinetic rates may still not be unique, even not modulo the usual axioms of arithmetics. This might seem surprising given that the ODEs of these normal forms are equal modulo these axioms. Finally, we present an example reaction network from systems biology for the failure of confluence with respect to kinetic rates, that we found in the BioModels SBML database [8] with an implementation of our rewrite rules.

Our positive confluence result shows that the graph structure of reaction networks after intermediate and dependency reduction is unique, and thus potentially meaningful biologically. The two negative confluence results show that the situation may be different without dependency reduction, and also for the kinetic rates that can be assigned to the reactions of the reduced network.

All proofs and missing parts are available in the Appendix of the long version.

2 Confluence notions

We recall confluence notions and their relationships from the literature.

Let $(S, \sim)$ be a set with an equivalence relation and $\to \subseteq S \times S$ a binary relation. We define $\to^0 = \sim$ and $\to^k = \to \circ \to^{k-1}$ for all $k > 0$. The relation $\to^* = \cup_{k \geq 0} \to^k$ is called the reflexive transitive closure of $\to$. We write $\to^\epsilon = \to^1 \cup \to^0$, and $\leftarrow = \{(s, s') \mid s' \to s\}$.

Definition 1 (Confluence modulo). *We say that a binary relation $\to$ on $(S, \sim)$ is* confluent *if $\leftarrow^* \circ \to^* \subseteq \to^* \circ {}^*\leftarrow$, locally confluent if $\leftarrow \circ \to \subseteq \to^* \circ {}^*\leftarrow$, strongly confluent if $\leftarrow \circ \to \subseteq \to^\epsilon \circ \sim \circ {}^\epsilon\leftarrow$, and* uniformly confluent *if $\leftarrow \circ \to \subseteq \sim \cup (\to \circ \sim \circ \leftarrow)$.*

Clearly, uniform confluence implies strong confluence, and strong confluence implies local confluence. It is also folklore that there exist locally confluent relations that are not confluent, while strong confluence implies confluence [7]. Uniform confluence implies for any $s \in S$ that all complete reduction sequences starting with s have the same length [15], which may be ∞ though.

In this paper, we will always use binary relations that are terminating, i.e., for any $s \in S$ there exists a $k \geq 0$ such that $\{s' \mid s \to^k s'\} = \emptyset$, i.e., the length reduction sequences starting with s is bounded. It is well known that locally confluent and terminating relations are confluent (Newman's lemma).

We say that $\sim$ commutes with $\to$ if $\sim \circ \to \subseteq \to \circ \sim$.

Lemma 1. *If $\to$ is confluent for $(S, \sim)$ and commutes with $\sim$, then the relation $\sim \circ \to \circ \sim$ is confluent for $(S, =_S)$.*

3 Simplification of systems of equations

In this section, we recall the definition of arithmetic expressions and ordinary differential equations. It is well known that such systems can be inferred from

reaction networks with deterministic semantics and partial steady state assumptions. We will then show how to simplify such systems in a confluent manner by eliminating intermediate variables.

Systems of equations. Let $\mathbb{R}_+$ be the set of non-negative real numbers, and $\mathbb{N}_0 \subseteq \mathbb{R}_+$ the set of natural numbers including 0. Denote by *Vars* a countable set of variables for functions of type $\mathbb{R}_+ \to \mathbb{R}_+$, and by *Param* a set of parameters. We define the set of *arithmetic expression* as the terms $e, e' \in \mathit{Expr}$ with the following abstract syntax:

$$e, e' \in \mathit{Expr} ::= x \mid k \mid n \mid e + e' \mid e * e' \mid 1/e \mid -e$$

where $x \in \mathit{Vars}$, $k \in \mathit{Param}$, $n \in \mathbb{R}$. In the following, the expression $1/e$ is permitted only if e can never become zero, as explained below. For convenience, we will write ee' for $e * e'$; e/e' for $e * (1/e')$, $e - e'$ for $e + (-e')$ and e^n for $e * \ldots * e$ with n repetitions of e.

We map variables to functions on non-negative real numbers, and parameters to positive (different from 0), which are identified with positive constant functions on non-negative real numbers. Given an assignment $\alpha : (\mathit{Vars} \to (\mathbb{R}_+ \to \mathbb{R}_+)) \cup (\mathit{Param} \to \mathbb{R}_+^*)$, any expression $e \in \mathit{Expr}$ can be interpreted as a function $[\![e]\!]_\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ in the usual way.

A *system of equations* S is a combination of equations and constraints, with some existential variables, defined as follows:

$$S ::= dx/dt = e \mid x = e \mid \mathit{nzero}(e) \mid \mathit{cst}(e) \mid S \wedge S' \mid \exists x.\ S.$$

$dx/dt = e$ is an *ordinary differential equation (ODE)*, and $x = e$ an *arithmetic equation*, for the variable x and with an expression $e \in \mathit{Expr}$. The *non-zero constraint* $\mathit{nzero}(e)$ is satisfied by an assignment α if e is never equal to zero, that is $\forall t.\ [\![e]\!]_\alpha(t) \neq 0$. The *positive constant constraint* $\mathit{cst}(e)$ is satisfied by a variable assignment α if $[\![e]\!]_\alpha$ is a positive constant function. And $\exists x.S$ allows us to *existentially quantify* some variables, that we actually want to remove to simplify S. We denote by $\mathit{Vars}(e)$ the set of variables of an expression e and by $\mathit{Vars}(S)$ the set of free variables of a system S. The *set of solutions* of a system of equations S is the set of assignments on the free variables of S that make S true, that is $\mathit{sol}(S) = \{\alpha \mid [\![S]\!]_\alpha = \mathit{true}\}$.

Example 1. The system of equations in Fig. 1 contains 4 ODEs for the variables $\{x_A, x_B, x_C, x_D\}$, and two arithmetic equations and positive constant constraints for the existentially quantified variables $\tilde{x} = \{x_Y, x_Z\}$.

