Minimization of Dynamical Systems over Monoids

Georgios Argyris
DTU Technical University of Denmark, Denmark

Alberto Lluch Lafuente
DTU Technical University of Denmark, Denmark

Mirco Tribastone
IMT School for Advanced Studies Lucca, Italy

Max Tschaikowski
Aalborg University, Denmark

Andrea Vandin
Sant'Anna School for Advanced Studies Pisa, Italy

Abstract

Quantitative notions of bisimulation are well-known tools for the minimization of dynamical models such as Markov chains and differential equations. In a forward-type bisimulation, each state in the quotient model represents an equivalence class and the dynamical evolution gives the overall sum of its members in the original model. Here we introduce generalized forward bisimulation (GFB) for dynamical systems over commutative monoids and develop a partition refinement algorithm to compute the largest one. When the monoid is $(\mathbb{R}, +)$, our framework recovers probabilistic bisimulation for Markov chains and more recent forward bisimulations for systems of nonlinear ordinary differential equations. When the monoid is $(\mathbb{R}, \cdot)$ we can obtain nonlinear model reductions for discrete-time dynamical systems and ordinary differential equations where each variable in the quotient model represents the product of original variables in the equivalence class. When the domain is a finite set such as the Booleans $\mathbb{B}$, we can apply GFB to Boolean networks, a widely used dynamical model in computational biology. Using a prototype implementation of our minimization algorithm for GFB, we find several disjunction- and conjuction-preserving reductions on 60 Boolean networks from two well-known model repositories.

2012 ACM Subject Classification CCS→Theory of Computation→Models of Computation→Timed and hybrid systems

Keywords and phrases Bisimulation, Markov chains, Dynamical systems, Regulatory networks

Acknowledgements Partially supported by the DFF project REDUCTO 9040-00224B, the Poul Due Jensen Foundation grant 883901, and the PRIN project SEDUCE 2017TWRCNB.

1 Introduction

Bisimulation is a fundamental tool in computer science for abstraction and minimization, relating models by useful logical and dynamical properties [40]. Originally developed to reason about concurrent processes in a non-quantitative setting [38], it has been extended to quantitative models based on labeled transition systems, such as, e.g., the notion of probabilistic bisimulation [24], closely related to ordinary lumpability for Markov chains [8].

Forward bisimulations relate states based on criteria that depend on their *outgoing* transitions (as opposed to *backward* bisimulations that depend on *incoming* transitions, e.g., [17]). When applied to a dynamical system (DS), forward bisimulations preserve properties related to sums of values of state variables. E.g, probabilistic bisimulation for Markov chains yields a quotient model where each state represents an equivalence class that preserves the sum of the probabilities of its members; forward bisimulation for reaction networks identifies equivalence classes among the chemical species that preserve the total concentration [11, 10]; forward differential equivalence (FDE) for nonlinear ordinary differential equations (ODEs) relates state variables and preserves sums of their solutions [12].

An attractive feature of bisimulation is that one can compute the largest bisimulation equivalence using partition refinement, based on the pioneering solution for concurrent processes [35]. This can make bisimulation an effective approach for the minimization of complex DS, adding to a wealth of cross-disciplinary methods originated in e.g., chemical engineering [34], control theory [1], and systems biology [43].

Thus far, one can identify two common properties of the various incarnations of forward bisimulation for DS. First, they preserve sums of state values; second, the DS variables take real $\mathbb{R}$ values. There are, however, reasons that call for extensions or generalizations of this setting. E.g., a forward bisimulation for ODEs can be seen as a special case of *linear lumping* [34], a minimization achieved by an appropriate linear projection of the state space operated by a matrix that encodes the partition of the state variables. However, one may be also interested in *nonlinear lumpings* where each state in the reduced model represents a nonlinear transformation of original variables [26]. Another motivating question tackled in this paper is the generalization of the domain on which the DS evolves. Forward bisimulation is not currently applicable to DS that evolve over finite domains. Consider, e.g., the DS

$$\begin{aligned} &x_1(k+1) = x_2(k) \vee x_3(k) \qquad\qquad && x_2(k+1) = x_1(k) \vee x_3(k) \\ &x_3(k+1) = \neg x_3(k) \wedge (x_1(k) \vee x_2(k)) && \end{aligned} \tag{1}$$

where the state variables x_1, x_2, and x_3 are defined over the Booleans $\mathbb{B} = \{0, 1\}$, and k denotes discrete time. This is an example of a Boolean network (BN), an established model of biological systems [22], sometimes introduced as a discrete-time qualitative analogue of differential equations [44], where quantitative species concentrations are abstracted into qualitative *activation values* of the variables (e.g., 0/1 for inactive/active components).

Here we develop a more abstract notion of forward bisimulation, *generalized forward bisimulation* (GFB), for a DS over a (commutative) monoid. We show that this is a conservative extension with respect to the literature because we recover available notions of forward bisimulation for DS when the monoid is $(\mathbb{R}, +)$. However, it is more general. E.g., over the monoid $(\mathbb{B}, \vee)$ one can prove that variables x_1 and x_2 in (1) are *GFB equivalent*, i.e., we can rewrite the model in terms of $x_1 \vee x_2$ and x_3. Indeed, by computing the disjunction of the left- and right-hand-side of x_1 and x_2 in (1) we get

$$x_1(k+1) \vee x_2(k+1) \quad = \quad x_2(k) \vee x_3(k) \vee x_1(k) \vee x_3(k) \quad = \quad x_3(k) \vee (x_1(k) \vee x_2(k))$$

By introducing the derived variable $x_{1,2} \equiv x_1 \vee x_2$, we get the *GFB-reduced model*:

$$x_{1,2}(k+1) = x_3(k) \vee x_{1,2}(k) \qquad\qquad x_3(k+1) = \neg x_3(k) \wedge x_{1,2}(k) \tag{2}$$

This can be used in place of the original model if one is not interested in the individual values of x_1 and x_2, but only in their disjunction.

In this paper we show that GFB satisfies two desirable properties for bisimulation relations.

i) Over any commutative monoid $(\mathbb{M}, \oplus)$, GFB characterizes $\oplus$-preserving reductions, in the sense that any DS with fewer state variables which coincide with $\oplus$-operations of original state variables must necessarily be the quotient of a GFB. This generalizes characterization results for Markov chains [24], chemical reaction networks [9], and nonlinear ODEs [12]. Notably, our characterization result also covers the asymptotic dynamics, often of interest when analyzing DS (see, e.g., [21]). We show that GFB preserves all *attractors*, i.e., the states towards which the DS tends to evolve and remain.

ii) GFB can be computed by a partition refinement algorithm. We develop a *template* algorithm which hinges on the computation of a formula whose decidability and complexity depend on the domain and the right-hand sides of the dynamical system under study. In general, this can be undecidable. However, when the monoid is $(\mathbb{R}, +)$ our algorithm reduces to that for forward differential equivalence for nonlinear ODEs [12]. Instead, when the domain is $\mathbb{B}$, the problem corresponds to Boolean satisfiability.

iii) For polynomial ODEs and the monoid $(\mathbb{R}, \cdot)$, we obtain, to the best of our knowledge, the first algorithm for nonlinear model reduction in (randomized) polynomial time.

On the other hand, while previous results are essentially agnostic to whether the time evolution of the DS is continuous or discrete, in the more general setting of GFB more care has to be taken. More specifically, it is well understood that the criteria for probabilistic bisimulation [24] are the same for both continuous-time and discrete-time Markov chains. Similarly, FDE equivalently applies to both a nonlinear ODE system in the form $\partial_t x = f(x)$ (where ∂_t denotes time derivative) and to a discrete-time nonlinear DS in the form $x(k+1) = f(x)$. For GFB, we show that this does not hold any longer. For this reason, we first develop GFB for discrete-time DS. Then, we consider continuous time by considering GFB for DS over the reals, thereby relating to, and extending, results for ODEs.

Applications. Using a prototype implementation, we apply GFB to case studies from different domains. We consider Boolean and multi-valued networks [22, 45], where the latter allows for finer degrees of activation than just inactive/active as in (1). These models are known to suffer from state-space explosion, making model reduction particularly appealing (see, e.g., [2]). We select two case studies from the literature to showcase the physical intelligibility of GFB reductions: we show how $(\mathbb{B},\wedge)$ allows to identify and abstract away from distinct *sub-models* (biological pathways); we show how finite monoids and operations min and max allow studying *full model (de)activation*, meaning that we obtain reductions that track groups of components whose activation status denote the (de)activation of different mechanisms of the model. We also perform a large-scale validation of GFB on 60 models from established repositories (GinSim [29], BioModelsDB [27]). We show that GFB is *useful* due to its high reduction power, and the high speed-up obtained in attractors computation.

2 Related work

Most of the literature about model minimization can be found for dynamical systems over the reals. In this context, the general framework of exact lumping considers model reductions by means of both linear and nonlinear operators [25, 46]. The aforementioned notions of bisimulation for Markov chains and FDE can be seen as specific linear reductions that are induced by a partition of the state space. Indeed, this corresponds to a specific type of minimization known as *proper lumping*, where each original variable is represented by only one variable in the reduced model [34]. Since also GFB is developed in the same style, it too can be seen as a special case of exact lumping. However, the largest GFB can be computed in randomized polynomial time when the dynamics is described by polynomials over the monoids $(\mathbb{R}, +)$ or $(\mathbb{R}, \cdot)$, see [14] and Section 5. Instead, the computation of exact lumpings hinges in the case of polynomial dynamics on symbolic computations which exhibit an exponential worst case complexity [26, Section 2.2].

$\mathcal{L}$-bisimulation [6, 7] can be seen as a generalization of backward differential equivalence (BDE) [12], a backward-type bisimulation for non-linear ODEs, and is thus complementary to FDE (hence, GFB), as discussed in [6, 7, 5]. It is also worth noting that neither BDE nor $\mathcal{L}$-bisimulation allow for model reduction through nonlinear transformations, in contrast to

GFB. Similarly to $\mathcal{L}$-bisimulation, consistent abstraction (aka bisimulation) [36, 37, 47] is complementary to GFB. Indeed, for a so-called observation function, the largest consistent abstraction gives rise to a minimal reduced dynamical system which coincides with the original one up to the chosen observation function. Instead, computing the largest GFB corresponds to the task of finding an observation function which induces a largest consistent abstraction. Hence, GFB reduces across observation functions, while consistent abstraction reduces with respect to a given observation function. Moreover, in contrast to consistent abstraction, GFB considers the subclass of observation functions induced by equivalence relations. To the best of our knowledge, the computation of an observation function yielding a minimal reduced model has been investigated for linear dynamics only [36].

