Minimization of Dynamical Systems over Monoids

Georgios Argyris*, Alberto Lluch Lafuente*, Alexander Leguizamon Robayo†, Mirco Tribastone‡,
Max Tschaikowski†, and Andrea Vandin§,*
*DTU Compute, Technical University of Denmark, Denmark
†Department of Computer Science, Aalborg University, Denmark
‡SysMA Research Unit, IMT Lucca, Italy
‡Institute of Economics and EMbeDS, 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 ordinary differential equations (ODEs). In *forward bisimulations*, 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 coarsest one. When the monoid is $(\mathbb{R}, +)$, we recover probabilistic bisimulation for Markov chains and more recent forward bisimulations for nonlinear ODEs. Using $(\mathbb{R}, \cdot)$ we get nonlinear reductions for discrete-time dynamical systems and ODEs 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 (BN), a widely used dynamical model in computational biology. Using a prototype implementation of our minimization algorithm for GFB, we find disjunction- and conjunction-preserving reductions on 60 BN from two well-known repositories, and demonstrate the obtained analysis speed-ups. We also provide the biological interpretation of the reduction obtained for two selected BN, and we show how GFB enables the analysis of a large one that could not be analyzed otherwise. Using a randomized version of our algorithm we find product-preserving (therefore non-linear) reductions on 21 dynamical weighted networks from the literature that could not be handled by the exact algorithm.**

I. INTRODUCTION

Bisimulation is a fundamental tool in computer science for abstraction and minimization, relating models by useful logical and dynamical properties [1]. Originally developed for concurrent processes in a non-quantitative setting [2], it has been extended to quantitative models such as, e.g., the notion of probabilistic bisimulation [3] closely related to ordinary lumpability for Markov chains [4]. *Forward* bisimulations relate states based on criteria that depend on their *outgoing* transitions (as opposed to *backward* bisimulations that depend on *incoming* ones, e.g., [5]). When applied to a dynamical system (DS), forward bisimulations preserve sums of values of state variables. E.g., probabilistic bisimulation yields a quotient Markov process where each state represents an equivalence class preserving 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 [6], [7]; forward differential equivalence (FDE) for nonlinear ordinary differential equations (ODEs) relates variables preserving sums of their solutions [8].

An attractive feature of bisimulation is that one can compute the largest bisimulation using partition refinement, based on the pioneering solution for concurrent processes [9]. Partition refinement algorithms start from an *initial partition* of variables which is iteratively refined (i.e., its blocks get split) until the partition is a bisimulation. Notably, initial partitions are useful to tune the reduction. For example, one can *separate* groups of variables according to given criteria to prevent that variables from different groups will be aggregated together. This makes bisimulation an effective approach for the minimization of complex DS, adding to cross-disciplinary methods originated in e.g., chemical and performance engineering [10], [11], control theory [12], and systems biology [13].

Thus far, one can identify two common properties of existing forward bisimulations for DS: they preserve sums of state values, and the DS variables take real $\mathbb{R}$ values. There are, however, motivations to generalize this setting. A forward bisimulation for ODEs can be seen as a special case of *linear lumping* [10], a minimization achieved by a linear projection of the state space by a matrix that encodes the partition of 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 [14].

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

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

where variables x_1, x_2, and x_3 are defined over the Booleans $\mathbb{B} = \{0, 1\}$, and k denotes discrete time. This is a Boolean network (BN), an established model of biological systems [15].

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. For example, 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

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

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

$$\begin{aligned} x_{1,2}(k+1) &= x_3(k) \vee x_{1,2}(k) \\ x_3(k+1) &= \neg x_3(k) \wedge x_{1,2}(k). \end{aligned} \quad (2)$$

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.

Here we show that GFB satisfies desirable properties for bisimulations.

1) Over any commutative monoid $(\mathbb{M}, \oplus)$, GFB characterizes $\oplus$-preserving reductions, in the sense that any DS with fewer state variables which coincides with $\oplus$-operations of original state variables must necessarily be the quotient of a GFB. This generalizes characterization results for Markov chains [3], chemical reaction networks [16], and nonlinear ODEs [8]. Notably, our characterization result also covers the asymptotic dynamics, often of interest when analyzing DS (see, e.g., [17]). We show that GFB preserves all *attractors*, i.e., the states towards which the DS tends to evolve and remain.

2) 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 DS under study. In general, this can be undecidable. However, when the monoid is $(\mathbb{R}, +)$ our algorithm reduces to that for FDE for nonlinear ODEs [8], [16], [18]. Instead, when the domain is $\mathbb{B}$, the problem corresponds to Boolean satisfiability.

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

4) GFB is effective in practice, both in terms of reduction power and of obtained analysis speed-ups.

Previous results are essentially agnostic to whether the time evolution of the DS is continuous or discrete. More specifically, the criteria for probabilistic bisimulation [3] 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 $x(k+1) = f(x(k))$. With GFB, instead, more care has to be taken because this verbatim correspondence does not hold any longer. For this reason, we first develop GFB for discrete-time DS. Then, we consider continuous time by studying GFB for DS over the reals relating to, and extending, results for ODEs.

Applications. Using a prototype implementation in the tool ERODE [19], we apply GFB to case studies from different domains. We consider Boolean and multi-valued networks [15], [20], where the latter allows for finer degrees of activation than just 0/1 as in (1). These models are known to suffer from state-space explosion, making model reduction appealing (see, e.g., [21]). Two selected case studies from the literature showcase the physical intelligibility of GFB reductions, and a third one shows how GFB can enable the analysis of otherwise intractable BNs. In the three case studies, we show how initial partitions can be devised using domain knowledge. For example, 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 denote the (de)activation of different mechanisms of the model. We also perform a large-scale validation of GFB on 60 Boolean and multi-valued networks from established repositories (GinSim [22], BioModelsDB [23]). We show that *default* initial partitions can be synthesized automatically. We also show that GFB is *useful* due to its high reduction power and analysis speed-ups. We also consider real-valued DS. We study a case study of a higher-order Lotka-Volterra model [24], and we perform a large-scale validation on 72 weighted networks from the Netzschleuder repository [25].

II. RELATED WORK

Most of the literature about model minimization can be found for DS over the reals. In this context, the framework of exact lumping considers reductions by means of both linear and nonlinear operators [26], [27]. The aforementioned notions of bisimulation for Markov chains and FDE, including its stochastic variants [28], [29], 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 [10]. Since also GFB is developed in this style, it too can be seen as a special case of exact lumping. However, the coarsest 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 Section V. Instead, the computation of exact lumpings hinges, in the case of polynomial dynamics, on symbolic computations with worst-case exponential complexity [14, Section 2.2].