Similar systems. We now define a syntactic notion of similarity between systems of equations, so that similar systems will have the same solutions. The *similarity relation* $\sim$ on arithmetic expressions is the least congruence that includes the usual arithmetic axioms of a field: commutativity and associativity of $+$ and $*$, removal of neutral elements 0 in sums and 1 in products, uniqueness and laws of inverses for $-$, distributivity, and simplification of real numbers. Similarity is decidable, by rewriting expressions to a fraction of polynomials, with the same denominator, and comparing the numerators.

$$\exists x_Y, x_Z.\ \begin{cases} \frac{dx_{\mathrm{A}}}{dt} = -(k_1+k_2)x_{\mathrm{A}} & \wedge\ \frac{dx_{\mathrm{B}}}{dt} = k_3 x_{\mathrm{Y}} & \wedge\ x_{\mathrm{Y}} = \frac{k_1}{k_3+k_5}x_{\mathrm{A}} + \frac{k_6}{k_3+k_5}x_{\mathrm{Z}} \wedge \\ \frac{dx_{\mathrm{C}}}{dt} = (k_4+k_5)x_{\mathrm{Y}} & \wedge\ \frac{dx_{\mathrm{D}}}{dt} = k_6 x_{\mathrm{Z}} & \wedge\ x_{\mathrm{Z}} = \frac{k_2}{k_6}x_{\mathrm{A}} + \frac{k_5}{k_6}x_{\mathrm{Y}} \wedge \\ cst(x_{\mathrm{Y}}) & \wedge\ cst(x_{\mathrm{Z}}) & \end{cases}$$

Fig. 1. The system of equations $S(N_X)$.

We always identify arithmetic expressions up to similarity (rather than syntactic equality), i.e., we rewrite modulo $\sim$. Given an assignment α, two similar expressions $e \sim e'$ have trivially the same interpretation $[\![e]\!]_\alpha = [\![e']\!]_\alpha$. The similarity relation is lifted to systems of equations in the obvious manner.

Safe linear systems. We will consider only *valid systems of equations* in which there is exactly one arithmetic equation per quantified variable and at most one ODE for all others. We also assume that the systems are linear in the existentially quantified variables as defined below, but not necessarily in the others:

Definition 2. *Given a sequence of variables $\tilde{x} = x_1, \ldots, x_n$, an expression e' is called $\tilde{x}$-linear if e' is similar to some expression $e + \sum_{1\leq i\leq n} x_i e_i$, where e and e_i do not contain any variables from $\tilde{x}$. We call a system $\exists\tilde{x}.S$* linear (in the quantified variables) *if for any quantified variable $x \in Vars(\tilde{x})$, the system S is similar to some system $x = e \wedge S'$ where e is an $\tilde{x}$-linear expression.*

In order to always avoid division by zero during the repeated elimination of quantified variables to come (see Lemmas 2 and 3), we introduce the following safety restriction of linear systems, which will be satisfied most of the time in the applications. Without this restriction, the simplification procedure could be shown to be only partially correct, similarly to [19].

Definition 3. *Let S be a system $\exists x_1, \ldots, x_n$. S' that is linear in the quantified variables, such that S' has the form $\bigwedge_{1\leq i\leq n} x_i = e^i + \sum_{1\leq j\leq n} x_j e^i_j \ \wedge S''$. We define a set expression $L_{S'}$ in which x and y are fresh variables:*

$$L_{S'} =_{df} \{\ (x,y) \mid \bigvee_{1\leq i,j\leq n} x = x_i \wedge y = x_j \wedge nzero(e^j_i)\ \}.$$

For any assignment of the free variables in the subexpressions e^i_j, the set expression $L_{S'}$ denotes a binary relation, that we call the linking relation of S'. We call the system S safe *if S' entails the following formula:*

$$S' \models \bigwedge_{i=1}^{n} \bigvee_{k=1}^{n} L^*_{S'}(x_i, x_k) \wedge nzero(e^k)\ \wedge\ (\ e^i \geq 0\ \wedge\ \bigwedge_{j=1}^{n} e^i_j \geq 0\ ).$$

We denote by SafeLin the set of safe linear systems of equations.

Simplifying safe linear systems. We want to simplify safe linear systems of equations by removing existentially quantified variables, while preserving the

$$\frac{x \notin \mathit{Vars}(e)}{\exists x.\ (S \wedge x = e) \quad \Rightarrow \quad S[x := e]_l} \ (\textsc{Quantified variable})$$

Fig. 2. Elimination of an existentially quantified variable x in a system of equations.

solutions. To do that, given an expression $x = e$ for a quantified variable x, we will substitute x by e, as described in the simplification rule in Fig. 2.

A *substitution* $[x_1 := e]$ is the replacement of any occurrences of x_1 by the expression e. Additionally, we also want to preserve the linearity and safety. Therefore, we define a *linear substitution*, that rewrites arithmetic expressions into linear ones after the substitution. Formally, given a $\tilde{x}$-linear expression $e \sim e^1 + x_2 e_2^1 + \sum_{3 \leq i \leq n} x_i e_i^1$ and an equation $E_2 = (x_2 = e^2 + x_1 e_1^2 + \sum_{3 \leq i \leq n} x_i e_i^2)$, with $\tilde{x} = \{x_1, \ldots, x_n\}$, the linear substitution of x_1 by e in E_2 is:

$$E_2[x_1 := e]_l = (x_2 = \frac{e^1 e_1^2 + e^2}{1 - e_1^2 e_2^1} + \sum_{3 \leq i \leq n} x_i \frac{e_i^1 e_1^2 + e_i^2}{1 - e_1^2 e_2^1}) \wedge \mathit{nzero}(1 - e_1^2 e_2^1)$$

The idea is to a) substitute x_1 by e in the equation of x_2, b) bring the factor $e_1^2 e_2^1 x_2$ from the right to the left, c) factorize the x_2, and d) divide by the factor $1 - e_1^2 e_2^1$ of x_2 we obtained.