Model-reduction approaches have been developed in the field of Boolean networks. Here, [3] proposes Boolean backward equivalence (BBE), a backward-type bisimulation in line with those for Markov chains [8] and BDE. Similarly to the notions for Markov chains and ODEs [12], it can be shown that BBE and GFB (when applied to BNs) are not comparable. Further approaches for BN reduction are based on the idea of *variable absorption* (see, e.g., [33, 48]) where selected variables are *absorbed* by the update functions of their target variables by replacing all occurrences of the absorbed variables with their update functions. These approaches are complementary to GFB because they do not compute exact reductions, meaning that they might introduce spurious behaviours.

3 Preliminaries

In this section we begin by formalizing the notion of dynamical system and notation considered in this paper. After that, we provide a running example used throughout the text.

▶ **Definition 1** (Dynamical System). *A discrete-time dynamical system (DS) is a pair* $D = (X, F)$ *where* $X = \{x_1, \ldots, x_n\}$ *is a set of variables and* $F = \{f_{x_1}, \ldots, f_{x_n}\}$ *is a set of update functions, where* $f_{x_i} : \mathbb{M}^X \to \mathbb{M}$ *is the update function of variable* x_i*. Elements of* $\mathbb{M}^X$ *are* states*. The solution (aka simulation) of* D *underlying the initial state* $s(0) \in \mathbb{M}^X$ *is given by the sequence* $(s(k))_{k\geq 0}$*, where* $s(k+1) = F(s(k))$ *for all* $k \geq 0$*.*

We use R to denote an equivalence relation over X, and $\mathcal{X}_R$ the induced partition. We often do not distinguish among an equivalence relation and its induced partition. If not explicitly mentioned, we assume that $\oplus : \mathbb{M} \times \mathbb{M} \to \mathbb{M}$ is such that $(\mathbb{M}, \oplus)$ is a commutative monoid with neutral element $0_\oplus$. Moreover, G^I denotes the set of all (total) functions from I to G and $f[a/b]$ is the term arising by replacing each occurrence of a by b in f.

To explain the main concepts, we utilize a published Boolean network from [4] that describes cell differentiation. Deeper biological interpretation of the model and its reduction presented in the paper will be given in Section 7.

▶ **Example 1.** *Consider the discrete-time DS* (X, F) *with Boolean variables*

$$X = \{\mathsf{SCR}, \mathsf{SHR}, \mathsf{JKD}, \mathsf{MGP}, \mathsf{WOX5}, \mathsf{CLEX}, \mathsf{PLT}, \mathsf{ARF}, \mathsf{AUXIAA}, \mathsf{AUXIN}\}$$

and Boolean update function $F : \mathbb{B}^X \to \mathbb{B}^X$ *with:*

$$f_{\mathsf{SCR}} = \mathsf{SHR} \wedge \mathsf{SCR} \wedge (\mathsf{JKD} \vee \neg\mathsf{MGP}) \qquad f_{\mathsf{CLEX}} = \mathsf{SHR} \wedge \mathsf{CLEX}$$

$$f_{\mathsf{SHR}} = \mathsf{SHR} \qquad f_{\mathsf{PLT}} = \mathsf{ARF}$$

$$f_{\mathsf{JKD}} = \mathsf{SHR} \wedge \mathsf{SCR} \qquad f_{\mathsf{ARF}} = \neg\mathsf{AUXIAA}$$

$$f_{\mathsf{MGP}} = \mathsf{SHR} \wedge \mathsf{SCR} \wedge \neg\mathsf{WOX5} \qquad f_{\mathsf{AUXIAA}} = \neg\mathsf{AUXIN}$$

$$f_{\mathsf{WOX5}} = \mathsf{ARF} \wedge \mathsf{SHR} \wedge \mathsf{SCR} \wedge \neg\mathsf{CLEX} \qquad f_{\mathsf{AUXIN}} = \mathsf{AUXIN}$$

Possible monoids for this DS are $(\mathbb{B}, \oplus)$ *with* $\oplus \in \{\wedge, \vee, XOR\}$ *with neutral elements* $1, 0$ *and* 0*, respectively.*

We next recall the notion of attractor for discrete-time DS [28].

▶ **Definition 2** (Attractor). *Let* $D = (X, F)$ *be a discrete-time DS. A non-empty set* $A \subseteq \mathbb{M}^X$ *is called attractor of* D *(with respect to some given topology of* $\mathbb{M}^X$*) whenever*

- A *is invariant under* F*, that is,* $F(A) \subseteq A$*;*
- *there is an open neighborhood* B *of* A *such that for any* $v \in B$ *there exists a* $\nu \geq 1$ *such that* $F^n(v) \in A$ *for all* $n \geq \nu$*.* B *is called a* basin of attraction *of* A*.*

▶ **Example 2.** *Let* $s = (0, 0, 0, 0, 0, 1, 1, 1, 0, 1) \in \mathbb{B}^X$ *denote a state of the DS from Example 1 where the variables* CLEX, PLT, ARF, AUXIN *are active, and all the others are inactive. By applying the update functions we get* $F(s) = s' = (0, 0, 0, 0, 0, 0, 1, 1, 0, 1) \in \mathbb{B}^X$*, where* PLT*,* ARF *and* AUXIN *are active, and all the others inactive. If we apply the update functions again, the system remains in the same state, i.e.,* $F(s') = s'$*, meaning that* $\{s'\}$ *is an attractor.*

4 Generalized Forward Bisimulation

In this section we define generalized forward bisimulation (GFB), the notion of GFB reduction, and show that GFB reductions preserve the original model dynamics.

▶ **Definition 3** (Generalized Forward Bisimulation). *Let* $D = (X, F)$ *be a discrete-time DS,* $(\mathbb{M}, \oplus)$ *a commutative monoid and* $\mathcal{X}_R$ *a partition of* X*. Then,* $\mathcal{X}_R$ *is a GFB when the following formula holds true:*

$$\forall s, s' \in \mathbb{M}^X. \bigwedge_{C \in \mathcal{X}_R} \Big(\bigoplus_{x_i \in C} s_{x_i} = \bigoplus_{x_i \in C} s'_{x_i} \Big) \implies \bigwedge_{C \in \mathcal{X}_R} \Big(\bigoplus_{x_i \in C} f_{x_i}(s) = \bigoplus_{x_i \in C} f_{x_i}(s') \Big).$$

The homomorphism of R*, denoted by* $\psi_R : \mathbb{M}^X \to \mathbb{M}^{\mathcal{X}_R}$*, is given by*

$$\psi_R(s)_C = \bigoplus_{x_i \in C} s_{x_i}, \quad \text{for all } C \in \mathcal{X}_R.$$

▶ **Example 3.** *Using* $\oplus = \wedge$*, we can show that* $\mathcal{X}_R$ *is a GFB for our running example, where*

$$\mathcal{X}_R = \{C, \{\text{PLT}\}, \{\text{ARF}\}, \{\text{AUXIAA}\}, \{\text{AUXIN}\}\}, \text{ with } C = \{\text{SCR}, \text{SHR}, \text{JKD}, \text{MGP}, \text{WOX5}, \text{CLEX}\}$$

Essentially, this implies that the running example can be rewritten solely in terms of conjunctions over all variables in C*, and the remaining individual variables. To this end, we first note that for all* $x_i \notin C$ *we have that* f_{x_i} *is independent of any* $x_j \in C$*.* [1] *Moreover, we observe that the update functions of* WOX5 *and* CLEX *contain terms* ¬CLEX *and* CLEX*, respectively, therefore the conjunction of their update functions (and of all variables in* C*) can be simply rewritten as* 0 *since:* $\bigwedge_{x_i \in C} f_{x_i}(s) = s_{\text{CLEX}} \wedge \neg s_{\text{CLEX}} \wedge (\ldots) = 0$*.*

We next introduce the notion of the reduced DS with respect to a GFB R.

[1] However, the original system is not trivially decoupled in variables in C and variables not in C, because ARF appears in the update function of WOX5.

▶ **Definition 4** (Reduced DS). *The reduction of a discrete-time DS $D = (X, F)$ for an equivalence relation R, denoted by D/R, is the DS $(\mathcal{X}_R, F_R)$ with $F_R = (f_C)_{C \in \mathcal{X}_R}$ such that*

$$f_C = \bigoplus_{x_i \in C} f_{x_i}[x_k/0_\oplus : x_k \notin \hat{X}][x_{i_{C'}}/x_{C'} : C' \in \mathcal{X}_R],$$

where $x_{i_C} \in C$ is a representative of $C \in \mathcal{X}_R$ and $\hat{X} = \{x_{i_C} : C \in \mathcal{X}_R\}$ is the set of all representatives.

▶ **Example 4.** *We compute the reduced DS of our running example for the GFB $\mathcal{X}_R$ from Example 3. We choose* JKD *as representative of C, while the choice of the representative for the other (singleton) blocks is immediate. With this, we obtain*

$$\begin{aligned} f_C &= \bigwedge_{x_k \in C} f_{x_k}[x_k/1 : x_k \notin \hat{X}][x_{i_{C'}}/x_{C'} : C' \in \mathcal{X}_R] \\ &= 1 \wedge 1 \wedge \big(C \vee \neg 1\big) \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge \neg 1 \wedge \{\mathsf{ARF}\} \wedge 1 \wedge 1 \wedge \neg 1 \wedge 1 \wedge 1 = 0 \end{aligned}$$

For all other blocks, instead, we obtain

$$f_{\{\mathsf{PLT}\}} = \{\mathsf{ARF}\}, \qquad f_{\{\mathsf{ARF}\}} = \neg\{\mathsf{AUXIAA}\}, \qquad f_{\{\mathsf{AUXIAA}\}} = \neg\{\mathsf{AUXIN}\}, \qquad f_{\{\mathsf{AUXIN}\}} = \{\mathsf{AUXIN}\}$$

▶ Remark 5. We note that, syntactically, the reduced DS depends on the choice of representatives. However, if R is a GFB, then Theorem 6 guarantees that such choice does not affect the *semantics* of the reduced DS.

We next prove that D and D/R share the same dynamics up to ψ_R iff R is a GFB.[2]

▶ **Theorem 6** (GFB characterization via model dynamics). *Fix a DS $D = (X, F)$, some partition $\mathcal{X}_R$ of X and let $D/R = (\mathcal{X}_R, F_R)$, for an arbitrary but fixed choice of representatives. Then, if $(\mathbb{M}, \oplus)$ is a commutative monoid, R is a GFB if and only if for any initial state $s_0 \in \mathbb{M}^X$, the solutions of D and D/R for s_0 and $\hat{s}_0 = \psi_R(s_0)$, respectively, are equal up to ψ_R, i.e.:*

$$\hat{s}_k = \psi_R(s_k), \quad \text{for } k \geq 0, \text{where } s_{k+1} = F(s_k) \text{ and } \hat{s}_{k+1} = F_R(\hat{s}_k).$$

Theorem 6 readily implies the following result on attractors.