Relying on polynomial invariants [30], [31], $\mathcal{L}$-bisimulation [32], [33] can be seen as a generalization of backward differential equivalence (BDE) [34]–[36], a backward-type bisimulation for non-linear ODEs, and is thus complementary to FDE (hence, GFB), as discussed in [32], [33], [37]. 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) [38]–[40] is complementary to GFB. Indeed, for a so-called observation function, the coarsest consistent abstraction gives rise to a minimal reduced DS which coincides with the original one up to the chosen observation function. Instead,

computing the coarsest GFB corresponds to finding an observation function which induces a minimal 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 [38].

In the model checking community there has been a large amount of work on predicate abstraction techniques, including counterexample-guided abstraction refinement (CEGAR) [41] and abstraction invariant checking (see, e.g., the recent work in [42] for polynomial ODEs). Our approach consists in a model-to-model abstraction reduction, where the final result is a reduced model specified in the same specification language as the original one (i.e., systems of equations). The abstraction function is discovered automatically, through a partition refinement algorithm that, at each iteration, computes and checks an invariant of a specific form that corresponds to the GFB conditions (roughly, that the current abstraction candidate is a sound one). This could be seen as a sort of counterexample-guided abstraction refinement process. The class of abstraction functions considered, however, is not arbitrary: only monoidal operations are considered, which yields two main advantages: efficiency in the partition refinement algorithm, and meaning for the class of application domains considered (e.g. full/partial activation in biological models).

Reduction techniques exist for BNs. Boolean backward equivalence (BBE) is a backward-type bisimulation [21], in line exact lumpability for Markov chains [4] and BDE. Hence, it can be shown that BBE and GFB (applied to BNs) are not comparable. Other approaches for BN reduction are based on *variable absorption* (e.g., [43], [44]) 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.

III. PRELIMINARIES

In this section we formalize the notion of discrete-time DS and of attractor for discrete-time DS [45], and notation considered in this paper. Then, we provide a running example used throughout the text.

Definition 1 (Dynamical System). *A discrete-time DS is a pair $D = (X, F)$ where $X = \{x_1, \ldots, x_n\}$ are 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 (simulation) of D for 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

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.

As running example we use a BN from [46] that describes cell differentiation. Deeper biological interpretation and its reduction will be given in Section VII.

Example 1. *Let (X, F) be a discrete-time DS with Boolean variables $X = \{$SCR, SHR, JKD, MGP, WOX5, CLEX, PLT, ARF, AUXIAA, AUXIN$\}$ and update function $F : \mathbb{B}^X \to \mathbb{B}^X$ with*

$$f_{\mathsf{SCR}} = \mathsf{SHR} \wedge \mathsf{SCR} \wedge (\mathsf{JKD} \vee \neg \mathsf{MGP}) \qquad f_{\mathsf{CLEX}} = \mathsf{SHR} \wedge \mathsf{CLEX}$$
$$f_{\mathsf{SHR}} = \mathsf{SHR} \qquad\qquad\qquad\qquad f_{\mathsf{PLT}} = \mathsf{ARF}$$
$$f_{\mathsf{JKD}} = \mathsf{SHR} \wedge \mathsf{SCR} \qquad\qquad\qquad f_{\mathsf{ARF}} = \neg \mathsf{AUXIAA}$$
$$f_{\mathsf{MGP}} = \mathsf{SHR} \wedge \mathsf{SCR} \wedge \neg \mathsf{WOX5} \qquad f_{\mathsf{AUXIAA}} = \neg \mathsf{AUXIN}$$
$$f_{\mathsf{WOX5}} = \mathsf{ARF} \wedge \mathsf{SHR} \wedge \mathsf{SCR} \wedge \neg \mathsf{CLEX} \qquad f_{\mathsf{AUXIN}} = \mathsf{AUXIN}$$

Monoids for the DS are $(\mathbb{B}, \oplus)$, $\oplus \in \{\wedge, \vee, XOR\}$, with neutral elements $1, 0, 0$.

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 (wrt 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 s.t. 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 only the variables CLEX, PLT, ARF, AUXIN are active. 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. 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.*

IV. GENERALIZED FORWARD BISIMULATION

Here 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:*

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

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. *For* $\oplus = \wedge$, $\mathcal{X}_R = \{C, \{\text{PLT}\}, \{\text{ARF}\}, \{\text{AUXIAA}\}, \{\text{AUXIN}\}\}$ *is a GFB for our running example, where* $C = \{\text{SCR}, \text{SHR}, \text{JKD}, \text{MGP}, \text{WOX5}, \text{CLEX}\}$. *This means that the running example can be rewritten solely in terms of conjunctions over all variables in* C, *and the other 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, the update functions of* WOX5 *and* CLEX *contain terms* $\neg\text{CLEX}$ *and* CLEX, *respectively, therefore the conjunction of their update functions (and of all variables in* C*) can be simply rewritten as* 0 *since:*
$$\bigwedge_{x_i \in C} f_{x_i}(s) = s_{\text{CLEX}} \wedge \neg s_{\text{CLEX}} \wedge (\dots) = 0.$$

Definition 4 (Reduced DS). *The reduction* D/R *of a discrete-time DS* $D = (X, F)$ *for an equivalence* 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 representative for the other (singleton) blocks is obvious. With this, we obtain*
$$f_C = \bigwedge_{x_k \in C} f_{x_k}[x_k/1 : x_k \notin \hat{X}][x_{i_{C'}}/x_{C'} : C' \in \mathcal{X}_R]$$
$$= 1 \wedge 1 \wedge (C \vee \neg 1) \wedge 1 \wedge 1 \wedge 1 \wedge 1$$
$$\wedge 1 \wedge \neg 1 \wedge \{\text{ARF}\} \wedge 1 \wedge 1 \wedge \neg 1 \wedge 1 \wedge 1 = 0$$

For all other blocks, instead, we obtain
$$f_{\{\text{PLT}\}} = \{\text{ARF}\}, \qquad f_{\{\text{ARF}\}} = \neg\{\text{AUXIAA}\},$$
$$f_{\{\text{AUXIAA}\}} = \neg\{\text{AUXIN}\}, \qquad f_{\{\text{AUXIN}\}} = \{\text{AUXIN}\}$$

Remark 1. *We note that, syntactically, the reduced DS depends on the choice of representatives. However, if* R *is a GFB, then Theorem 1 guarantees that such choice does not affect the semantics of the reduced DS.*

We now show that D and D/R have *same dynamics up to* ψ_R iff R is a GFB.

Theorem 1 (GFB characterization via model dynamics). *Fix a DS* $D = (X, F)$, *a partition* $\mathcal{X}_R$ *of* X, $D/R = (\mathcal{X}_R, F_R)$, *and a commutative monoid* $(\mathbb{M}, \oplus)$. *Then,* R *is a GFB iff 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. *That is:*
$$\hat{s}_k = \psi_R(s_k), \quad \text{for } k \geq 0,$$
where $s_{k+1} = F(s_k)$ *and* $\hat{s}_{k+1} = F_R(\hat{s}_k)$.