Lemma 2. *If S is safe and with the above equations then $S \models \mathit{nzero}(1 - e_1^2 e_2^1)$.*

We define $S[x_1 := e]_l$ by replacing x_1 by e in the ODEs and the constraints of S and by performing the linear substitution as above to all nondifferential equations of S. The relation $S \Rightarrow S'$ defined in Fig. 2 simplifies a safe linear system S to S': a quantified variable is eliminated by applying a linear substitution.

Lemma 3. *The simplification of a safe linear system is a safe linear system.*

Lemma 4. *The simplification preserves the solutions of safe linear systems: if $S \Rightarrow S'$, then $\mathit{sol}(S) = \mathit{sol}(S')$.*

Example 2. For instance, in the system from Example 1, we can substitute the intermediate variable x_{Y} by $e = \frac{k_1}{k_3 + k_5} x_{\mathrm{A}} + \frac{k_6}{k_3 + k_5} x_{\mathrm{Z}}$. Since we still have the constraint $\mathit{cst}(x_{\mathrm{Z}})$, the constraint $\mathit{cst}(e)$ can be simplified into $\mathit{cst}(x_{\mathrm{A}})$. The never-zero constraint $\mathit{nzero}(1 - \frac{k_5 k_6}{(k_3 + k_5) k_6})$ is similar to $\mathit{nzero}((k_3 + k_5) k_6 - k_5 k_6)$ and then $\mathit{nzero}(k_3 k_6)$, and therefore is always true, and can be removed. We obtain the system depicted in Fig. 3 (left). By doing the same with the variable x_{Z}, we obtain the system in Fig. 3 (right). Note that we used the fact that $k_6/k_6 \sim 1$, that is always true, since parameters are assigned to positive numbers.

For safe linear systems, this simplification modulo similarity is confluent, implying that whatever the order adopted for the elimination of quantified variables, it is always possible to find the same fully simplified system, modulo similarity. We actually establish uniform confluence, implying that any simplification leading to the fully simplified system will have the same number of steps.

$$\exists x_Z.\ \left\{\begin{array}{l} \frac{dx_A}{dt} = -(k_1+k_2)x_A \qquad \frac{dx_D}{dt} = k_6 x_Z \\ \frac{dx_B}{dt} = \frac{k_1k_3}{k_3+k_5}x_A + \frac{k_3k_6}{k_3+k_5}x_Z \\ \frac{dx_C}{dt} = \frac{k_1(k_4+k_5)}{k_3+k_5}x_A + \frac{k_6(k_4+k_5)}{k_3+k_5}x_Z \\ x_Z = \frac{k_1k_5+k_2k_3+k_2k_5}{k_3k_6}x_A \\ cst(x_A) \wedge cst(x_Z) \end{array}\right. \quad \left\{\begin{array}{l} \frac{dx_A}{dt} = -(k_1+k_2)x_A \\ \frac{dx_B}{dt} = (k_1+k_2)x_A \\ \frac{dx_C}{dt} = \frac{(k_1+k_2)(k_4+k_5)}{k_3}x_A \\ \frac{dx_D}{dt} = \frac{k_1k_5+k_2k_3+k_2k_5}{k_3}x_A \\ cst(x_A) \end{array}\right.$$

Fig. 3. Simplifications of $S(N_X)$.

Theorem 1. *The binary relation* $\Rightarrow$ *on* $(SafeLin, \sim)$ *is uniformly confluent.*

4 Reaction networks

In this section, we introduce reaction networks, intermediate species, and the interpretation of a network as a system of equations.

Let *Spec* be a countable set of *molecular species* ranged over by A. We associate to each species A a *concentration variable* x_A, and denote the set of these variables by $Vars = \{x_A \mid A \in Spec\}$. A kinetic expression is a non-negative arithmetic expression on variables *Vars*, i.e. for any non-negative assignment α for the concentrations, $[\![e]\!]_\alpha(t) \geq 0$ for all t.

We define a *(chemical) solution* $s \in Sol : Spec \to \mathbb{N}_0$ as a multiset of molecular species, i.e. a function from species to natural numbers, with finite support. Given numbers $n_1, \ldots, n_k$, we denote by $n_1A_1 + \cdots + n_kA_k$ the solution that contains n_i molecules of species A_i for $1 \leq i \leq k$, and 0 molecules of others species. Given $s_1, s_2 \in Solutions$, their intersection is defined for any A by $(s_1 \cap s_2)(A) = min(s_1(A), s_2(A))$. A *kinetic reaction* $r = (s_1 \rightarrow s_2;\ e)$ is a pair composed of a *reaction* $s_1 \rightarrow s_2$ and a kinetic expression $e \in Expr$. The reaction transforms the solution s_1, called *reactants*, into the solution s_2, called *products*. The reaction vector vr_r of the reaction r is defined for any $A \in Spec$ by $vr_r(A) = s_2(A) - s_1(A)$. We denote by $kin(r) = e$ the kinetic expression of r.

Given a reaction $r = (s_1 \rightarrow s_2;\ e)$ and the solution $s = s_1 \cap s_2$, the *normalization* of r is the reaction $(s_1 - s \rightarrow s_2 - s;\ e)$. In the following, we always assume that every reaction is normalized, and normalization is implicitly applied after every simplification. A *reaction network* N is composed of normalized kinetic reactions, constraints, and bound species (that we want to remove):

$$N ::= r \mid cst(e) \mid N \wedge N' \mid \exists X.\ N$$

We assume the usual structural congruence rules for conjunction and existential quantification. We denote by $C(N)$ the set of constraints of N.