▶ **Corollary 7.** *Let $D = (X, F)$ be a DS, $(\mathbb{M}, \oplus)$ a commutative monoid, R a GFB and $D/R = (\mathcal{X}_R, F_R)$ the underlying reduced DS. Then, we have the following two (equivalent) statements.*

- *If $A \subseteq \mathbb{M}^X$ is an attractor of D, then $\psi_R(A) \subseteq \mathbb{M}^{\mathcal{X}_R}$ is an attractor of D/R.*
- *If $A \subseteq \mathbb{M}^{\mathcal{X}_R}$ is not an attractor of D/R, then $\psi_R^{-1}(A) \subseteq \mathbb{M}^X$ is not an attractor of D.*

▶ **Example 5.** *We consider the attractor $A = \{(0, 0, 0, 0, 0, 0, 1, 1, 0, 1)\}$ of the running example from Example 2. The homomorphism ψ_R maps the attractor to $\psi_R(A) = \big\{(0, 1, 1, 0, 1)\big\}$. The Corollary 7 ensures that the set $\psi_R(A)$ is an attractor of the reduced system D/R. Indeed, by applying the update functions F_R to the state $(0, 1, 1, 0, 1)$, the reduced system D/R remains at the same state, and thus $\psi_R(A)$ is invariant under F_R.*

[2] Note: all proofs are given in the appendix.

5 Computation of the largest GFB

The computation of the largest (alternatively, coarsest) GFB that refines some given initial partition is based on the classic partition refinement algorithm [35] where the blocks of an initial partition are iteratively refined (or split) until a GFB is obtained. The largest GFB is obtained when the initial partition contains a singleton block; the freedom in choosing an arbitrary initial partition can be useful in applications to tune the reductions to preserve variables of interest, as discussed in Section 7. Here we prove that there exists a unique largest GFB that refines a given initial partition, and that the algorithm computes it.

▶ **Theorem 8.** *Let $D = (X, F)$ be a discrete-time DS, R an equivalence relation over X, and $\mathcal{X}_R$ the induced partition. There exists a unique coarsest GFB $\mathcal{H}$ that refines $\mathcal{X}_R$.*

A partition refinement algorithm for computing GFB needs a condition allowing us to tell: (i) if the current partition is a GFB, and, if not, (ii) how to split its blocks towards getting a GFB. Definition 3 can only be used for Point (i). Theorem 9 below provides a binary, relation-driven, characterization of GFB which allows for Point (ii). The intuition is that, by applying such binary characterization pairwise to all variables in each block of the current partition, we get the sub-blocks in which they should be split in the next iteration.

▶ **Theorem 9** (Binary Characterization of GFB)**.** *Let $D = (X, F)$ be a DS, $(\mathbb{M}, \oplus)$ a commutative monoid, R an equivalence relation on X, and $\mathcal{X}_R$ the induced partition. Then, $\mathcal{X}_R$ is a GFB if and only if for any $(x_i, x_j) \in R$ with $x_i \neq x_j$, the following formula holds:*

$$\Psi^{\mathcal{X}_R}_{x_i,x_j} \equiv \bigwedge_{C \in \mathcal{X}_R} \Big(\bigoplus_{x_k \in C} f_{x_k} = \bigoplus_{x_k \in C} f_{x_k}[x_i/0_\oplus][x_j/(x_i \oplus x_j)] \Big),$$

where $0_\oplus$ is the neutral element of $\oplus$.

Intuitively, the binary characterization tells us that we can rewrite an $\oplus$-expression of the update functions of a block of a GFB partition in terms of $\oplus$-expressions of pairs of GFB equivalent variables x_i and x_j. This can be done by successively moving, pair by pair, all variables of a GFB equivalence class to a chosen representative.

We now exemplify $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ on our running example.

▶ **Example 6.** *Let us consider again the GFB $\mathcal{X}_R$ from Example 3, the only non-singleton block $C \in \mathcal{X}_R$, and the two variables $\mathsf{SHR}, \mathsf{JKD} \in C$. With $\oplus = \wedge$ and $0_\wedge = 1$, we obtain*

$$\begin{aligned}\bigwedge_{x_k \in C} f_{x_k} &= \mathsf{SHR} \wedge \mathsf{SCR} \wedge (\mathsf{JKD} \vee \neg\mathsf{MGP}) \wedge \mathsf{SHR} \wedge \mathsf{SHR} \wedge \mathsf{SCR} \wedge \mathsf{SHR} \wedge \mathsf{SCR} \wedge \neg\mathsf{WOX5} \wedge \mathsf{ARF} \wedge \mathsf{SHR} \\ &\quad \wedge \mathsf{SCR} \wedge \neg\mathsf{CLEX} \wedge \mathsf{SHR} \wedge \mathsf{CLEX} \\ &= 0 \\ &= 1 \wedge \mathsf{SCR} \wedge ((\mathsf{JKD} \wedge \mathsf{SHR}) \vee \neg\mathsf{MGP}) \wedge 1 \wedge 1 \wedge \mathsf{SCR} \wedge 1 \wedge \mathsf{SCR} \wedge \neg\mathsf{WOX5} \wedge \mathsf{ARF} \wedge 1 \\ &\quad \wedge \mathsf{SCR} \wedge \neg\mathsf{CLEX} \wedge 1 \wedge \mathsf{CLEX} \\ &= \bigwedge_{x_k \in C} f_{x_k}[\mathsf{SHR}/1, \mathsf{JKD}/(\mathsf{SHR} \wedge \mathsf{JKD})]\end{aligned}$$

For any $C' \in \mathcal{X}_R$ with $C' \neq C$, the clause is trivially true because SHR and JKD appear only in the update functions of variables in C. Therefore, $\Psi^{\mathcal{X}_R}_{\mathsf{SHR},\mathsf{JKD}}$ is valid. In a similar way, one can show that $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid for all $(x_i, x_j) \in R$ with $x_i \neq x_j$, confirming that $\mathcal{X}_R$ is a GFB.

The next result addresses the algorithmic computation of the largest GFB.

Algorithm 1 Compute the largest GFB that refines an initial partition $\mathcal{X}_R$ for DS (X, F).

```
while true do
    $\mathcal{H}' \leftarrow \emptyset$
    for all $H \in \mathcal{H}$ do
        $R \leftarrow \{(x_i, x_j) \in H \times H$ : if $x_i \neq x_j$,
          then $\Psi^{\mathcal{H}}_{x_i,x_j}$ and $\Psi^{\mathcal{H}}_{x_j,x_i}\}$
        $\mathcal{H}' \leftarrow \mathcal{H}' \cup (H/R)$
    end for
    if $\mathcal{H} = \mathcal{H}'$ then
        return $\mathcal{H}$
    else
        $\mathcal{H} \leftarrow \mathcal{H}'$
    end if
end while
```

▶ **Theorem 10.** *Let $D = (X, F)$ be a discrete-time DS and X_R a partition. Algorithm 1 computes the largest GFB refining R by deciding at most $\mathcal{O}(|X|^3)$ instances of formula $\Psi^{\mathcal{H}}_{x_i,x_j}$. If $\mathbb{M}$ is finite, any formula $\Psi^{\mathcal{H}}_{x_i,x_j}$ is decidable.*

The decidability of $\Psi^{\mathcal{H}}_{x_i,x_j}$ in case where $\mathbb{M}$ is infinite is less immediate. Indeed, since deciding $\Psi^{\mathcal{H}}_{x_i,x_j}$ amounts to deciding identities between functions, decidability over infinite domains critically hinge on the nature of the update functions. For instance, if $\mathbb{M} = \mathbb{R}$, the conditions of $\Psi^{\mathcal{H}}_{x_i,x_j}$ require one to decide the equivalence of real-valued functions. If $\oplus = +$ and update function terms arise through addition and multiplication of variables and may contain minima and maxima expressions, the problem is double exponential [12]. If also exponential and trigonometric functions are allowed, the problem becomes undecidable [39].

We thus study the complexity of deciding $\Psi^{\mathcal{H}}_{x_i,x_j}$ when $(f_{x_i})_{x_i \in X}$ are polynomials and $\oplus \in \{+, \cdot\}$. In such a case, checking $\Psi^{\mathcal{H}}_{x_i,x_j}$ amounts to deciding whether the polynomials

$$\bigoplus_{x_k \in C} f_{x_k} \quad \text{and} \quad \bigoplus_{x_k \in C} f_{x_k}[x_i/0_{\oplus}][x_j/(x_i \oplus x_j)]$$

are equal. In case of the real and complex field, this question is equivalent to polynomial identity testing for which no holistic algorithms with polynomial time complexity are known [41].[3] Fortunately, the following result readily follows from the Schwartz-Zippel lemma [41].

▶ **Theorem 11.** *Let $D = (X, F)$ be a discrete-time DS and $\mathcal{X}_R$ a partition. Then, if $(f_{x_i})_{x_i \in X}$ are polynomials over some (sufficiently large) field $\mathbb{M}$ and $\oplus \in \{+, \cdot\}$, Algorithm 1 runs in randomized polynomial time. More specifically, assume that $\Psi^{\mathcal{H}}_{x_i,x_j}$ is false and that it involves polynomials of degree less or equal d. Then, for any finite set $S \subseteq \mathbb{M}$, any $C \in \mathcal{H}$ and a uniformly sampled $v \in S^X$, we have that*

$$\mathbb{P}\Big\{ \bigoplus_{x_k \in C} f_{x_k}(v) = \bigoplus_{x_k \in C} f_{x_k}[x_i/0_{\oplus}][x_j/(x_i \oplus x_j)](v) \Big\} \leq \frac{d}{|S|},$$

where $\mathbb{P}\{A\}$ denotes the probability of event A. In particular, one obtains a polynomial time randomized algorithm whenever $\mathbb{M}$ has more than d elements.

6 Continuous-time DS

We hereby relate GFB to continuous-time DS, showing how GFB encapsulates previous notions of bisimulations for (nonlinear) systems of ODEs. Thus, in what follows we will

[3] The common holistic approach rewrites a polynomial into a sum of monomials. Hence, if $\oplus = \cdot$ and all f_{x_k} have, say, 2 monomials, a direct computation of the monomials of $\oplus_{x_k \in C} f_{x_k}$ requires $\mathcal{O}(2^{|C|})$ steps.

consider $\mathbb{R}$ as the domain over which the DS evolves. In this case, one can study minimizations for an ODE system $\partial_t v(t) = \Phi(v(t))$ (where ∂_t denotes time derivative) using GFB on its time discretization (X, F), where $F(s) = s + \tau\Phi(s)$. Standard results imply that the approximation error between the ODE system and its time discretization vanishes if τ approaches zero [20].