Proof of Theorem 1. 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

[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.

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. $\qquad\square$

Theorem 1 readily implies the following result on attractors.

Corollary 1. *Let* $D = (X, F)$ *be a DS,* $(\mathbb{M}, \oplus)$ *a commutative monoid,* R *a GFB and* $D/R = (\mathcal{X}_R, F_R)$. *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* $s' = \{(0, 0, 0, 0, 0, 0, 1, 1, 0, 1)\}$ *from Example 2. The homomorphism* ψ_R *maps the attractor to* $\psi_R(s') = \{(0, 1, 1, 0, 1)\}$. *Corollary 1 ensures that the set* $\psi_R(s')$ *is an attractor of the reduced system* D/R. *Indeed, by applying the update functions* F_R *to* $(0, 1, 1, 0, 1)$, *the reduced system remains at the same state, and thus* $\psi_R(s')$ *is invariant under* F_R.

V. COMPUTATION OF THE COARSEST GFB

Computing the coarsest GFB that refines a given initial partition is based on the classic partition refinement algorithm [9] where the blocks of an initial partition are iteratively refined (or split) until a GFB is obtained. The coarsest GFB is obtained when the initial partition contains one block only. Different initial partitions can be useful to tune reductions to preserve variables of interest (see, e.g., Section VII). Here we prove that there exists a unique coarsest GFB that refines a given initial partition, and that the algorithm computes it.

Theorem 2. *Let* $D = (X, F)$ *be a discrete-time DS, and* $\mathcal{X}_R$ *a partition of* X. *There exists a unique coarsest GFB* $\mathcal{H}$ *that refines* $\mathcal{X}_R$.

Proof of Theorem 2. Fix arbitrary GFBs $\sim_1, \dots, \sim_\nu \subseteq R$ and let $\mathcal{H}_1, \dots, \mathcal{H}_\nu$ 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 $x_0, \dots, x_k \in X$ and $i_0, \dots, i_{k-1} \in \{1, \dots, \nu\}$ so that $\{x_0, \dots, x_k\} = H^*$, $x_k = x_{i_{H^*}}$ 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) $G_1^i, \dots, G_{m_i}^i \in \mathcal{H}_i$ such that

$\biguplus_{l=1}^{m_i} G_l^i = G^*$. Since $x_j \sim_{i_j} x_{j+1}$ and $\mathcal{H}_{i_j}$ is a GFB, we obtain

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

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

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

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

A partition refinement algorithm for computing GFB needs a condition 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 3 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 3 (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 $x_i \ne x_j$, the following formula holds (where $0_\oplus$ is the neutral element of $\oplus$):*

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

Proof of Theorem 3. 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 $x_l \ne x_k$. Then, since $\oplus$ is commutative and associative and because $\mathcal{X}_R$ is a GFB, we have that

$$
\bigwedge_{C \in \mathcal{X}_R} \left(\bigoplus_{x_i \in C} f_{x_i}(s) = \bigoplus_{x_i \in C} f_{x_i}(s') \right). \tag{3}
$$

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

$$
\bigwedge_{C \in \mathcal{X}_R} \left(\bigoplus_{x_i \in C} s_{x_i} = \bigoplus_{x_i \in C} s'_{x_i} \right) \tag{4}
$$

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 $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ for which $\hat{s} = (((s_{i_1 \to i_2})_{i_2 \to i_3}) \cdots)_{i_{k-1} \to i_k}$ is such that

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

$\hat{s}_{x_i} = 0_\oplus$ 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 $\Psi_{x_{i_l}, x_{i_{l+1}}}^{\mathcal{X}_R}$ is valid for all $1 \le l \le k-1$, we obtain

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

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

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

Thanks to (4), we infer that $\hat{s} = \hat{s}'$. This, in turn, implies the desired relation (3), thus showing that $\mathcal{X}_R$ is a GFB if and only if $\Psi_{x_i, x_j}^{\mathcal{X}_R}$ is valid for all $(x_i, x_j) \in R$. $\qquad\square$

The binary characterization tells us that we can rewrite an $\oplus$-expression of the update functions of a block of a GFB 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.

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

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

For any other block the clause is trivially true because SHR *and* JKD *appear only in the update functions of variables in C. Hence, $\Psi_{\text{SHR}, \text{JKD}}^{\mathcal{X}_R}$ is valid. Similarly, we can show that $\Psi_{x_i, x_j}^{\mathcal{X}_R}$ is valid for all $(x_i, x_j) \in R, x_i \ne x_j$. Hence $\mathcal{X}_R$ is a GFB.*

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

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

1: **while true do**
2:     $\mathcal{H}' \leftarrow \emptyset$
3:     **for all** $H \in \mathcal{H}$ **do**
4:         $R \leftarrow \{(x_i, x_j) \in H \times H : \text{if } x_i \neq x_j, \text{ then } \Psi^{\mathcal{H}}_{x_i, x_j} \text{ and } \Psi^{\mathcal{H}}_{x_j, x_i}\}$
5:         $\mathcal{H}' \leftarrow \mathcal{H}' \cup (H/R)$
6:     **end for**
7:     **if** $\mathcal{H} = \mathcal{H}'$ **then**
8:         **return** $\mathcal{H}$
9:     **else**
10:        $\mathcal{H} \leftarrow \mathcal{H}'$
11:     **end if**
12: **end while**

Theorem 4. *Let $D = (X, F)$ be a discrete-time DS and X_R a partition. Algorithm 1 computes the coarsest 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.*

Proof of Theorem 4. Pick the coarsest GFB $\mathcal{H}_*$ that refines X_R using Theorem 2. With this, set $\mathcal{H}_0 := X_R$ and define for all $k \geq 0$ 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 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

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

amounts to a finite number of checks over finite sets and is thus decidable. $\qquad \square$

The decidability of $\Psi^{\mathcal{H}}_{x_i, x_j}$ for $\mathbb{M}$ 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 hinges 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 [8]. If also exponential and trigonometric functions are allowed, the problem becomes undecidable [47].

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 [48].[2] Fortunately, the following result readily follows from the Schwartz-Zippel lemma [48].

Theorem 5. *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*

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

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.

---

[2]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.

VI. CONTINUOUS-TIME DS

We relate GFB to continuous-time DS, showing how GFB encapsulates existing bisimulations for (nonlinear) ODEs. Thus, in what follows we consider DS with domain $\mathbb{R}$. We 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 ODEs and its time discretization vanishes if τ approaches zero [49].