Once again, we need to add some conditions on the bound species, called *intermediate species*, in order to be able to fully remove them in a confluent way. We usually denote by $\mathcal{U}$ the intermediate species, and by $\bar{\mathcal{U}}$ the other species.

$$\exists \tilde{x}.\ \left[\frac{dx_A}{dt} = \sum_{r \in N} vr_r(A) kin(r)\right]_{A \in \bar{\mathcal{U}}} \wedge \left[x_X = \frac{\sum_{\{r \in N | X \in Prod(r)\}} kin(r)}{\sum_{\{r \in N | X \in Cons(r)\}} kin(r)/x_X}\right]_{X \in \mathcal{U}} \wedge C(N)$$

Fig. 4. Definition of the system of equations $S(N)$, for the network N, with intermediate species $\mathcal{U}$ and with $\tilde{x} = \{x_X \mid X \in \mathcal{U}\}$.

Given a set $\mathcal{U}$ of molecules, and a reaction $r = (s_1 \rightarrow s_2;\ e)$, we define the *consumption* $Cons_{\mathcal{U}}(r) = s_1 \cap \mathcal{U}$ (resp. *production* $Prod_{\mathcal{U}}(r) = s_2 \cap \mathcal{U}$) of r with respect to $\mathcal{U}$ as the molecules of $\mathcal{U}$ that are consumed (resp. produced) by r.

A molecule $X \in \mathcal{U}$ is *output-connected* (resp. *input-connected*) in N with respect to $\mathcal{U}$ if $\exists r \in N$ with $Cons_{\mathcal{U}}(r) = \{X\}$ (resp. $Prod_{\mathcal{U}}(r) = \{X\}$) and either $Prod_{\mathcal{U}}(r) = \emptyset$ (resp. $Cons_{\mathcal{U}}(r) = \emptyset$), or $Prod_{\mathcal{U}}(r) = \{Y\}$ (resp. $Cons_{\mathcal{U}}(r) = \{Y\}$) with Y output-connected (resp. input-connected). This property will correspond to the safety property of quantified variables in linear systems of equations.

A reaction network $\exists \mathcal{U}.\ N$ is *linear* if the following properties hold:

- connectivity: for any $X \in \mathcal{U}$, X is output and input-connected in N,
- $\mathcal{U}$-stoichiometry: $\forall r \in N$, $|Cons_{\mathcal{U}}(r)| \leq 1$ and $|Prod_{\mathcal{U}}(r)| \leq 1$,
- $\mathcal{U}$-linearity: $\forall r \in N.\ Cons_{\mathcal{U}}(r) = \{X\} \Rightarrow kin(r) = x_X e$, with $\forall Y \in \mathcal{U}. x_Y \notin e$,
- kinetic non-interaction: $\forall r \in N$, $Cons_{\mathcal{U}}(r) = \emptyset$ and $Prod_{\mathcal{U}}(r) \neq \emptyset$ implies $x_X \notin kin(r)$ for any $X \in \mathcal{U}$,
- partial steady-state: $\forall X \in \mathcal{U}$, $cst(x_X) \in C(N)$.

In the following, we will only consider linear networks, and denote by *Nets* the set of linear reaction networks.

Given a linear network $N \in Nets$, we can define the interpretation of N in terms of a system of equations $S(N)$, as described in Fig. 4.

Lemma 5. *For any $N \in Nets$, the interpretation $S(N)$ is a (valid) safe linear system.*

Example 3. We consider the reaction network N_X in Fig. 5, with the reactions on the left and the reaction graph on the right. The set of species is $\{A, B, C, D, Y, Z\}$, where Y and Z are considered intermediates, and the set of reactions is $\{r_1, \ldots, r_6\}$. The parameters in the rates are some positive reals $k_1, \ldots, k_6$. All reactions have mass action kinetics, except for reaction r_4 which is activated by Y. Its associated system is $S(N_X)$, described in Example 1.

Given a network N, we can compute its system of equations $S(N)$, and then simplify it in a confluent way, as explained in Section 3. But we might sometimes be more interested in the network itself, rather than its system of equations. And unfortunately, rebuilding a reaction network from the equations can be difficult, and the network obtained is not unique [3] It seems then more appropriate to proceed with the simplification directly on the reaction network.

$$\exists Y, Z.\ cst(x_{\mathrm{Y}}) \qquad \wedge\ cst(x_{\mathrm{Z}})$$
$$\mathrm{r}_1 = (\mathrm{A} \to \mathrm{Y};\ k_1 x_{\mathrm{A}}) \qquad \mathrm{r}_4 = (0 \to \mathrm{C};\ k_4 x_{\mathrm{Y}})$$
$$\mathrm{r}_2 = (\mathrm{A} \to \mathrm{Z};\ k_2 x_{\mathrm{A}}) \qquad \mathrm{r}_5 = (\mathrm{Y} \to \mathrm{Z} + \mathrm{C};\ k_5 x_{\mathrm{Y}})$$
$$\mathrm{r}_3 = (\mathrm{Y} \to \mathrm{B};\ k_3 x_{\mathrm{Y}}) \qquad \mathrm{r}_6 = (\mathrm{Z} \to \mathrm{Y} + \mathrm{D};\ k_6 x_{\mathrm{Z}})$$

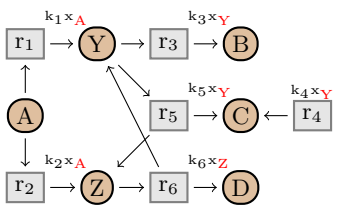

Fig. 5. The reaction network N_X.

$$\frac{e = \sum_{\{r \in N \mid X \in Prod(r)\}} kin(r) \qquad e' = \sum_{\{r \in N \mid X \in Cons(r)\}} kin(r) \qquad e' = x_X e''}{\bigwedge_{\{r, r' \in N \mid X \in Prod(r) \cap Cons(r')\}} r \diamond_{e'} r'} \ (\textsc{Intermediate})$$
$$\exists X.\ N \Rrightarrow^{\mathrm{Inter}} \bigwedge_{\{r \in N \mid X \notin Prod(r) \cup Cons(r)\}} r[x_X := \frac{e}{e''}] \wedge C(N)[x_X := \frac{e}{e''}]$$

Fig. 6. Intermediate simplification rule, with $C(N)$ the constraints of N.

5 Elimination of intermediate species

In this section, we introduce the Intermediate simplification rule for reaction networks, and apply it to an example.