Exact lumpability. GFB-type reductions can be captured by exact lumpability, a well-established notion of model minimization for ODEs [25, 46]. This is because an exact lumping must not be necessarily induced by a partition of the state variables. However, we will show that when an exact lumping on an ODE system is described by the homomorphism ψ_R of an equivalence relation R, then it must necessarily be a GFB for its discretization.

We start with the definition of exact lumping [25].

▶ **Definition 12.** *Given an ODE system $\partial_t v(t) = \Phi(v(t))$ with a differentiable function $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, a twice differentiable function $\psi : \mathbb{R}^X \to \mathbb{R}^{\hat{X}}$ is an exact lumping if $|\hat{X}| < |X|$ and there is a unique differentiable function $\hat{\Phi} : \mathbb{R}^{\hat{X}} \to \mathbb{R}^{\hat{X}}$ such that for any $v : [0;T] \to \mathbb{R}^X$ satisfying $\partial_t v(t) = \Phi(v(t))$, it holds that $\partial_t \psi(v(t)) = \hat{\Phi}(\psi(v(t)))$ for all $t \in [0;T]$.*

For example, consider the model given by $v_{x_1} = v_{x_1}$ and $v_{x_2} = v_{x_2}$. Then, $\psi(v_{x_1}, v_{x_2}) = v_{x_1}v_{x_2}$ is an exact lumping since $\partial_t\psi(v) = (\partial_{x_1}\psi(v)\partial_{x_2}\psi(v))\cdot\Phi(v) = (v_{x_2}, v_{x_1})\cdot(\partial_t v_{x_1}, \partial_t v_{x_2})^T = 2v_{x_1}v_{x_2} = 2\psi(v)$, where superscript T denotes the transpose of a vector. Now we can observe that this can be discovered using GFB on the time discretization of the ODE system, given by $f_{x_1}(s) = s_{x_1} + \tau s_{x_1}$, and $f_{x_2}(s) = s_{x_2} + \tau s_{x_2}$. Indeed $\mathcal{X}_R = \{\{x_1, x_2\}\}$ is a GFB over $(\mathbb{R}, \cdot)$ since $f_{x_1}\cdot f_{x_2} = (x_1+\tau x_1)\cdot(x_2+\tau x_2) = x_1x_2+2\tau x_1x_2+\tau^2 x_1x_2 = (f_{x_1}\cdot f_{x_2})[x_2/1, x_1/x_1x_2]$. This shows that ψ_R is indeed an exact lumping of. The next result formalizes this relationship.

▶ **Theorem 13.** *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable function $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, consider the DS $D = (X, F)$ with $F(s) = s + \tau\Phi(s)$ where $\tau > 0$. Further, let us assume that $\oplus : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is twice differentiable and that $(\mathbb{R}, \oplus)$ is a commutative monoid. Then, for any partition $\mathcal{X}_R$ of X:*

1) *If R is a GFB of D, then ψ_R is an exact lumpability of $\partial_t v(t) = \Phi(v(t))$.*

2) *If ψ_R is linear, then R is a GFB of D if and only if ψ_R is an exact lumpability of $\partial_t v(t) = \Phi(v(t))$.*

With the exception of the important special case where ψ_R is linear, Theorem 13 does not address the question whether GFB is also a necessary condition for exact lumpability. Indeed, it turns out that a characterization requires one to relax formula $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ so that, roughly speaking, it deliberately ignores the terms of (higher) order $\tau^2, \tau^3, \ldots$ and so on.

▶ **Lemma 14.** *Consider the continuous DS given by $\partial_t v_{x_1} = v_{x_1}\log(v_{x_2})$ and $\partial_t v_{x_2} = v_{x_2}\log(v_{x_1})$. Together with $v \oplus v' = \log(v) + \log(v')$ and $\mathcal{X}_R = \{\{x_1, x_2\}\}$, it then holds that $\psi_R(v_{x_1}, v_{x_2}) = \log(v_{x_1}) + \log(v_{x_2})$ is an exact lumping, while $\mathcal{X}_R$ is not a GFB.*

The next result characterizes exact lumpings of the form ψ_R and thus accounts for Lemma 14 and generalizes Theorem 13. As anticipated, we have to ignore higher-order terms $\mathcal{O}(\tau^2)$ when checking $\Psi^{\mathcal{X}_R}_{x_i,x_j}$, where $\mathcal{O}$ is the *big O* notation from numerical analysis.

▶ **Theorem 15.** *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable vector field $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, consider the DS (X, F) with $F(s) = s + \tau\Phi(s)$ where $\tau > 0$. Further, let us assume that $\oplus : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is twice differentiable and that $(\mathbb{R}, \oplus)$ constitutes a commutative monoid. Then, for any partition $\mathcal{X}_R$ of X, function ψ_R is an exact lumping if and only if for all $(x_i, x_j) \in R$ with $x_i \neq x_j$ formula $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid up to $\mathcal{O}(\tau^2)$, that is*

$$\bigwedge_{C \in \mathcal{X}_R} \Big(\bigoplus_{x_k \in C} f_{x_k} + \mathcal{O}(\tau^2) \;=\; \bigoplus_{x_k \in C} f_{x_k}[x_i/0_\oplus][x_j/(x_i \oplus x_j)] + \mathcal{O}(\tau^2) \Big). \quad (3)$$

We conclude this section by noting that if the update functions $(f_{x_i})_{x_i \in X}$ are polynomials, then (3) can be checked algorithmically by representing polynomials as sums of monomials and by dropping afterwards all monomials containing a term τ^ν with $\nu \geq 2$.

Forward differential equivalence and Markov chains. With the results of the previous subsection in place, we are now ready to relate GFB with related work on analogous bisimulation relations for dynamical systems. We start by restating the notion of forward differential equivalence of FDE from [12].

▶ **Definition 16** (FDE). *Given ODEs $\partial_t v(t) = \Phi(v(t))$ with a differentiable function $\Phi : \mathbb{R}^X \to \mathbb{R}^X$. A partition $\mathcal{X}_R$ of X is called FDE if ψ_R in case of $\oplus = +$ is an exact lumpability.*

The next result is a consequence of Theorem 13 showing that GFB encapsulates FDE [12].

▶ **Corollary 17.** *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable vector field $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, consider the DS $D = (X, F)$ with $F(s) = s + \tau\Phi(s)$ where $\tau > 0$. Then, for monoid $(\mathbb{R}, +)$, we have that R is a GFB of D if and only if R is an FDE of $\partial_t v(t) = \Phi(v(t))$.*

Similarly, the next corollary relates GFB with continuous-time Markov chains [8] and probabilisitic bisimulation of discrete-time Markov chains [24].

▶ **Corollary 18.** *Let (X, Q) be a continuous-time Markov chain with states X and transition rate matrix $Q \in \mathbb{R}^{X \times X}$. Consider the DS $D = (X, F)$ with $F(s) = s + \tau Q^T s$ where $\tau > 0$. Then, D is an embedded discrete-time Markov chain of (X, Q) for sufficiently small $\tau > 0$. With this, for monoid $(\mathbb{R}, +)$ the following three conditions are equivalent: 1. R is a GFB of D; 2. R is an ordinary lumpability of (X, Q); 3. R is a probabilistic bisimulation of D.*

▶ Remark 19. The above discussion ensures that $\Psi^{\mathcal{H}}_{x_i, x_j}$ from Algorithm 1 can be decided in polynomial time for forward differential equivalence and probabilistic bisimulation, see [12].

Attractors of continuous-time DS. The notion of attractor from Definition 2 also exists for continuous-time dynamics [23].

▶ **Definition 20** (Attractor). *Consider an ODE system $\partial_t v(t) = \Phi(v(t))$ with a differentiable vector field $\Phi : \mathbb{R}^X \to \mathbb{R}^X$. A compact nonempty set $A \subseteq \mathbb{R}^X$ is an attractor (aka asymptotically stable), if there exists an open neighborhood B of A such that for any $\varepsilon > 0$ there is some time $t' \geq 0$ such that for any $v[0] \in B$, the solution of $\partial_t v(t) = \Phi(v(t))$ with $v(0) = v[0]$ satisfies $d(v(t), A) \leq \varepsilon$ for all $t \geq t'$. Here, $d(v(t), A) = \min_{a \in A} d(v(t), a)$ and distance d is induced, similarly to B, by some norm.*

The next result from [23] essentially ensures that attractors of an ODE system can be approximated by attractors of its discrete-time discretization.

▶ **Theorem 21** ([23]). *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable vector field $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, let $A \subseteq \mathbb{R}^X$ be an attractor of $\partial_t v(t) = \Phi(v(t))$. Then, for any $\tau > 0$, there exists a set $A(\tau) \subseteq \mathbb{R}^X$ such that*

- *$F(A(\tau)) \subseteq A(\tau)$, where $F(s) = s + \tau\Phi(s)$ and;*
- *The sets $A(\tau)$ converge to the set A in the Hausdorff metric as $\tau \to 0$.*

Together with Corollary 7, Theorem 21 allows to use GFB to argue on attractors of ODE systems. Less importantly, we note that Theorem 21 does not explicitly provide basins of attraction for the sets $A(\tau)$. However, $A(\tau)$ are attractors when the discrete topology is used.

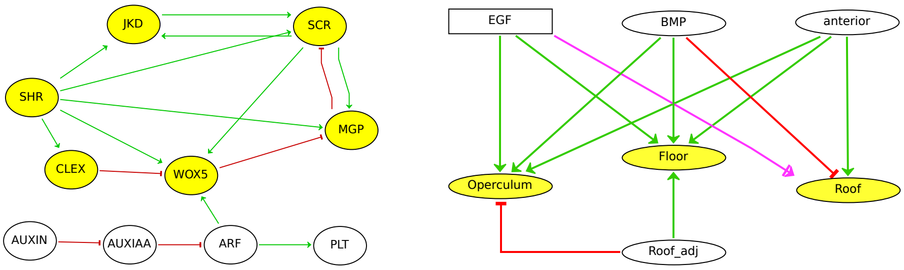

Figure 1 (Left) Pictorial representation of the Boolean network from Example 1 using GinSim [30], adapted from [4]. (Right) Pictorial representation using GinSim [30] adapted from [19] of the model on eggshell formation for drosophila melanogaster flies.

7 Applications

7.1 Regulatory Networks

We now apply GFB to Boolean and multi-valued networks from the literature.

Boolean network case study. We present in greater detail the BN from Example 1 used as running example. To ease the interpretation of the results, we will use the typical graphical notation of *influence graphs*, as shown in Figure 1 (left). Each variable is denoted as a node, while arrows denote *influences* among nodes coming from the update functions: green arrows denote a positive influence (*promotion*), while red arrows denote a negative one (*inhibition*). In Example 1, ARF promotes PLT due to the term ARF in f_{PLT}. , while AUXIN inhibits AUXIAA due to term $\neg$AUXIN in f_{AUXIAA}.