A. Exact lumpability

GFB-type reductions can be captured by exact lumpability, an established reduction notion for ODEs [26], [27]. Indeed, exact lumping must not be necessarily induced by a partition of the 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 if and only if it is a GFB for its discretization, provided higher-order discretization errors are ignored. We start with the definition of exact lumping [26].

Definition 5. *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 $t \in [0; T]$.*

Consider, e.g., the model

$$\partial_t v_{x_1} = v_{x_1}$$
$$\partial_t v_{x_2} = v_{x_2}.$$

Then, $\psi(v_{x_1}, v_{x_2}) = v_{x_1} v_{x_2}$ is an exact lumping since

$$\begin{aligned}
\partial_t \psi(v) &= (\partial_{x_1} \psi(v), \partial_{x_2} \psi(v)) \cdot \Phi(v) \\
&= (v_{x_2}, v_{x_1}) \cdot (\partial_t v_{x_1}, \partial_t v_{x_2})^T \\
&= 2 v_{x_1} v_{x_2} = 2\psi(v)
\end{aligned}$$

where superscript T denotes the transpose of a vector. We can observe that this can be discovered using GFB on the time discretization of the ODE system, given by

$$f_{x_1}(s) = s_{x_1} + \tau s_{x_1} \quad \text{and} \quad f_{x_2}(s) = s_{x_2} + \tau s_{x_2}.$$

Indeed $\mathcal{X}_R = \{\{x_1, x_2\}\}$ is a GFB over $(\mathbb{R}, \cdot)$ since

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

This shows that ψ_R is indeed an exact lumping. The next result formalizes this relationship.

Theorem 6. *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable function $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, consider the DS $D_\tau = (X, F)$ 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 all D_τ, 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 all D_τ if and only if ψ_R is an exact lumping of $\partial_t v(t) = \Phi(v(t))$.*

Proof of Theorem 6. See proof of Theorem 7. $\qquad\square$

With the exception of the important case where ψ_R is linear, Theorem 6 does not address whether GFB is also a necessary condition for exact lumpability. Indeed, it turns out that a characterization requires to relax formula $\Psi^{\mathcal{X}_R}_{x_i, x_j}$ to, roughly speaking, ignore the terms of (higher) order $\tau^2, \tau^3, \dots$ and so on. This is exemplified next.

Example 7. *Consider $\partial_t v(t) = \Phi(v(t))$ where Φ is given by*

$$\partial_t v_{x_1} = v_{x_1} \log(v_{x_2}) \quad \text{and} \quad \partial_t v_{x_2} = v_{x_2} \log(v_{x_1})$$

when both right-hand sides are defined, and zero otherwise. Then, for monoid $(\mathbb{R}, \cdot)$ and $\mathcal{X}_R = \{\{x_1, x_2\}\}$, it holds that $\psi_R(v_{x_1}, v_{x_2}) = v_{x_1} \cdot v_{x_2}$ is an exact lumping, while $\mathcal{X}_R$ is not a GFB. In order to see this, we start by noting that

$$\begin{aligned}
\partial_t \psi_R(v) &= v_{x_2} \partial_t v_{x_1} + v_{x_1} \partial_t v_{x_2} \\
&= v_{x_1} v_{x_2} (\log(v_{x_2}) + \log(v_{x_1})) \\
&= \psi_R(v) \log(\psi_R(v))
\end{aligned}$$

At the same time, the ODE discretization of the model is

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

With this, we obtain

$$\begin{aligned}
f_{x_1} f_{x_2} &= (x_1 + \tau x_1 \log(x_2))(x_2 + \tau x_2 \log(x_1)) \\
&= \psi(v) + \tau \psi(v) \log(\psi(v)) + \tau^2 \psi(v) \log(x_1) \log(x_2)
\end{aligned}$$

Since the higher-order term $\tau^2 x_1 x_2 \log(x_1) \log(x_2)$ cannot be expressed in terms of $x_1 x_2$, we conclude that $\mathcal{X}_R$ is not a GFB.

We now characterize exact lumpings of the form ψ_R, accounting for Example 7 and generalizing Theorem 6. As anticipated, we 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 7. *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable vector field $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, consider the DS $D_\tau = (X, F)$ with $F(s) = s + \tau \Phi(s)$ where $\tau > 0$. 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, function ψ_R is an exact lumping iff 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*

$$\begin{aligned}
\bigwedge_{C \in \mathcal{X}_R} \Big(\bigoplus_{x_k \in C} f_{x_k} + \mathcal{O}(\tau^2) = \\
\bigoplus_{x_k \in C} f_{x_k}[x_i/0_\oplus][x_j/(x_i \oplus x_j)] + \mathcal{O}(\tau^2) \Big).
\end{aligned} \tag{5}$$

Proof of Theorem 7. To improve readability, we write ψ instead of ψ_R in the present proof. Since $\oplus$ is twice differen-

tiable 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 $\partial_t v(t) = \Phi(v(t))$, by [27] the derivative of $t \mapsto \psi_H(v(t))$ can be written as a function of $\big(\psi_H(v(t))\big)_{H \in \mathcal{X}_R}$. Since $v(0) \in \mathbb{R}^X$ can be chosen arbitrarily, there is thus a function $\wp_H$ such that $\wp_H(\psi(s)) = (\partial_s\psi_H)(s + \tau\Phi(s)) \cdot \Phi(s) + \mathcal{O}(\tau)$ for all $s \in \mathbb{R}^X$. Indeed, by [27], there exists an $\hat{f}$ such that $\partial_t\psi(v(t)) = \hat{f}(\psi(v(t)))$. At the same time, the chain rule yields $\partial_t\psi_H(v(t)) = (\partial_s\psi_H)(v(t)) \cdot \Phi(v(t))$. Setting $v(0) = s + \tau\Phi(s)$ for an arbitrary s, we thus get

$$\begin{aligned}
\hat{f}_H(\psi(s + \tau\Phi(s))) &= (\partial_s\psi_H)(v(0)) \cdot \Phi(v(0)) \\
&= (\partial_s\psi_H)(s + \tau\Phi(s)) \cdot \Phi(s + \tau\Phi(s)) \\
&= (\partial_s\psi_H)(s + \tau\Phi(s)) \cdot \Phi(s) + O(\tau)
\end{aligned}$$