The (INTERMEDIATE) rule presented in Fig. 6 aims at removing an intermediate species $X \in \mathcal{U}$: any reaction r_{prod} that produces X is combined with any reaction r_{cons} that consumes X, and x_X is replaced by its value at steady state in the other reactions. This merging operation is achieved by the operator $\diamond_e$:

$$(s_1 \to s_2;\ e) \diamond_{e''} (s_1' \to s_2';\ e') = (s_1 + s_1' \to s_2 + s_2';\ \frac{ee'}{e''}).$$

Since we only consider normalized reactions, in merged reactions the intermediate molecule is implicitly discarded.

The interpretation $S(N)$ is a simulation from $(Nets, \Rrightarrow^{\mathrm{Inter}})$ to $(SafeLin, \Rightarrow)$:

Lemma 6. *Given a network $N \in Nets$, if $N \Rrightarrow^{Inter} N'$, then $S(N) \Rightarrow S(N')$.*

This implies as expected that both a network and its simplification have the same deterministic dynamics.

The next example shows that the rewriting system given by the elimination of intermediate species alone is not confluent, given that different dependent reactions may be produced for different elimination orders.

Example 4. Starting from network N_X from Fig. 5, we can either remove Y or Z and obtain the networks depicted in Fig. 7. If we first remove Z, then we obtain the reaction network N_{XZ}. From N_{XZ} we can eliminate the intermediate Y and obtain N_{XZY}. This network cannot be simplified any further. Alternatively, we

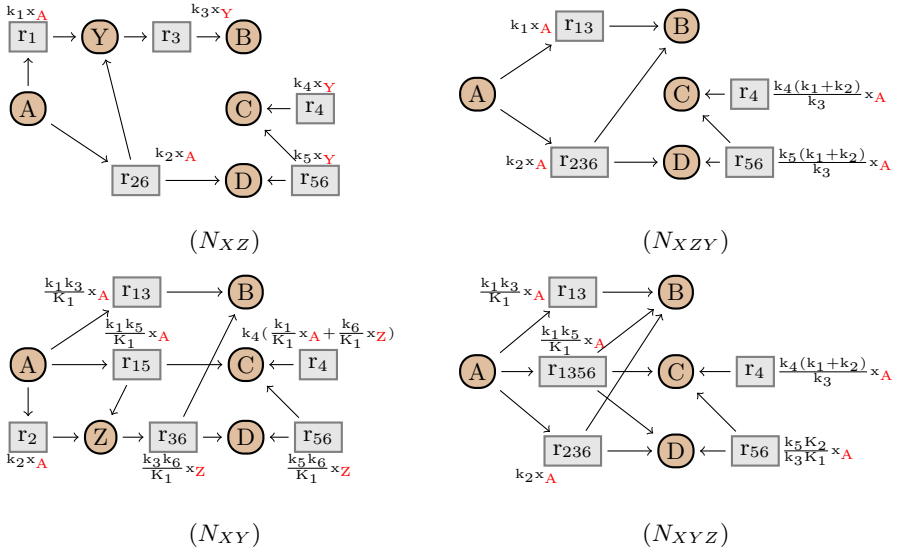

Fig. 7. Two elimination strategies to simplify N_X of Fig. 5: either first eliminate Z to obtain the network N_{XZ} and then Y to obtain N_{XZY}, or swap the elimination order to obtain first N_{XY} and then N_{XYZ}. Simplified networks N_{XZY} and N_{XYZ} are structurally different since the latter has the additional reaction r_{1356}. The new parameters are $K_1 = k_3 + k_5$ and $K_2 = k_1k_5 + k_2K_1$.

can eliminate Y from N_X in a first step, obtain N_{XY}, and then remove Z and obtain the network N_{XYZ}.

Unfortunately, N_{XYZ} and N_{XZY} do not have the same structure, since N_{XYZ} has an additional reaction r_{1356}, which is a combination of r_{13} and r_{56}. Such dependent reactions can be removed, as we will show in the next section.

6 Elimination of dependent reactions

In this section we clarify the notion of dependency between reactions, and introduce an additional simplification rule based on this notion. The addition of this rule is sufficient to establish confluence for the structure of simplified networks. However, we will show that this modification is not enough, in general, to guarantee full confluence.

We formalize the notion of dependency with respect to an initial set of reactions with the notion of *flux*. Flux vectors at steady state are a standard tool for computing elementary modes [5], that correspond to the unique set of reactions in the network normal form that we obtain with the techniques of this paper. Our simplification method, unlike the elementary modes approach, deals with the impact of the simplification on kinetic rates as well as the network structure.

Given an ordered set of m reactions $\mathcal{R} = \{r_1, \ldots, r_m\}$ called *reaction basis*, a flux is a pair $w = (v; e)$ of a *flux vector* $v \in \mathbb{R}^m$ and an expression $e \in \mathit{Expr}$. The

$$\frac{e = \sum_{\{w \in W \mid X \in Prod_{\mathcal{R}}(w)\}} kin(w) \qquad e' = \sum_{\{w \in W \mid X \in Cons_{\mathcal{R}}(w)\}} kin(w) \qquad e' = x_X e''}{\exists X.\ W \Rrightarrow^{\text{Inter}}_{\mathcal{R}} \bigwedge_{\{w,w' \in W \mid X \in Prod_{\mathcal{R}}(w) \cap Cons_{\mathcal{R}}(w')\}} w \diamond_{e'} w' \quad \bigwedge_{\{w \in W \mid X \notin Prod_{\mathcal{R}}(w) \cup Cons_{\mathcal{R}}(w)\}} w[x_X := \frac{e}{e''}] \quad \wedge C(W)[x_X := \frac{e}{e''}]} \text{ (Intermediate)}$$

$$\frac{}{W \bigwedge_{1 \leq i \leq k}(v_i, e_i) \wedge (\sum_{1 \leq i \leq k} n_i v_i, e) \ \Rrightarrow^{\text{Dep}}_{\mathcal{R}} \ W \bigwedge_{1 \leq i \leq k}(v_i, e_i + n_i e)} \text{ (Dependent)}$$