The BN consists of two connected pathways: one for the transcription factor SHR with its signalling to the other variables of the pathway (we highlight in yellow the involved nodes), and the other involves the hormone AUXIN and its signaling to the plethora (PLT) genes.

BN variables are commonly categorized into three groups [31]: *inputs* (SHR, and AUXIN) that do not have incoming edges, *outputs* (PLT) that do not have outgoing edges, and *internal nodes*, i.e., those that have both incoming and outgoing edges. The distinction is also obvious from update functions: input variables have a constant update function, while output variables do not appear in the update function of other variables. In particular, input variables are often set by the modeler to perform different *what if* experiments, whereas output ones are used to observe the response dynamics of the model. Here, each input *controls* its own pathway, meaning that the modeller can decide whether to enable them via appropriate initial states.

Considering the GFB $\mathcal{X}_R$ from Example 3 obtained for $\oplus = \wedge$, we can see that the only non-trivial block $C = \{\mathsf{SCR}, \mathsf{SHR}, \mathsf{JKD}, \mathsf{MGP}, \mathsf{WOX5}, \mathsf{CLEX}\}$ corresponds to the yellow nodes in Figure 1 (left). This GFB is computed by our algorithm for the initial partition that consists of two blocks separating outputs and non-output nodes.

Considering the reduced model for $\mathcal{X}_R$ from Example 6, all yellow nodes in Figure 1 (left) get collapsed into one, meaning that the SHR pathway is abstracted away. In other words, in this example GFB has automatically identified and *simplified* a pathway in the model, offering a coarser representation of the system focusing on one pathway only (the AUXIN one).

Multi-valued network case study. We now apply GFB to a multi-valued regulatory network from [19]. Intuitively, as we will see, a multi-valued network (MV) is a BN where variables might admit more than two values. This is a single-cell model describing the development of eggshell structures in drosophila melanogaster flies. The model has seven variables with relations depicted in Figure 1 (right) coming from the update functions:

$$
\begin{array}{lll}
f_{\mathsf{EGF}} = \mathsf{EGF} & f_{\mathsf{Roof}} = \mathsf{Ant}{:}1 \wedge \mathsf{EGF}{:}1 \wedge \mathsf{BMP}{:}0 & f_{\mathsf{RoofAdj}} = \mathsf{RoofAdj} \\
f_{\mathsf{BMP}} = \mathsf{BMP} & f_{\mathsf{Floor}} = \mathsf{Ant}{:}1 \wedge (\mathsf{EGF}{:}2 \vee (\mathsf{EGF}{:}1 \wedge \mathsf{BMP}{:}1)) \wedge \mathsf{RoofAdj}{:}1 & \\
f_{\mathsf{Ant}} = \mathsf{Ant} & f_{\mathsf{Operc}} = \mathsf{Ant}{:}1 \wedge (\mathsf{EGF}{:}2 \vee (\mathsf{EGF}{:}1 \wedge \mathsf{BMP}{:}1)) \wedge \mathsf{RoofAdj}{:}0 &
\end{array}
$$

Using the notation in [19], "var : v" stands for *variable* var *has value* v. This is a Boolean predicate evaluating to 1 if var has value v, and 0 otherwise. Variable EGF, denoted with a rectangular node in Figure 1 (right), can take values 0, 1, 2, denoting three different activation levels (absent, intermediate and high). All other variables are Boolean (0 or 1).[4]

Differently from Figure 1 (left), the variables divide in two groups only: the *inputs* EGF, BMP, Ant, and RoofAdj, and the *outputs* Operc, Floor, and Roof. Another difference is that we have a third edge type, the purple one from EGF to Roof. This is to visually stress that EGF influences Roof only when in intermediate level and not when in high level, respectively values 1 and 2.

The model relates three distinct follicle cell fates (the outputs), to combinations of values of the four inputs. EGF and BMP are known signaling pathways responsible for patterning of the drosophila eggshell [19]. This is encoded in the model in the sense that EGF and BMP influence, in different ways, all the three outputs. Finally, Ant, which stands for *anterior*, models the anterior competence region, therefore it is required by all outputs, while RoofAdj accounts for the state of neighboring cells by promoting Floor and inhibiting Operc (*operculum*).

Interestingly, the partition with one block for the three outputs, and singleton blocks for each other variable, is a GFB for $\oplus \in \{\max, \min\}$. Applying Definition 4 we get two different reduced models in the two cases, enabling complementary studies. Case $\oplus = \max$ allows to study cases of *full output deactivation*, meaning that the resulting reduced variable will get value 0 only if all output variables have value 0. Instead, case $\oplus = \min$ allows to study *full output activation*, as the corresponding reduced variable will have value 1 only when all output variables have value 1. By naming outputs the reduced variable corresponding to the block of outputs, after applying Definition 4 and some algebraic simplification we get

$$
f_{\mathsf{outputs}} = \mathsf{Ant}{:}1 \wedge (\mathsf{EGF}{:}1 \vee \mathsf{EGF}{:}2), \quad \text{for } \oplus = \max, \qquad f_{\mathsf{outputs}} = 0, \quad \text{for } \oplus = \min,
$$

while the update functions of the input variables remain unchanged. From this we get that: despite the three outputs have different dependencies on Ant, BMP, RoofAdj, and on different values of EGF, in the $\oplus = \max$ case it is enough to consider only ANT and EGF to answer questions related to full output deactivation. Furthermore, it is not necessary anymore to use three values for EGF, as we are only interested in the cases in which it is 0 or positive (EGF :1 ∨ EGF :2). Instead, from the $\oplus = \min$ case we know that the original model never expresses cases of full activation, i.e., it never happens that the three outputs contemporary have value 1. Indeed, by studying the update functions of the three original output variables, we can see that there are no values for the involved variables that makes all of them true.

Large-scale validation of GFB on regulatory networks. We present a large-scale validation of GFB on the BNs and MVs from the repositories GinSim (accessible at `http://ginsim.org/models_repository`) and BioModelsDB [27]. We validate GFB in terms of aggregation power and of speed-up offered for attractor analysis.

[4] We note that our framework requires all variables to have same domain $\mathbb{M}$. In order to support MV networks, we implicitly *expand* the domain of all variables to the largest one (e.g., $\{0, 1, 2\}$ of EGF). This does not change the models' dynamics, in the sense that when setting initial states fitting in the original domain will only encounter states fitting in the original domain.

Experimental setting. Experiments were performed using our prototype implementation of GFB in ERODE [13].[5] The formulas $\Psi^{\mathcal{H}}_{x_i,x_j}$ in Algorithm 1 were checked exploiting an integration of ERODE with the SMT solver Z3 [16]. We added to ERODE an importer for SBML Qual [15], an XML-based format supported by both repositories, allowing us to import all 43 BNs and 50 MVs. In order to automatically synthesize physically relevant initial partitions for each model, we implemented a check to infer candidate output variables, those which do not appear in the update function of any other variable. These variables were used to build *output-preserving* initial partitions (one block for the output variables, and one for the rest). This guarantees that the reduced models allow, e.g., for studies on full output (de)activation discussed before. In order to perform a consistent and fair treatment, we selected the 29 BNs and 31 MVs with at least one candidate output.

Validation of aggregation power. Figure 2 (left) provides the reduction ratios obtained for the BNs using $\oplus \in \{\wedge, \vee\}$. For each model we provide the reduction ratio as the number of reduced variables over that of original ones. For each operator $\oplus$, the ratios were sorted in ascending order. We can see that $\oplus = \wedge$ has high aggregation power, with about one third of the models having a reduction ratio below 0.6, while for $\oplus = \vee$ most of the models have 0.8 or more. In the $\oplus = \wedge$ case we can see that some models have particularly low ratios, below 0.2, some of which due to the fact that the reduced model has 2 species only. We do not consider these as *degenerate* reductions, because of the used initial partitions, as discussed. We do not present results on maximal reductions, e.g. those obtained with the trivial initial partition with one block only. These are significantly smaller, but some are degenerate with one species only. We leave for future work a more detailed study on finer intermediate reductions obtained for model-specific initial partitions preserving exactly the variables of interest for the modeler. Practically, modelers could decide to preserve only some outputs rather than all/none. Figure 2 (right) presents a similar study performed on the MV networks using $\oplus = \min$ and $\oplus = \max$ confirming the aggregation power of GFB.

Validation of analysis speed-up. Corollary 7 ensures that GFB maps all attractors of the original system to attractors of the reduced one. Here we show that this can speed-up attractor computation. We use the COLOMOTO Notebook [32], an environment incorporating a variety of tools for BN analysis. An example is BNS [18], which combines SAT-solving and bounded model checking to identify attractors. We computed the attractors of the 29 considered BNs and of their reductions.[6] Figure 2 (bottom) shows the obtained runtime ratios (computation time of attractors in the reduced model over that in the original one). In several cases the reduction led to significant analysis speed-ups: in 11 BNs the ratio is less than 0.3. We remark that GFB is *useful*, because the analysis of the original BNs, the AND- and OR-reductions took on average 100s, 30s and 60s, respectively. Notably, reductions with low reduction ratios are particularly fast (fewer algorithm iterations): the 6 AND-reductions in Figure 2 (left) with ratio smaller than 0.3 take less than 1.5 seconds on average.

7.2 Nonlinear Reduction of a Lotka-Volterra Model over Monoid $(\mathbb{R}, \cdot)$

We present an example of exact lumping where ψ_R is not linear, and thus cannot be captured by linear lumping methods such FDE. We use the commutative monoid $(\mathbb{R}, \cdot)$ with the neutral element 1, and a prototypical three variables higher-order Lotka-Volterra model [42] where

[5] Please contact us if interested in the tool, the models, and all material to replicate the experiments in this paper.

[6] We could not consider MVs because we are not aware of tools for general attractor analysis for MVs.