With this, we can set $\wp_H = \hat{f}_H$ and 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 3, we infer that for all $(x_i, x_j) \in R$ with $x_i \neq x_j$ formula $\Psi_{x_i,x_j}^{\mathcal{X}_R}$ 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 $\Psi_{x_i,x_j}^{\mathcal{X}_R}$ 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 $\Psi_{x_i,x_j}^{\mathcal{X}_R}$, the argumentation from the proof of Theorem 3 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 $\partial_t v(t) = \Phi(v(t))$ and $v(0) = s + \tau\Phi(s)$, the chain rule implies

$$\begin{aligned}
\partial_t\psi_H(v(0)) &= (\partial_s\psi_H)(v(0)) \cdot \Phi(v(0)) \\
&= (\partial_s\psi_H)(s + \tau\Phi(s)) \cdot \Phi(s + \tau\Phi(s)) \\
&= (\partial_s\psi_H)(s + \tau\Phi(s)) \cdot \Phi(s) + O(\tau),
\end{aligned}$$

thus yielding $\tau \cdot \partial_t\psi_H(v(0)) = \tau \cdot \wp_H(\psi(v(0))) + O(\tau^2)$. With this, we obtain that ψ is an exact lumping. This completes the proof of Theorem 7. We next turn to the proofs of 1) and 2) of Theorem 6. For 1), we note that $\Psi_{x_i,x_j}^{\mathcal{X}_R}$ 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). $\square$

Theorem 6 is related to geometric integration where it has been shown [50, Section IV.1] that discrete-time approximations preserve invariants of continuous-time DS only when these are linear or quadratic, but not if they are cubic or of higher degree. In contrast, Theorem 7 provides a one-to-one correspondence between continuous- and discrete-time invariants by dropping the higher order terms. Additionally, Theorem 6 and 7 allow in contrast to [50] for the algorithmic computation of (nonlinear) invariants.

We end the subsection by noting that if $(f_{x_i})_{x_i \in X}$ are polynomials, then (5) can be checked algorithmically by representing polynomials as sums of monomials and by dropping afterwards all monomials containing a term τ^ν with $\nu \geq 2$.

B. Forward differential equivalence and Markov chains

Using the results of this section we can relate GFB with analogous bisimulations for DS. We start by restating the notion of forward differential equivalence (FDE) from [8].

Definition 6 (FDE). *Let us consider an ODE system $\partial_t v(t) = \Phi(v(t))$ with a differentiable function $\Phi : \mathbb{R}^X \to \mathbb{R}^X$. A partition $\mathcal{X}_R$ of X is an FDE if ψ_R in case of $\oplus = +$ is an exact lumpability.*

The next result follows from Theorem 6, relating GFB and FDE [8].

Corollary 2. *Given $\partial_t v(t) = \Phi(v(t))$ with a differentiable vector field $\Phi : \mathbb{R}^X \to \mathbb{R}^X$, and the DS $D_\tau = (X, F)$ with $F(s) = s + \tau\Phi(s)$, where $\tau > 0$. Then, for $(\mathbb{R}, +)$, we have that R is a GFB of all D_τ iff R is an FDE of $\partial_t v(t) = \Phi(v(t))$.*

Proof of Corollary 2. Set $\oplus = +$ in Theorem 6. $\square$

Similarly, the next corollary relates GFB with continuous-time Markov chains [4] and probabilistic bisimulation of discrete-time Markov chains [3].

Corollary 3. *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 $D_\tau = (X, F)$ with $F(s) = s + \tau Q^T s$ where $\tau > 0$. Then, D_τ 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 all D_τ; 2. R is an ordinary lumpability of (X, Q); 3. R is a probabilistic bisimulation of all D_τ that describe a discrete-time Markov chain.*

Proof of Corollary 3. 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 [8], 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 2 yields the equivalence of 1) and 2). The equivalence of 2) and 3), instead, is a well-known fact [4]. $\square$

Remark 2. *The discussion shows that $\Psi_{x_i,x_j}^{\mathcal{H}}$ in Algorithm 1 can be decided in polynomial time for probabilistic bisimulation and FDE of polynomial differential equations [8], [16], which can be in principle also extended to differential algebraic equations [51].*
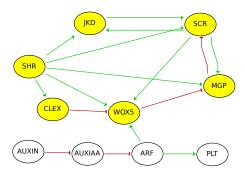
Fig. 1. Pictorial representation of the Boolean network from Example 1 using GinSim [53], adapted from [46].

Fig. 2. Pictorial representation using GinSim [53] adapted from [55] of the model on eggshell formation for drosophila melanogaster flies.

C. Attractors of continuous-time DS

The notion of attractor from Definition 2 also exists for continuous-time dynamics [52].

Definition 7 (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 [52] essentially ensures that attractors of an ODE system can be approximated by attractors of its discrete-time discretization.

Theorem 8 ([52]). *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 $\tau \to 0$.*

Corollary 1 and Theorem 8 allow to use GFB to argue on attractors of ODEs. Less importantly, Theorem 8 does not explicitly provide basins of attraction for the sets $A(\tau)$. However, $A(\tau)$ are attractors when the discrete topology is used.

VII. APPLICATIONS

A. Regulatory Networks

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

*a) **BN case study**:* We study the BN used as running example (Example 1). To ease interpretation, Fig. 1 uses the typical graphical notation of *influence graphs* (offered, e.g., by GinSIM [53]). Nodes denote variables, while arrows denote *influences* among nodes. Influences come from the update functions: green and red arrows denote, respectively, positive

(*promotion*) and negative (*inhibition*) influence. In Example 1, ARF promotes PLT due to term ARF in f_{PLT}, while AUXIN inhibits AUXIAA due to term ¬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 one involving the hormone AUXIN and its signaling to the plethora (PLT) genes.

BN variables can be categorized into three groups [54]: *inputs* (SHR, and AUXIN) that do not have incoming edges, *outputs* (PLT) that do not have outgoing edges, and the remaining *internal nodes*. The distinction is obvious from update functions: inputs have constant update functions, while outputs do not appear in the update function of other variables. Inputs are often set by the modeler to perform *what if* experiments, whereas outputs permit to observe the response dynamics of the model. In this BN, each input *controls* its own pathway, meaning that the modeller can decide to enable them via appropriate initial states.

Considering the GFB $\mathcal{X}_R$ from Example 3 for $\oplus = \wedge$, the only non-trivial block $C = \{$SCR, SHR, JKD, MGP, WOX5, CLEX$\}$ corresponds to the yellow nodes in Fig. 1. This GFB is computed using the initial partition with two blocks separating outputs and non-output nodes. Considering the reduced model for $\mathcal{X}_R$ from Example 6, all yellow nodes in Fig. 1 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 the AUXIN pathway only.