Fig. 8. Simplification rules of flux networks.

function $react_{\mathcal{R}}$ maps fluxes to reactions w.r.t. a reaction basis $\mathcal{R}$ as follows:

$$react_{\mathcal{R}}(v; e) = (\sum_{1 \leq i \leq m} v_i s_i \rightarrow \sum_{1 \leq i \leq m} v_i s'_i;\ e).$$

Consequently, the i-th vector u_i of the standard basis is mapped to the i-th reaction r_i of the reaction basis $\mathcal{R}$. Now, instead of simplifying reaction networks, we directly simplify *flux networks* W defined as reaction networks but with fluxes in place of reactions:

$$W ::= w \mid cst(e) \mid W \wedge W' \mid \exists X.\ W.$$

We lift $react_{\mathcal{R}}$ to map flux networks to reaction networks as follows:

$$react_{\mathcal{R}}(cst(e)) = cst(e), \quad react_{\mathcal{R}}(W \wedge W') = react_{\mathcal{R}}(W) \wedge react_{\mathcal{R}}(W'),$$
$$react_{\mathcal{R}}(\exists X.\ W) = \exists X.\ react_{\mathcal{R}}(W).$$

We denote $FNets_{\mathcal{R}}$ the set of flux networks W such that $react_{\mathcal{R}}(W)$ is a linear reaction network for $\mathcal{U}$. The interpretation of $W \in FNets_{\mathcal{R}}$ in terms of system of equations is defined as $S_{\mathcal{R}}(W) = S(react_{\mathcal{R}}(W))$. Finally, we translate some previous definitions to the context of flux networks:

$$Prod_{\mathcal{R}}(w) = Prod(react_{\mathcal{R}}(w)), \qquad Cons_{\mathcal{R}}(w) = Cons(react_{\mathcal{R}}(w)),$$
$$kin(v; e) = e, \qquad (v; e) \diamond_{e''} (v'; e') = (v + v'; \tfrac{ee'}{e''}).$$

We then define two simplification rules for flux networks in Fig. 8. First, (Intermediate) is simply a reformulation of the one in Fig. 6 but in the terminology of flux networks. The new rule (Dependent) removes a *dependent flux*, that is one whose flux vector can be written as a positive linear combination of the flux vectors of some other fluxes. The rate of the removed flux is added to the rate of the fluxes that it depends on. This guarantees that the system of ordinary differential equations associated to the reaction network is unchanged:

Lemma 7. *Given $W \in FNets_{\mathcal{R}}$, if $W \Rrightarrow_{\mathcal{R}}^{Dep} W'$, then $S_{\mathcal{R}}(W) \sim S_{\mathcal{R}}(W')$.*

Two fluxes are *structurally similar*, denoted $(v, e) \sim^{struc} (v', e')$, if they have the same flux vector, that is $v = v'$. Two vector networks are *structurally similar*, denoted $W \sim^{struc} W'$ if they have structurally similar fluxes.

We can now state the Theorem on the structural confluence for this simplification system. We denote by $\Rrightarrow_{\mathcal{R}}= (\Rrightarrow_{\mathcal{R}}^{\text{Inter}} \cup \Rrightarrow_{\mathcal{R}}^{\text{Dep}})$ the simplification of vector networks with the rules of Fig. 8.

Theorem 2. *The relation $\Rrightarrow_{\mathcal{R}}$ on $(FNets_{\mathcal{R}}, \sim^{struc})$ is confluent.*

Proof (Scetch). The outline of the proof is as follows:

1. the simplification relation $\Rrightarrow_{\mathcal{R}}$ preserves the set of intermediate species,
2. the local confluence holds for $\Rrightarrow_{\mathcal{R}}$,
3. the binary relation is terminating, so by Newman's lemma, it is confluent.

Note that adding a rule that eliminates reactions whose reaction vectors can be written as sums of the reaction vectors of other reactions in the same network (instead of using a reaction basis) does not guarantee the confluence for the network structure.

Example 5. In Example 4, the elimination of the intermediates Y and Z in two different orders was shown to generate two different networks N_{XZY} and N_{XYZ} the latter having the additional reaction r_{1356}. Let us take $\{r_1, \ldots, r_6\}$ as a reaction basis. If we translate the simplifications to flux networks, the flux vector associated to reaction r_{ij} is $\mathsf{u}_i + \mathsf{u}_j$. Also, the flux vector associated to r_{1356} is $\mathsf{u}_1 + \mathsf{u}_3 + \mathsf{u}_5 + \mathsf{u}_6$, that is the sum of the flux vectors of r_{13} and r_{56}. Thus, the application of the (DEPENDENT) rule to the flux associated to r_{1356} results in a flux network W such that $react_{\mathcal{R}}(W) = N'_{XYZ}$. Since r_{1356} is eliminated, the networks N_{XZY} and N'_{XYZ} have the same structure. The rate of reaction r_{13} in N'_{XYZ} is given by the rate of r_{13} in N_{XYZ}, plus the rate of r_{1356} in N_{XYZ}, and is therefore equal to $\frac{k_1 k_3}{K_1} x_{\mathrm{A}} + \frac{k_1 k_5}{K_1} x_{\mathrm{A}} \sim k_1 x_{\mathrm{A}}$, that is the rate of r_{13} in N_{XZY}. Similarly, one can show that the rates of r_{56} in the two networks also coincide, and both networks have the same kinetics.

The following variation on the same example shows that confluence of the kinetics is not in general guaranteed.