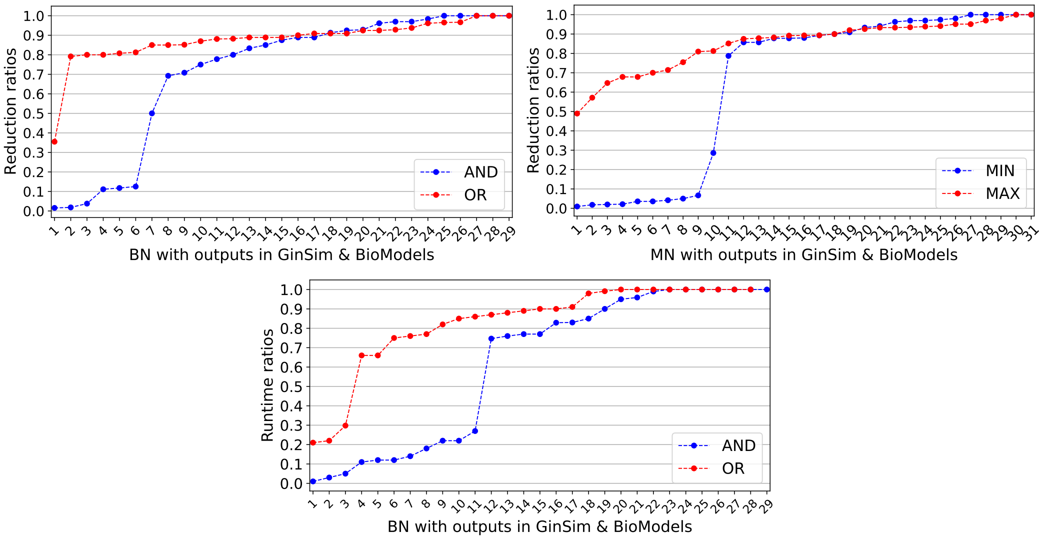

Figure 2 (Left) Reduction ratios (reduced variables over original ones) in ascending order for the 29 BN with outputs from GINsim and BioModelsDB. We used $\oplus \in \{\wedge, \vee\}$ with initial partitions with two blocks separating output and non-output species. **(Right)** Same as (Left) for the 31 MV networks with outputs from the two repositories considering $\oplus \in \{\min, \max\}$. **(Bottom)** Runtime ratios in ascending order for computation of attractors for the 29 BNs and their reductions.

x_1 preys x_2 and x_3, while x_2 and x_3 prey together x_1. The corresponding ODE system is

$$\partial_t v_{x_1} = v_{x_1}(1 - v_{x_2} v_{x_3}), \qquad \partial_t v_{x_2} = v_{x_2}(1 - v_{x_1}), \qquad \partial_t v_{x_3} = v_{x_3}(1 - v_{x_1}). \tag{4}$$

The underlying ODE discretization of (4) is

$$f_{x_1}(s)=s_{x_1}+\tau s_{x_1}(1-s_{x_2}s_{x_3}), \quad f_{x_2}(s)=s_{x_2}+\tau s_{x_2}(1-s_{x_1}), \quad f_{x_3}(s)=s_{x_3}+\tau s_{x_3}(1-s_{x_1}).$$

By Theorem 13, the *nonlinear* function $\psi_R(v_{x_1}, v_{x_2}, v_{x_3}) = (v_{x_1}, v_{x_2} \cdot v_{x_3})$ is an exact lumping of (4). Indeed, $\mathcal{X}_R = \{\{x_1\}, \{x_2, x_3\}\}$ is a GFB of (4) when $\oplus = \cdot$ because $\Psi^{\mathcal{X}_R}_{x_2,x_3}$ is then valid thanks to the identities $f_{x_1} = x_1 + \tau x_1(1 - x_2 x_3) = f_{x_1}[x_2/1, x_3/x_2x_3]$ and

$$\begin{aligned} f_{x_2} \cdot f_{x_3} &= (x_2 + \tau x_2(1 - x_1)) \cdot (x_3 + \tau x_3(1 - x_1)) \\ &= x_2 x_3 + 2\tau x_2 x_3(1 - x_1) + \tau^2 x_2 x_3 (1 - x_1)^2 \;=\; (f_{x_2} \cdot f_{x_3})[x_2/1, x_3/x_2x_3]. \end{aligned}$$

The lumped ODE system is then given by $\partial_t v_{x_1} = v_{x_1}(1 - v_{x_2} v_{x_3})$ and $\partial_t (v_{x_2} v_{x_3}) = \partial_t v_{x_2} \cdot v_{x_3} + v_{x_2} \cdot \partial_t v_{x_3} = v_{x_2}(1 - v_{x_1}) v_{x_3} + v_{x_2} v_{x_3} (1 - v_{x_1}) = 2 v_{x_1} v_{x_2} (1 - v_{x_1})$.

8 Conclusion

We introduced generalized forward bisimulation, a technique for dimensionality reduction of dynamical systems over commutative monoids. One needs to specify a dynamical system (i.e., a set of variables and their corresponding update functions), a commutative monoid (the variables' domain and an operation), and an initial partition of the variables. These are fed to a partition refinement algorithm that minimizes the system over the operation of the monoid. We implemented the method and applied it to 3 popular formalisms: difference equations with monoid $(\mathbb{R}, \cdot)$, Boolean networks $(\mathbb{B}, \wedge)$ and $(\mathbb{B}, \vee)$, and multi-valued networks $(\{0, 1, 2\}, \min)$ and $(\{0, 1, 2\}, \max)$. In all these cases, we got nonlinear reductions. Considering 60 Boolean and multi-valued networks from two popular repositories, we have shown the high aggregation power of our technique, and the analysis speed-ups that it can offer.

References

1 A. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Advances in Design and Control. SIAM, 2005.

2 Georgios Argyris, Alberto Lluch Lafuente, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Reducing boolean networks with backward boolean equivalence. In *International Conference on Computational Methods in Systems Biology*, pages 1–18. Springer, 2021.

3 Georgios Argyris, Alberto Lluch-Lafuente, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Reducing boolean networks with backward boolean equivalence. In *Computational Methods in Systems Biology, CMSB*, pages 1–18, 2021. doi:10.1007/978-3-030-85633-5_1.

4 Eugenio Azpeitia, Mariana Benítez, Iliusi Vega, Carlos Villarreal, and Elena R Alvarez-Buylla. Single-cell and coupled grn models of cell patterning in the arabidopsis thaliana root stem cell niche. *BMC systems biology*, 4(1):1–19, 2010.

5 Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Efficient local computation of differential bisimulations via coupling and up-to methods. In *Symposium on Logic in Computer Science, LICS*, pages 1–14, 2021.

6 Michele Boreale. Algebra, coalgebra, and minimization in polynomial differential equations. *Log. Methods Comput. Sci.*, 15(1), 2019. doi:10.23638/LMCS-15(1:14)2019.

7 Michele Boreale. Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial odes. *Sci. Comput. Program.*, 193:102441, 2020.

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

9 L. Cardelli, M. Tribastone, Max Tschaikowski, and A. Vandin. Maximal aggregation of polynomial dynamical systems. *Proceedings of the National Academy of Sciences*, 114(38):10029 – 10034, 2017.

10 Luca Cardelli, Isabel Cristina Pérez-Verona, Mirco Tribastone, Max Tschaikowski, Andrea Vandin, and Tabea Waizmann. Exact maximal reduction of stochastic reaction networks by species lumping. *Bioinform.*, 37(15):2175–2182, 2021. doi:10.1093/bioinformatics/btab081.

11 Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Forward and backward bisimulations for chemical reaction networks. In *26th International Conference on Concurrency Theory, CONCUR*, pages 226–239, 2015. URL: http://cse.lab.imtlucca.it/~mirco.tribastone/papers/concur2015.pdf, doi:10.4230/LIPIcs.CONCUR.2015.226.

12 Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Symbolic computation of differential equivalences. In *POPL*, pages 137–150, 2016. doi:10.1145/2837614.2837649.

13 Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Erode: a tool for the evaluation and reduction of ordinary differential equations. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 310–328. Springer, 2017.

14 Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Guaranteed error bounds on approximate model abstractions through reachability analysis. In *QEST*, pages 104–121, 2018. doi:10.1007/978-3-319-99154-2_7.

15 Claudine Chaouiya, Duncan Bérenguier, Sarah M Keating, Aurélien Naldi, Martijn P Van Iersel, Nicolas Rodriguez, Andreas Dräger, Finja Büchel, Thomas Cokelaer, Bryan Kowal, et al. SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC systems biology*, 7(1):1–15, 2013.

16 Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

17 Rocco De Nicola, Ugo Montanari, and Frits Vaandrager. Back and forth bisimulations. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR '90 Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 152–165. Springer Berlin Heidelberg, 1990. URL: http://dx.doi.org/10.1007/BFb0039058, doi:10.1007/BFb0039058.

18 Elena Dubrova and Maxim Teslenko. A sat-based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 8(5):1393–1399, 2011.

19 A Fauré, BMI Vreede, E Sucena, and C Chaouiya. A discrete model of drosophila eggshell patterning reveals cell-autonomous and juxtacrine effects. *PLoS Comput Biol*, 10:e1003527, 2014. `doi:10.1371/journal.pcbi.1003527`.

20 C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1971.

21 Martin Hopfensitz, Christoph Müssel, Markus Maucher, and Hans A Kestler. Attractors in boolean networks: a tutorial. *Computational Statistics*, 28(1):19–36, 2013.

22 S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437 – 467, 1969.

23 P. E. Kloeden and J. Lorenz. Stable attracting sets in dynamical systems and in their one-step discretizations. *SIAM Journal on Numerical Analysis*, 23(5):986–995, 1986. `doi:10.1137/0723066`.

24 Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

25 Genyuan Li and Herschel Rabitz. A general analysis of exact lumping in chemical kinetics. *Chemical Engineering Science*, 44(6):1413–1430, 1989.

26 Genyuan Li, Herschel Rabitz, and János Tóth. A general analysis of exact nonlinear lumping in chemical kinetics. *Chemical Engineering Science*, 49(3):343–361, 1994. URL: `https://www.sciencedirect.com/science/article/pii/0009250994870063`, `doi:https://doi.org/10.1016/0009-2509(94)87006-3`.

27 Rahuman S Malik-Sheriff, Mihai Glont, Tung V N Nguyen, Krishna Tiwari, Matthew G Roberts, Ashley Xavier, Manh T Vu, Jinghao Men, Matthieu Maire, Sarubini Kananathan, Emma L Fairbanks, Johannes P Meyer, Chinmay Arankalle, Thawfeek M Varusai, Vincent Knight-Schrijver, Lu Li, Corina Dueñas-Roca, Gaurhari Dass, Sarah M Keating, Young M Park, Nicola Buso, Nicolas Rodriguez, Michael Hucka, and Henning Hermjakob. BioModels — 15 years of sharing computational models in life science. *Nucleic Acids Research*, 48(D1):D407–D415, 1 2020. gkz1055. `doi:10.1093/nar/gkz1055`.

28 John Milnor. On the concept of attractor. *Communications in Mathematical Physics*, 99(2):177–195, 1985. `doi:10.1007/BF01212280`.

29 A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya. Logical modelling of regulatory networks with ginsim 2.3. *Biosystems*, 97(2):134–139, 2009.

30 Aurélien Naldi, Duncan Berenguier, Adrien Fauré, Fabrice Lopez, Denis Thieffry, and Claudine Chaouiya. Logical modelling of regulatory networks with ginsim 2.3. *Biosystems*, 97(2):134–139, 2009.

31 Aurélien Naldi, Pedro T Monteiro, and Claudine Chaouiya. Efficient handling of large signalling-regulatory networks by focusing on their core control. In *International Conference on Computational Methods in Systems Biology*, pages 288–306. Springer, 2012.