*b) **Multi-valued network case study**:* We apply GFB to a multi-valued regulatory network (MV) from [55]. Intuitively, an MV is a BN where variables can admit more than two values. This is a single-cell model describing the development of eggshell structures in drosophila melanogaster flies. The MV has 7 variables with relations depicted in Fig. 2 and update functions:

$$f_{\mathsf{EGF}} = \mathsf{EGF}$$
$$f_{\mathsf{BMP}} = \mathsf{BMP}$$
$$f_{\mathsf{Ant}} = \mathsf{Ant}$$
$$f_{\mathsf{RoofAdj}} = \mathsf{RoofAdj}$$
$$f_{\mathsf{Roof}} = \mathsf{Ant}\!:\!1 \wedge \mathsf{EGF}\!:\!1 \wedge \mathsf{BMP}\!:\!0$$
$$f_{\mathsf{Floor}} = \mathsf{Ant}\!:\!1 \wedge (\mathsf{EGF}\!:\!2 \vee (\mathsf{EGF}\!:\!1 \wedge \mathsf{BMP}\!:\!1)) \wedge \mathsf{RoofAdj}\!:\!1$$
$$f_{\mathsf{Operc}} = \mathsf{Ant}\!:\!1 \wedge (\mathsf{EGF}\!:\!2 \vee (\mathsf{EGF}\!:\!1 \wedge \mathsf{BMP}\!:\!1)) \wedge \mathsf{RoofAdj}\!:\!0$$

Using the notation in [55], "$\mathsf{var} : v$" 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, the rectangular node in Fig. 2, can take values 0, 1, 2, denoting absent/intermediate/high activation levels. All other variables are Boolean (0/1).[3]

Differently from Fig. 1, variables divide in two groups only: the *inputs* EGF, BMP, Ant, and $\mathsf{RoofAdj}$, and the *outputs* Operc, Floor, and Roof. We also have a third edge type, the purple one from EGF to Roof. This visually stresses that EGF influences Roof only when in intermediate level and not when in high level.

The MV relates three follicle cell fates, the outputs, to combinations of values of the inputs. EGF and BMP are known pathways responsible for patterning of the drosophila eggshell [55]. This is encoded in the model because EGF and BMP influence, in different ways, all outputs. Finally, Ant models the anterior competence region, therefore it is required by all outputs, while $\mathsf{RoofAdj}$ accounts for the state of neighboring cells by promoting Floor and inhibiting Operc (*operculum*).

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

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

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, $\mathsf{RoofAdj}$, 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

---

[3]Our framework requires all variables to have same domain $\mathbb{M}$. In order to support MV, 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 we will remain within the original domain.

three values for EGF, as we are only interested in the cases in which it is 0 or positive ($\mathsf{EGF}\!:\!1 \vee \mathsf{EGF}\!:\!2$). 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 have all value 1. Indeed, by studying the update functions of the original outputs, we see that there are no values for the involved variables that makes all of them true.

c) 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 (ginsim.org/models_repository) and BioModelsDB [23]. [4] We validate GFB in terms of aggregation power and of speed-up offered for attractor analysis.

Experimental setting. We created a prototype implementation of GFB integrated with the SMT solver Z3 [57] to check formulas from $\Psi^{\mathcal{H}}_{x_i,x_j}$ in Algorithm 1 in the tool for model reduction ERODE [19], which has been recently extended with support for BNs [58]. In doing so, we added to ERODE an importer for SBML Qual [59], an XML format supported by both repositories, allowing us to import all 43 BNs and 50 MVs. In order to obtain physically-relevant initial partitions, we infer *candidate outputs*, variables not appearing in the update function of other variables. For each model, we create *output-preserving* initial partitions: these consist of two blocks, one containing all outputs and one containing the remaining variables. This guarantees that reduced models allow, e.g., for full output (de)activation studies discussed before. In order to perform a consistent treatment, we restricted our analysis on the 29 BNs and 31 MVs with at least one candidate output.

Validation of aggregation power. Fig. 3 (Left) provides the reduction ratios obtained for the BNs using $\oplus \in \{\wedge, \vee\}$. For each model we plot the reduction ratio, defined 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 fourth of the models having reduction ratio below 0.6, while for $\oplus = \vee$ most of the models have 0.8 or more. For $\oplus = \wedge$, some models have particularly low ratios, below 0.2, some of which due to the fact that the reduced model has 2 variables only. We remark that these shall not be considered *degenerate* reductions, because of the used initial partitions, as discussed. We do not present results on maximal reductions, obtained with the initial partition with one block only. These are significantly smaller, but some are degenerate with one variable only. We leave for future work a detailed study on finer intermediate reductions using model-specific initial partitions preserving variables of interest for the modeler. For example, a modeler could be interested in preserving only some outputs. Fig. 3 (Right) presents a similar study performed on the MVs using $\oplus = \min$ and $\oplus = \max$, confirming the aggregation power of GFB.

Validation of analysis speed-up. Corollary 1 ensures that

---

[4]BioModelsDB contains both BN and ODE models. Here we focus on BNs, while ODEs were considered in [56] for ODE-based reduction techniques.
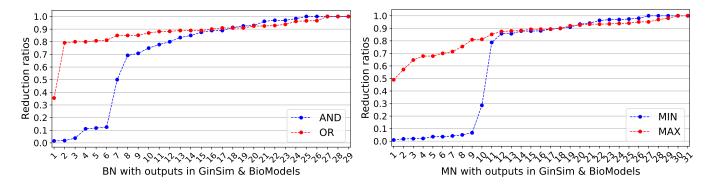
Fig. 3. **(Left)** Reduction ratios (reduced variables over original ones) in ascending order for the 29 BN with outputs from GINsim and BioModelsDB for $\oplus \in \{\wedge, \vee\}$ and initial partitions with two blocks separating output and non-outputs. **(Right)** Same as (Left) for the 31 MV with outputs from the two repositories using $\oplus \in \{\min, \max\}$.
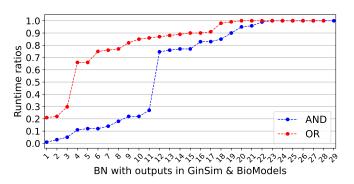

Fig. 4. Runtime ratios in ascending order for computation of attractors for the 29 BNs from Fig. 3 (Left) and their reductions.

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 [60], an environment incorporating a variety of tools for BN analysis. An example is BNS [61], which combines SAT-solving and bounded model checking to identify attractors. We computed the attractors of the 29 considered BNs and of their reductions. We could not consider MVs because we are not aware of tools for general attractor analysis for MVs. Fig. 4 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 Fig. 3 (Left) with ratio smaller than 0.3 take less than 1.5 seconds on average.

*d) **Enabling analysis of large BNs using GFB:*** We now apply GFB to a large BN of signalling pathways central to macrophage activation [62]. This BN contains *321* variables, making attractor computation infeasible even using the most efficient tool for this task [61]. In particular, the analysis does not terminate within an arbitrarily chosen time limit of 10 hours. Our crucial hypothesis is that *GFB can enable some*

analysis of this otherwise not analyzable BN, although with certain restrictions imposed by what is exactly preserved by the reduction.