Example 6. Now we shall examine again the simplifications performed in Example 4, but this time we look at the reaction networks as simplifications of the larger network N_ϵ in Fig. 9 from which N_X results after elimination of X. The reaction basis is now $\mathcal{R}' = \{r_{1'}, r_{2'}, r_3, r_{4'}, r_{5'}, r_6\}$ and the reaction r_1 in N_X is obtained from N_ϵ by merging $r_{1'}$ and $r_{2'}$ (that, following our convention, we denote $r_{1'2'}$) and is thus associated to the flux $(\mathsf{u}_1 + \mathsf{u}_2, k_1 x_{\mathrm{A}})$ w.r.t. $\mathcal{R}'$. Similarly, $r_2 = r_{1'4'}$ is associated to $(\mathsf{u}_1 + \mathsf{u}_4, k_2 x_{\mathrm{A}})$, $r_4 = r_{2'5'}$ to $(\mathsf{u}_2 + \mathsf{u}_5, k_4 x_{\mathrm{Y}})$, and $r_5 = r_{4'5'}$ to $(\mathsf{u}_4 + \mathsf{u}_5, k_5 x_{\mathrm{Y}})$.

The eliminations of Z first and Y after, represented in Fig. 7, generate the reactions r_{26}, r_{56}, r_{13} and r_{236} (with flux vectors respectively $\mathsf{u}_1 + \mathsf{u}_4 + \mathsf{u}_6$,

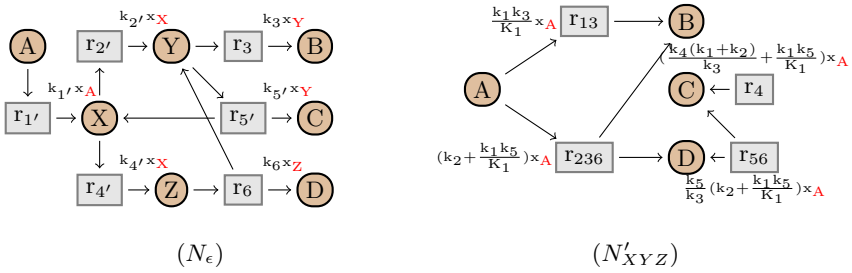

Fig. 9. Initial network (N_ϵ) and network (N'_{XYZ}) obtained after elimination of X, Y, dependent rule r_{15} and then Z. We have $K_1 = k_3 + k_5$.

$\mathsf{u}_4 + \mathsf{u}_5 + \mathsf{u}_6$, $\mathsf{u}_1 + \mathsf{u}_2 + \mathsf{u}_3$ and $\mathsf{u}_1 + \mathsf{u}_3 + \mathsf{u}_4 + \mathsf{u}_6$), with no dependent reactions. Consider now the elimination of Y from N_X. Reaction r_{15} has flux $(\mathsf{u}_1 + \mathsf{u}_2 + \mathsf{u}_4 + \mathsf{u}_5, \frac{k_1 k_5}{K_1} x_A)$ in network N_{XY} and is dependent on reactions r_2 and r_4. If we choose to eliminate reaction r_{15} using the (DEPENDENT) rule and apply the (INTERMEDIATE) rule on Z we obtain the network N'_{XYZ} in Fig. 9. No further simplification rule can be applied. Notice that this network is structurally the same as network N_{XZY} in Fig. 7, but all reactions have different kinetic rates.

7 Normalization modulo kinetic rates

We now present the principal result of this paper, about confluence of the simplification system modulo the kinetic rates. In other words, whatever the order of simplification, we can always obtain a fully simplified network with the same structure and with similar system of equations, but the kinetic rates associated to the fluxes can be different, as illustrated before in the example 6.

Given a fixed set of intermediate species $\mathcal{U}$ and an initial reaction basis $\mathcal{R}$, two networks are *similar*, denoted $W \sim_{\mathcal{R}} W'$, if they are structurally similar ($W \sim^{struc} W'$), and their systems of equations are similar ($S_{\mathcal{R}}(W) \sim S_{\mathcal{R}}(W')$).

Theorem 3. *The relation* $\Rrightarrow_{\mathcal{R}}$ *on* $(FNets_{\mathcal{R}}, \sim_{\mathcal{R}})$ *is confluent.*

8 An example from the BioModels database

We have shown that the simplification system that we presented can exhibit non-confluence of the rates, even in a simple scenario with a small number of intermediates. To find if such a situation occurs in practice, we investigated the SBML models in the curated BioModels database [8]. For each mass-action model, we created the graph of complexes and searched it for cycles of intermediates, to identify possible candidates for non-confluence. Then, with an implementation

of the simplification rules, we considered the elimination of triples or quadruples of intermediates in different orders, and compared the resulting networks.

We were thus able to identify two different reduced networks for model BIOMD0000000173. This is a model of the Smad-based signal transduction mechanisms from the cell membrane to the nucleus, presented in [21]. A subnetwork of this model is represented in Fig. 10. It includes all reactions involving cytoplasmic and nuclear Smad4 and Smad2/Smad4 complexes (abbreviated $S4_c$, $S4_n$, $S24_c$ and $S24_n$): shuttling of Smad4, formation of Smad2/Smad4 complex, import of Smad2/Smad4 into the nucleus, and formation of EGFP-Smad2/Smad4 complex. This network is linear for the four intermediate species $S4_c$, $S4_n$, $S24_c$, $S24_n$. The different orders of elimination yield simplified networks with the same structure but different kinetics. This confirms that the or of simplifying a biological network may indeed affect the result.

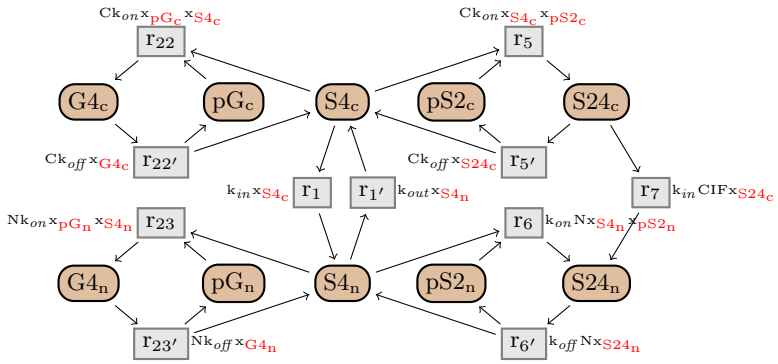

Fig. 10. Subnetwork of the Smad signal transduction network in [21].