32 Aurélien Naldi, Pedro T Monteiro, Christoph Müssel, Consortium for Logical Models, Tools, Hans A Kestler, Denis Thieffry, Ioannis Xenarios, Julio Saez-Rodriguez, Tomas Helikar, and Claudine Chaouiya. Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics*, 31(7):1154–1159, 2015.

33 Aurélien Naldi, Elisabeth Remy, Denis Thieffry, and Claudine Chaouiya. Dynamically consistent reduction of logical regulatory graphs. *Theoretical Computer Science*, 412(21):2207–2218, 2011.

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

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

36 George J. Pappas, Gerardo Lafferriere, and Shankar Sastry. Hierarchically consistent control systems. *IEEE Trans. Automat. Contr.*, 45(6):1144–1160, 2000. doi:10.1109/9.863598.

37 George J. Pappas and Slobodan Simic. Consistent abstractions of affine control systems. *IEEE Trans. Automat. Contr.*, 47(5):745–756, 2002. doi:10.1109/TAC.2002.1000269.

38 D. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science*, pages 167–183, 1981.

39 Daniel Richardson. Some undecidable problems involving elementary functions of a real variable. *The Journal of Symbolic Logic*, 33(4):514–520, 1968.

40 Davide Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.

41 Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.

42 P. Singh and G. Baruah. Higher order interactions and species coexistence. *Theoretical Ecology*, 14:71–83, 2021.

43 Thomas J. Snowden, Piet H. van der Graaf, and Marcus J. Tindall. Methods of model reduction for large-scale biological systems: A survey of current methods and trends. *Bulletin of Mathematical Biology*, 79(7):1449–1486, 2017. doi:10.1007/s11538-017-0277-2.

44 Claudia Stötzel, Susanna Röblitz, and Heike Siebert. Complementing ode-based system analysis using boolean networks derived from an euler-like transformation. *PloS one*, 10(10):e0140954, 2015.

45 René Thomas, Denis Thieffry, and Marcelle Kaufman. Dynamical behaviour of biological regulatory networks — i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of mathematical biology*, 57(2):247–276, 1995.

46 Alison S Tomlin, Genyuan Li, Herschel Rabitz, and János Tóth. The effect of lumping and expanding on kinetic differential equations. *SIAM Journal on Applied Mathematics*, 57(6):1531–1556, 1997.

47 A. J. van der Schaft. Equivalence of dynamical systems by bisimulation. *IEEE Transactions on Automatic Control*, 49, 2004.

48 Alan Veliz-Cuba. Reduction of boolean network models. *Journal of theoretical biology*, 289:167–172, 2011.

A Proofs

Proof of Theorem 6. Let R be a GFB, pick $s_0 \in \mathbb{M}^X$ and set $\hat{s}_0 = \psi_R(s_0) \in \mathbb{M}^{\mathcal{X}_R}$. We next show that $\hat{s}_k = \psi_R(s_k)$ by induction over $k \geq 0$. Since the base case $k = 0$ is true by construction, we can turn to the induction step. For $k \geq 0$, we obtain

$$\hat{s}_{k+1} = F_R(\hat{s}_k) = F_R(\psi_R(s_k)) = \psi_R(F(s_k)) = \psi_R(s_{k+1}),$$

where the second identity follows from the induction hypothesis, while the third identity follows from the definition of F_R and the fact that R is a GFB. Conversely, if $\hat{s}_k = \psi_R(s_k)$ for all $k \geq 0$, we can conclude for $k = 0$ and arbitrary $s_0 \in \mathbb{M}^X$ that

$$\psi_R(F(s_0)) = \psi_R(s_1) = \hat{s}_1 = F_R(\hat{s}_0) = F_R(\psi_R(s_0)),$$

thus showing that R is a GFB. ◀

Proof of Theorem 8. Fix arbitrary GFBs $\sim_1, \ldots, \sim_\nu \subseteq R$ and let $\mathcal{H}_1, \ldots, \mathcal{H}_\nu$ be the corresponding partitions, i.e., $\mathcal{H}_i = X_{\sim_i}$. Moreover, let $\sim_* := \left(\bigcup_{i=1}^m \sim_i \right)^*$ and $\mathcal{H}^* := X_{\sim_*}$, where the asterisk denotes transitive closure of a relation. At last, let $x_{i_{H^*}} \in H^*$ denote some representative of $H^* \in \mathcal{H}^*$. With this, pick an arbitrary $H^* \in \mathcal{H}^*$. By construction of $\mathcal{H}^*$, there exist $x_0, \ldots, x_k \in X$ and $i_0, \ldots, i_{k-1} \in \{1, \ldots, \nu\}$ so that $\{x_0, \ldots, x_k\} = H^*$, $x_k = x_{i_{H^*}}$ and $x_j \sim_{i_j} x_{j+1}$ for all $0 \leq j \leq k-1$. Moreover, for any $G^* \in \mathcal{H}^*$ and $1 \leq i \leq \nu$, there exist (unique) $G^i_1, \ldots, G^i_{m_i} \in \mathcal{H}_i$ such that $\biguplus_{l=1}^{m_i} G^i_l = G^*$. Since $x_j \sim_{i_j} x_{j+1}$ and $\mathcal{H}_{i_j}$ is a GFB, we obtain

$$\begin{aligned}
\bigoplus_{x_\iota \in G^*} f_{x_\iota} &= \bigoplus_{l=1}^{m_{i_j}} \bigoplus_{x_\iota \in G^{i_j}_l} f_{x_\iota} \\
&= \bigoplus_{l=1}^{m_{i_j}} \bigoplus_{x_\iota \in G^{i_j}_l} f_{x_\iota}[x_j/0_\oplus][x_{j+1}/(x_j \oplus x_{j+1})] \\
&= \bigoplus_{x_\iota \in G^*} f_{x_\iota}[x_j/0_\oplus][x_{j+1}/(x_j \oplus x_{j+1})]
\end{aligned}$$

Since $\{x_0, x_1, \ldots, x_k\} = H^*$ and $x_k = x_{i_{H^*}}$, an application of the argument for all $0 \leq j \leq k-1$ implies that $\bigoplus_{x_\iota \in G^*} f_{x_\iota}$ is equivalent to

$$\bigoplus_{x_\iota \in G^*} f_{x_\iota}[x_k/0_\oplus : x_k \in H^*, x_k \neq x_{i_{H^*}}][x_{i_{H^*}}/ \bigoplus_{x_l \in H^*} x_l]$$

Since the choice of $G^*, H^* \in \mathcal{H}^*$ was arbitrary, we infer that $\mathcal{H}^*$ is a GFB. ◀

Proof of Theorem 9. Let us assume first that $\mathcal{X}_R$ is a GFB, pick an arbitrary $(x_i, x_j) \in R$ and pick the unique $C' \in \mathcal{X}_R$ such that $x_i, x_j \in C'$. With this, define $s' := s[x_i \mapsto 0_\oplus][x_j \mapsto s_{x_i} \oplus s_{x_j}]$ for an arbitrary $s \in \mathbb{M}^X$, where $s[x_k \mapsto b]_{x_k} = b$ and $s[x_k \mapsto b]_{x_l} = s_{x_l}$ for all $b \in \mathbb{M}$ and $x_l \neq x_k$. Then, since $\oplus$ is commutative and associative and because $\mathcal{X}_R$ is a GFB, we have that

$$\bigwedge_{C \in \mathcal{X}_R} \Big(\bigoplus_{x_i \in C} f_{x_i}(s) = \bigoplus_{x_i \in C} f_{x_i}(s') \Big). \tag{5}$$

Since the choice of $(x_i, x_j) \in R$ and $s \in \mathbb{M}^X$ was arbitrary, we infer that $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid. For the converse, let us assume that $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ holds true for all $(x_i, x_j) \in R$ and pick any two $s, s' \in \mathbb{M}^X$ such that

$$\bigwedge_{C\in\mathcal{X}_R} \Big(\bigoplus_{x_i\in C} s_{x_i} = \bigoplus_{x_i\in C} s'_{x_i} \Big) \tag{6}$$

With this, pick for any $C \in \mathcal{X}_R$ some arbitrary representative $x_{i_C} \in C$ and let $\hat{X} = \{x_{i_C} : C \in \mathcal{X}_R\}$ be the set of all representatives. For any $(x_i, x_j) \in R$, define $s_{i\to j} := s[x_i \mapsto 0_\oplus, x_j \mapsto s_{x_i} \oplus s_{x_j}]$. With this, the fact that $\oplus$ is commutative and associative ensures the existence of a sequence $x_{i_1}, x_{i_2}, ..., x_{i_k}$ for which $\hat{s} = (((s_{i_1\to i_2})_{i_2\to i_3})\ldots)_{i_{k-1}\to i_k}$ is such that

$$\bigwedge_{C\in\mathcal{X}_R} \Big(\bigoplus_{x_i\in C} s_{x_i} = \bigoplus_{x_i\in C} \hat{s}_{x_i} \Big),$$

$\hat{s}_{x_i} = 0_\oplus$ for all $x_i \notin \hat{X}$ and $\hat{s}_{x_{i_C}} = \bigoplus_{x_i\in C} s_{x_i}$ for all $C \in \mathcal{X}_R$. Since $\Psi^{\mathcal{X}_R}_{x_{i_l},x_{i_{l+1}}}$ is valid for all $1 \le l \le k-1$, we obtain

$$\bigwedge_{C\in\mathcal{X}_R} \Big(\bigoplus_{x_i\in C} f_{x_i}(s) = \bigoplus_{x_i\in C} f_{x_i}(\hat{s}) \Big).$$

A similar argument for s' ensures that there is an $\hat{s}'$ such that $\hat{s}'_{x_i} = 0_\oplus$ for all $x_i \notin \hat{X}$, $\hat{s}'_{x_{i_C}} = \bigoplus_{x_i\in C} s'_{x_i}$ for all $C \in \mathcal{X}_R$ and

$$\bigwedge_{C\in\mathcal{X}_R} \Big(\bigoplus_{x_i\in C} s'_{x_i} = \bigoplus_{x_i\in C} \hat{s}'_{x_i} \Big),$$
$$\bigwedge_{C\in\mathcal{X}_R} \Big(\bigoplus_{x_i\in C} f_{x_i}(s') = \bigoplus_{x_i\in C} f_{x_i}(\hat{s}') \Big).$$

Thanks to (6), we infer that $\hat{s} = \hat{s}'$. This, in turn, implies the desired relation (5), thus showing that $\mathcal{X}_R$ is a GFB if and only if $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid for all $(x_i, x_j) \in R$. ◀