The results are presented in Table I. In this experiment we focus on $\oplus = \wedge$. We can see that the maximal reduction is not physically-relevant, as it reduces to 1 variable only. The output-preserving reduction, instead, leads to a reduced model with 189 variables. Despite this, the obtained reduced model is still not analyzable within the chosen time limit. We now show how two alternative initial partitions lead to reduced models that can be effectively analyzed. In particular, we assume that the modeler is not interested in preserving all 68 outputs, but two different subsets of them: $O_1 = \{\texttt{s\_28}, \texttt{s\_26}, \texttt{s\_198}, \texttt{s\_11}\}$ and $O_2 = \{\texttt{s\_184}, \texttt{s\_188}\}$. In both cases, we use an initial partition with one block for the selected outputs, and one for all the other variables. In these two cases, we obtained models with 70 and 33 variables, respectively, which admit analysis. In particular, the obtained reduced models can now be analyzed using less than a second.

B. Non-linear reductions of Differential and Difference Equations

We present examples of exact lumping where ψ_R is not linear, and thus cannot be captured by linear lumpings such FDE. We use $(\mathbb{R}, \cdot)$ with neutral element 1.

*a) **Nonlinear Reduction of a Lotka-Volterra Model over*** $(\mathbb{R}, \cdot)$: We start considering a prototypical higher-order Lotka-Volterra model [24] where x_1 preys x_2 and x_3, while x_2 and

| Model | Variables | Attractors analysis | |
		Count	Runtime(s)
Original	321	—Time Out—	
Output separated	189	—Time Out—	
O1	70	64	0.668
O2	33	64	0.325
Maximal	1	1	0.001

TABLE I
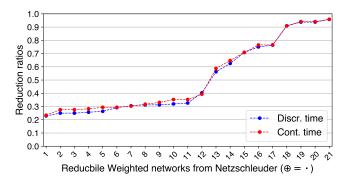GFB ENABLES ATTRACTORS COMPUTATION ON LARGE BN [62].

Fig. 5. Similarly to Fig. 3, we plot reduction ratios for the discrete- and continuous-time dynamical interpretations of 21 weighted networks from Netzschleuder. We used one operator, $\oplus = \cdot$, and initial partitions separating the first node in the network from the others.

x_3 prey together x_1. The corresponding ODE system is

$$\partial_t v_{x_1} = v_{x_1}(1 - v_{x_2}v_{x_3}),$$
$$\partial_t v_{x_2} = v_{x_2}(1 - v_{x_1}), \qquad (6)$$
$$\partial_t v_{x_3} = v_{x_3}(1 - v_{x_1}).$$

The ODE discretization of (6) is given by

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

By Theorem 6, 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 (6). Indeed, $\mathcal{X}_R = \{\{x_1\}, \{x_2, x_3\}\}$ is a GFB of (6) for $\oplus = \cdot$ because $\Psi_{x_2,x_3}^{\mathcal{X}_R}$ is valid thanks to the identities $f_{x_1} = f_{x_1}[x_2/1, x_3/x_2x_3]$, and

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

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

$$\partial_t(v_{x_2}v_{x_3}) = \partial_t v_{x_2} \cdot v_{x_3} + v_{x_2} \cdot \partial_t v_{x_3}$$
$$= v_{x_2}(1 - v_{x_1})v_{x_3} + v_{x_2}v_{x_3}(1 - v_{x_1})$$
$$= 2v_{x_1}v_{x_2}(1 - v_{x_1}).$$

b) Nonlinear Reduction of Dynamical Weighted Networks over $(\mathbb{R}, \cdot)$: We now consider real-valued DS obtained from weighted networks from the Netzschleuder repository [25]. We considered all 72 weighted networks with at most 200 nodes (by restricting to at most the first 15 models from each family of models). The undirected ones were expanded in directed by replacing every undirected edge with two corresponding directed ones with same weight.

We consider two different dynamical interpretations. For A the adjacency matrix of a network, we study the discrete-time DS $x(t + 1) = Ax(t)$, and the (ODE discretization of the) continuous-time DS $\partial_t v(t) = Av(t)$. In both cases,

we use one variable per node.[5] We use $\oplus = \cdot$, obtaining nonlinear reductions. This leads to high nonlinearities in formulas $\Psi_{x_i,x_j}^{\mathcal{X}_R}$ from Theorem 3, complex to handle for Z3. Indeed, Algorithm 1 failed to terminate within an arbitrarily chosen time-out of 1 hour even for models of moderate size. Hence, for our experiments we used the randomized version of the algorithm discussed in Section V, performing 40 tests per formula $\Psi_{x_i,x_j}^{\mathcal{X}_R}$ from Theorem 3 after sampling values for all variables. Currently, our prototype is still based on Z3, to which we provide the sampled values making all $\Psi_{x_i,x_j}^{\mathcal{X}_R}$ formulas variable free. In this setting, Z3 never failed.

Fig. 5 provides the results for the 21 networks that admitted a reduction, 29% of the 72 considered. We got similar reduction ratios in the two interpretations, with slightly better ones for the discrete-time one. The lower reduction power of the continuous-time case comes from two factors: (i) Models have higher nonlinearities due to the τ term; (ii) Theorem 6 gives only a necessary condition for aggregation in this case (our prototype does not support the results of Theorem 7). The largest runtimes for the continuous- and discrete-time cases were about 500 and 400 seconds, respectively, for a model with 145 nodes.

VIII. CONCLUSION

Generalized forward bisimulation (GFB) is a technique for dimensionality reduction of discrete- and continuous-time dynamical systems that captures and generalizes existing techniques. GFB allows to compute nonlinear reductions. One needs to specify a dynamical system, a commutative monoid (the variables' domain and an operation used to aggregate them), and an initial partition of the variables (used to tune the reduction power to preserve variables of interest). A partition refinement algorithm then minimizes the system over the operation of the monoid. We implemented GFB and applied it to four popular formalisms: difference and differential equations with monoid $(\mathbb{R}, \cdot)$, Boolean networks with $(\mathbb{B}, \wedge)$ and $(\mathbb{B}, \vee)$, multi-valued networks with $(\{0, 1, 2\}, \min)$ and $(\{0, 1, 2\}, \max)$. In all cases, GFB yielded notable nonlinear reductions. On 60 Boolean and multi-valued networks from two popular repositories, we have shown high aggregation power and analysis speed-ups. Using an existing large Boolean network with 321 variables we have shown that GFB might enable the analysis of otherwise untractable models. On 21 ODEs originated from weighted networks from a popular repository, we have computed nonlinear reductions thanks to the $\cdot$ operation, showing high aggregation power. Future work will study the reduction of optimization problems from systems biology [63], performance engineering [64] and AI [65].

---

[5]In the continuous-time case, we also have an additional variable for the τ term from the ODE discretization, to which we give constant update function. This guarantees that the obtained reductions hold for any value of τ.

REFERENCES

[1] D. Sangiorgi, *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.

[2] D. Park, "Concurrency and automata on infinite sequences," in *Theoretical Computer Science*, 1981, pp. 167–183.