9 Related work

In this section, we compare our work to two other simplification methods.

Radulescu et. al. [17] propose a simplification method for eliminate intermediate species at steady-state, while preserving the ODEs semantics. Compared to the method presented in this paper, their simplification removes may intermediate species in a single big step. Rather than imposing restrictions such as linearity with respect to intermediates, they rely on approximations for the kinetic rates in the general case. In a first step, they simplify the graph structure by using the elementary modes [5]. Then, they resolve approximately the steady-state equations after the concentration of the intermediates, and therefore obtain a simplified ODEs system. They finally assign the kinetic rates to the reactions, to obtain a reduced network corresponding to the simplified ODEs system.

The computation of the elementary modes in their method produces an unique network structure. Since our elimination of the dependent reactions is

based on similar methods, the network structures obtained with our simplification or with their method are the same. Since they use approximations, the kinetic rates are however not the same. With an exact simplification, we could expect to find the same underlying ODEs system. And, similarly to our simplification, the assignment of the rates to the reactions is not unique with their simplification.

Note that in [18], they proposed an additional step to their method, after computing the simplified graph structure, that can remove reactions if their reaction vectors are dependent, instead of the flux vectors. After this step, the assignment of the kinetic rates can be done in an unique manner. However, this elimination based on reaction vectors is not confluent, even for the graph structure, and so this method will still not be confluent.

Saez et. al. [20] improved their simplification procedure from [19], while producing some overlapping results but independently to the best of our knowledge. As before, their producere removes many intermediate species in steady state in one big step. Their conditions for the simplification are similar to us (linearity, stoichiometry, etc.). If applied to a single intermediate species, their method actually gives exactly the same result that the rule (INTERMEDIATE). However, their method applied in one step on a set of intermediates is not similar to applying it sequentially on one intermediate. It seems that their one-step method directly removes the dependent reactions. Since they obtain an unique simplified reaction network, with an unique assignment of the rates, it would be interesting to understand in more detail how they distinguish their unique result from the many others that are possible with our method.

Conclusion

We have shown that the elimination of linear intermediate species is not confluent in general. We provided a new simplification rule to remove dependent reactions, and proved that the extended rewrite system is confluent up to kinetic rates, that is, all normal forms of the same network will have the same structure and similar systems of equations, but can have different kinetic rates. Future research efforts is needed to characterize networks that possess a unique normal form.

References

1. L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, July 2006.
2. C. Chaouiya. Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4):210–219, 2007.
3. F. Fages, S. Gay, and S. Soliman. Inferring reaction systems from ordinary differential equations. *Theoretical Computer Science*, 599:64–78, 2015.
4. M. Feinberg. Chemical reaction network structure and the stability of complex isothermal reactors—i. the deficiency zero and deficiency one theorems. *Chemical Engineering Science*, 42(10):2229 – 2268, 1987.

5. J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC bioinformatics*, 5(1):175, 2004.
6. M. Hucka, et. al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.
7. G. P. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.
8. N. Juty et. al. BioModels: Content, Features, Functionality and Use. *CPT: Pharmacometrics & Systems Pharmacology*, 2015.
9. E. L. King and C. Altman. A schematic method of deriving the rate laws for enzyme-catalyzed reactions. *Journal of physical chemistry*, 60(10):1375–1378, 1956.
10. C. Kuo-Chen and S. Forsen. Graphical rules of steady-state reaction systems. *Canadian Journal of Chemistry*, 59(4):737–755, 1981.
11. C. Kuttler, C. Lhoussaine, and M. Nebut. Rule-based modeling of transcriptional attenuation at the tryptophan operon. *Transactions on Computational Systems Biology*, XII:199–228, 2010.
12. G. Madelaine, C. Lhoussaine, and J. Niehren. Attractor equivalence: An observational semantics for reaction networks. In *Formal Methods in Macro-Biology*, pages 82–101. Springer, 2014.
13. G. Madelaine, C. Lhoussaine, J. Niehren, E. Tonello. Structural simplification of chemical reaction networks in partial steady states. *Journal extension of CMSB'15.*
14. U. Mäder, A. G. Schmeisky, L. A. Flórez, and J. Stülke. Subtiwiki—a comprehensive community resource for the model organism bacillus subtilis. *Nucleic acids research*, 2011.
15. J. Niehren. Uniform confluence in concurrent computation. *Journal of Functional Programming*, 10(5):453–499, Sept. 2000.
16. J. Niehren, M. John, C. Versari, F. Coutte, and P. Jacques. Qualitative reasoning for reaction networks with partial kinetic information. In *CMSB*. 2015.
17. O. Radulescu, A. Gorban, A. Zinovyev and A. Lilienbaum. Robust simplifications of multiscale biochemical networks. *BMC systems biology*, 2(1):86, 2008.
18. O. Radulescu, A. N. Gorban, A. Zinovyev, V. Noel. Reduction of dynamical biochemical reactions networks in computational biology. *Frontiers in Genetics*, 2012.
19. M. Sáez, C. Wiuf, and E. Feliu. Graphical reduction of reaction networks by linear elimination of species. *arXiv preprint arXiv:1509.03153*, 2015.
20. M. Sáez, C. Wiuf, and E. Feliu. Graphical reduction of reaction networks by linear elimination of species. *Journal of Mathematical Biology*, pages 1–43, 2016.
21. B. Schmierer, A. L. Tournier, P. A. Bates, and C. S. Hill. Mathematical modeling identifies smad nucleocytoplasmic shuttling as a dynamic signal-interpreting system. *Proceedings of the National Academy of Sciences*, 105(18):6608–6613, 2008.
22. E. Tonello, M. R. Owen, and E. Farcot. On the elimination of intermediate species in chemical reaction networks. *In preparation*, 2016.