Proof of Theorem 10. Pick the largest (i.e., coarsest) GFB $\mathcal{H}_*$ that refines X_R using Theorem 8. With this, set $\mathcal{H}_0 := X_R$ and define for all $k \ge 0$ and $H \in \mathcal{H}_k$

$$R_k(H) := \{(x_i, x_j) \in H \times H : x_i \neq x_j \Rightarrow \Psi^{\mathcal{H}_k}_{x_i,x_j} \wedge \Psi^{\mathcal{H}_k}_{x_j,x_i}\}$$
$$\mathcal{H}_{k+1} := \bigcup_{H\in\mathcal{H}_k} H/R^*_k(H),$$

where $R^*_k(H)$ denotes the transitive closure of $R_k(H)$. By construction, $R_k(H)$ is reflexive and symmetric, thus implying $\bigoplus_{x_i\in H} f_{x_i}(s) = \bigoplus_{x_i\in H} f_{x_i}(\tilde{s})$ for all $s \in \mathbb{M}^X$, $H \in \mathcal{H}_k$, where

$$\tilde{s} = s[x_j \mapsto 0_\oplus : x_j \notin \hat{X}_{k+1}][x_{i_{C'}} \mapsto \bigoplus_{x_j\in C'} s_{x_j} : C' \in \mathcal{H}_{k+1}]$$

and $x_{i_C} \in C$ is a representative of class $C \in \mathcal{H}_{k+1}$, while $\hat{X}_{k+1} = \{x_{i_C} : C \in \mathcal{H}_{k+1}\}$. (Note that $H \in \mathcal{H}_k$, while $C \in \mathcal{H}_{k+1}$ and $\hat{X}_{k+1}$ is defined using $\mathcal{H}_{k+1}$.) This implies that R_k is transitive. Indeed, for any $(x_i, x_j), (x_j, x_k) \in R_k$ and $s' \in \mathbb{M}^X$, the previous equation ensures for state $s := s'[x_i \mapsto 0_\oplus, x_k \mapsto s'_{x_i} \oplus s'_{x_k}]$ and any $H \in \mathcal{H}_k$ that

$$\bigoplus_{x_i\in H} f_{x_i}(s) = \bigoplus_{x_i\in H} f_{x_i}(\tilde{s}') = \bigoplus_{x_i\in H} f_{x_i}(\tilde{s}'') = \bigoplus_{x_i\in H} f_{x_i}(\tilde{s}'''),$$

where

$$\tilde{s}' = s[x_l \mapsto 0_\oplus : x_l \notin \hat{X}_{k+1}][x_{i_{C'}} \mapsto \bigoplus_{x_j \in C'} s_{x_j} : C' \in \mathcal{H}_{k+1}],$$
$$\tilde{s}'' = s'[x_l \mapsto 0_\oplus : x_l \notin \hat{X}_{k+1}][x_{i_{C'}} \mapsto \bigoplus_{x_j \in C'} s'_{x_j} : C' \in \mathcal{H}_{k+1}],$$
$$\tilde{s}''' = s'[x_i \mapsto 0_\oplus, x_j \mapsto 0_\oplus, x_k \mapsto s'_{x_i} \oplus s'_{x_j} \oplus s'_{x_k}].$$

Hence, $R_k^* = R_k$ and the expression H/R is indeed well-defined in Algorithm 1. Further, a proof by induction over $k \geq 1$ shows that a) $\mathcal{H}_*$ is a refinement of $\mathcal{H}_k$ and b) $\mathcal{H}_k$ is a refinement of $\mathcal{H}_{k-1}$. Since $\mathcal{H}_*$ is a refinement of any $\mathcal{H}_k$, it holds that $\mathcal{H}_* = \mathcal{H}_k$ if $\mathcal{H}_k$ is a GFB partition. Since X is finite, b) allows us to fix the smallest $k \geq 1$ with $\mathcal{H}_k = \mathcal{H}_{k-1}$. This, in turn, implies that $\mathcal{H}_{k-1}$ is a GFB. To see the complexity statement, we note that the algorithm can perform at most $|X|$ refinements, while each iteration compares $\mathcal{O}(|X|^2)$ pairs. For the decidability, instead, we first note that the finiteness of $\mathbb{M}$ ensures the finiteness of $\oplus \subseteq \mathbb{M} \times \mathbb{M}$ and any $f_{x_i} \subseteq \mathbb{M}^X \times \mathbb{M}$. Hence, checking

$$\bigwedge_{C \in \mathcal{H}} \Big(\bigoplus_{x_k \in C} f_{x_k} = \bigoplus_{x_k \in C} f_{x_k}[x_i/0_\oplus][x_j/(x_i \oplus x_j)] \Big)$$

amounts to a finite number of checks over finite sets and is thus decidable. ◀

Proof of Theorem 13. See proof of Theorem 15. ◀

Proof of Lemma 14. We start by noting that $\psi_R(v_{x_1}, v_{x_2}) = \log(v_{x_1}) + \log(v_{x_2})$ is an exact lumping because $\partial_t \psi_R(v) = (v_{x_1}^{-1}, v_{x_2}^{-1}) \cdot (\partial_t v_{x_1}, \partial_t v_{x_2})^T = \psi_R(v)$. At the same time, the ODE discretization of the model is

$$f_{x_1} = x_1 + \tau x_1 \log(x_2), \qquad\qquad f_{x_2} = x_2 + \tau x_2 \log(x_1).$$

Writing $h = \tau x_1 x_2 \log(x_1 x_2) + \tau^2 x_1 x_2 \log(x_1)\log(x_2)$ for convenience, we observe that

$$\begin{aligned} \log(f_{x_1}) + \log(f_{x_2}) = \log(f_{x_1} f_{x_2}) &= \log\big(x_1 x_2 + h\big) \\ &= \log(x_1 x_2) + (\partial \log)(x_1 x_2) h + (\partial^2 \log)(x_1 x_2) \tfrac{h^2}{2} + \mathcal{O}(\tau^3) \\ &= \log(x_1 x_2) + \frac{h}{x_1 x_2} - \frac{h^2}{2 x_1^2 x_2^2} + \mathcal{O}(\tau^3) \\ &= \log(x_1 x_2) + \tau \log(x_1 x_2) + \tau^2 \log(x_1)\log(x_2) - \tau^2 \tfrac{1}{2} \log(x_1 x_2)^2 + \mathcal{O}(\tau^3). \end{aligned}$$

Here, $\mathcal{O}$ refers to *big O* notation from numerical analysis, while the third identity follows from Taylor's theorem and from $\partial \log(x) = x^{-1}$ and $\partial^2 \log(x) = -x^{-2}$. Since the higher-order term $\tau^2 \log(x_1)\log(x_2)$ cannot be expressed in terms of $\log(x_1 x_2)$, we conclude that $\mathcal{X}_R$ is not a GFB. ◀

Proof of Theorem 15. To improve readability, we write ψ instead of ψ_R in the present proof. Since $\oplus$ is twice differentiable by assumption, so is $\psi = (\psi_H)_{H \in \mathcal{X}_R}$. For any $H \in \mathcal{X}_R$, Taylor's theorem thus ensures

$$\begin{aligned} \psi_H(F(s)) &= \psi_H(s + \tau \Phi(s)) \\ &= \psi_H(s) + (\partial_s \psi_H)(s + \tau \Phi(s)) \cdot \tau \Phi(s) + \mathcal{O}(\tau^2) \\ &= \psi_H(s) + \tau \cdot (\partial_s \psi_H)(s + \tau \Phi(s)) \cdot \Phi(s) + \mathcal{O}(\tau^2) \end{aligned}$$

We begin by assuming that ψ is an exact lumping. Then, with $\partial_t v(t) = \Phi(v(t))$, the derivative of $t \mapsto \psi_H(v(t))$ can be written as a function of $\big(\psi_C(v(t))\big)_{C \in \mathcal{X}_R}$ by [46]. Since $v(0) \in \mathbb{R}^X$ can be chosen arbitrarily, there is thus a function $\wp_H$ such that $\wp_H(\psi(s)) = (\partial_s \psi_H)(s + \tau\Phi(s)) \cdot \Phi(s)$ for all $s \in \mathbb{R}^X$. Overall, we conclude for all $s \in \mathbb{R}^X$

$$\psi_H(F(s)) = \psi_H(s) + \tau \cdot \wp_H(\psi(s)) + \mathcal{O}(\tau^2)$$

Since $H \in \mathcal{H}$ can be chosen arbitrarily, following the argumentation from the proof of Theorem 9, we infer that for all $(x_i, x_j) \in R$ with $x_i \neq x_j$ formula $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid up to $\mathcal{O}(\tau^2)$. For the converse, let us assume that for all $(x_i, x_j) \in R$ with $x_i \neq x_j$ formula $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid up to $\mathcal{O}(\tau^2)$. Then, Taylor's theorem yields as before

$$\psi_H(F(s)) = \psi_H(s) + \tau \cdot (\partial_s \psi_H)(s + \tau\Phi(s)) \cdot \Phi(s) + \mathcal{O}(\tau^2)$$

With this and the validity of the aforementioned $\Psi^{\mathcal{X}_R}_{x_i,x_j}$, the argumentation from the proof of Theorem 9 ensures the existence of functions $(\wp_H)_{H \in \mathcal{X}_R}$ over $\mathbb{R}^{\mathcal{X}_R}$ such that

$$\psi_H(F(s)) = \psi_H(s) + \tau \cdot \wp_H(\psi(s)) + \mathcal{O}(\tau^2)$$

for all $H \in \mathcal{X}_R$ and $s \in \mathbb{R}^X$. Hence, with $\partial_t v(t) = \Phi(v(t))$, the derivative of $t \mapsto \psi_H(v(t))$ can be written as a function of $\big(\psi_C(v(t))\big)_{C \in \mathcal{X}_R}$. Since $v(0) \in \mathbb{R}^X$ can be chosen arbitrarily, we obtain that ψ is an exact lumping. This completes the proof of Theorem 15. We next turn to the proofs of 1) and 2) of Theorem 13. For 1), we note that $\Psi^{\mathcal{X}_R}_{x_i,x_j}$ is valid up to $\mathcal{O}(\tau^2)$ for all $(x_i, x_j) \in R$ when R is a GFB. Instead, for 2) we observe that for a linear ψ_R there are no higher-order terms, i.e., $\mathcal{O}(\tau^2) = 0$. This two observations, combined with the foregoing discussion, yield statements 1) and 2). ◀

Proof of Corollary 17. Set $\oplus = +$ in Theorem 13. ◀

Proof of Corollary 18. The vector of transient probabilities of the Markov chain at time $t \geq 0$ satisfies the forward Kolmogorov equations $\partial_t \pi(t) = Q^T \pi(t)$. Moreover, by [12], an equivalence relation R over X is an ordinary lumpability if and only if R is an FDE the forward Kolmogorov equations. With this, Corollary 17 yields the equivalence of 1) and 2). The equivalence of 2) and 3), instead, is a well-known fact [8]. ◀