[3] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *Inf. Comput.*, vol. 94, no. 1, pp. 1–28, 1991.

[4] P. Buchholz, "Exact and ordinary lumpability in finite Markov chains," *Journal of Applied Probability*, vol. 31, no. 1, pp. 59–75, 1994.

[5] R. De Nicola, U. Montanari, and F. Vaandrager, "Back and forth bisimulations," in *CONCUR '90 Theories of Concurrency: Unification and Extension*, ser. Lecture Notes in Computer Science, J. Baeten and J. Klop, Eds. Springer Berlin Heidelberg, 1990, vol. 458, pp. 152–165.

[6] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Forward and backward bisimulations for chemical reaction networks," in *CONCUR*, 2015, pp. 226–239.

[7] L. Cardelli, I. C. Pérez-Verona, M. Tribastone, M. Tschaikowski, A. Vandin, and T. Waizmann, "Exact maximal reduction of stochastic reaction networks by species lumping," *Bioinform.*, vol. 37, no. 15, pp. 2175–2182, 2021.

[8] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Symbolic computation of differential equivalences," in *POPL*, 2016, pp. 137–150.

[9] R. Paige and R. E. Tarjan, "Three partition refinement algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987.

[10] M. S. Okino and M. L. Mavrovouniotis, "Simplification of mathematical models of chemical reaction systems," *Chemical Reviews*, vol. 2, no. 98, pp. 391–408, 1998.

[11] M. Tribastone, "Behavioral relations in a process algebra for variants," in *SPLC*, S. Gnesi, A. Fantechi, P. Heymans, J. Rubin, K. Czarnecki, and D. Dhungana, Eds. ACM, 2014, pp. 82–91.

[12] A. Antoulas, *Approximation of Large-Scale Dynamical Systems*, ser. Advances in Design and Control. SIAM, 2005.

[13] T. J. Snowden, P. H. van der Graaf, and M. J. Tindall, "Methods of model reduction for large-scale biological systems: A survey of current methods and trends," *Bulletin of Mathematical Biology*, vol. 79, no. 7, pp. 1449–1486, 2017.

[14] G. Li, H. Rabitz, and J. Tóth, "A general analysis of exact nonlinear lumping in chemical kinetics," *Chemical Engineering Science*, vol. 49, no. 3, pp. 343–361, 1994.

[15] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437 – 467, 1969.

[16] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Maximal aggregation of polynomial dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 114, no. 38, pp. 10029 – 10034, 2017.

[17] M. Hopfensitz, C. Müssel, M. Maucher, and H. A. Kestler, "Attractors in boolean networks: a tutorial," *Computational Statistics*, vol. 28, no. 1, pp. 19–36, 2013.

[18] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Guaranteed error bounds on approximate model abstractions through reachability analysis," in *QEST*, 2018, pp. 104–121.

[19] ——, "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*. Springer, 2017, pp. 310–328.

[20] R. Thomas, D. Thieffry, and M. 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*, vol. 57, no. 2, pp. 247–276, 1995.

[21] G. Argyris, A. Lluch Lafuente, M. Tribastone, M. Tschaikowski, and A. Vandin, "Reducing boolean networks with backward boolean equivalence," in *International Conference on Computational Methods in Systems Biology*. Springer, 2021, pp. 1–18.

[22] A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya, "Logical modelling of regulatory networks with ginsim 2.3," *Biosystems*, vol. 97, no. 2, pp. 134–139, 2009.

[23] R. S. Malik-Sheriff, M. Glont, T. V. N. Nguyen, K. Tiwari, M. G. Roberts, A. Xavier, M. T. Vu, J. Men, M. Maire, S. Kananathan, E. L. Fairbanks, J. P. Meyer, C. Arankalle, T. M. Varusai, V. Knight-Schrijver, L. Li, C. Dueñas-Roca, G. Dass, S. M. Keating, Y. M. Park, N. Buso, N. Rodriguez, M. Hucka, and H. Hermjakob, "BioModels — 15 years of sharing computational models in life science," *Nucleic Acids Research*, vol. 48, no. D1, pp. D407–D415, 1 2020, gkz1055. [Online]. Available: https://doi.org/10.1093/nar/gkz1055

[24] P. Singh and G. Baruah, "Higher order interactions and species coexistence," *Theoretical Ecology*, vol. 14, pp. 71–83, 2021.

[25] T. P. Peixoto, "The netzschleuder network catalogue and repository," 2020. [Online]. Available: https://networks.skewed.de/

[26] G. Li and H. Rabitz, "A general analysis of exact lumping in chemical kinetics," *Chemical Engineering Science*, vol. 44, no. 6, pp. 1413–1430, 1989.

[27] A. S. Tomlin, G. Li, H. Rabitz, and J. Tóth, "The effect of lumping and expanding on kinetic differential equations," *SIAM Journal on Applied Mathematics*, vol. 57, no. 6, pp. 1531–1556, 1997.

[28] L. Cardelli, I. C. Pérez-Verona, M. Tribastone, M. Tschaikowski, A. Vandin, and T. Waizmann, "Exact maximal reduction of stochastic reaction networks by species lumping," *Bioinform.*, vol. 37, no. 15, pp. 2175–2182, 2021.

[29] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Syntactic markovian bisimulation for chemical reaction networks," in *Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, L. Aceto, G. Bacci, G. Bacci, A. Ingólfsdóttir, A. Legay, and R. Mardare, Eds., vol. 10460, 2017, pp. 466–483.

[30] K. Ghorbal and A. Platzer, "Characterizing algebraic invariants by differential radical invariants," in *TACAS*, E. Ábrahám and K. Havelund, Eds., vol. 8413. Springer, 2014, pp. 279–294.

[31] E. Bartocci, L. Kovács, and M. Stankovic, "Automatic generation of moment-based invariants for prob-solvable loops," in *ATVA*, Y. Chen, C. Cheng, and J. Esparza, Eds., 2019, pp. 255–276.

[32] M. Boreale, "Algebra, coalgebra, and minimization in polynomial differential equations," *Log. Methods Comput. Sci.*, vol. 15, no. 1, 2019.

[33] ——, "Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial odes," *Sci. Comput. Program.*, vol. 193, p. 102441, 2020.

[34] M. Tschaikowski and M. Tribastone, "Exact fluid lumpability in markovian process algebra," *Theor. Comput. Sci.*, vol. 538, pp. 140–166, 2014.

[35] ——, "Tackling continuous state-space explosion in a markovian process algebra," *Theor. Comput. Sci.*, vol. 517, pp. 1–33, 2014.

[36] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Symbolic computation of differential equivalences," *Theor. Comput. Sci.*, vol. 777, pp. 132–154, 2019.

[37] G. Bacci, G. Bacci, K. G. Larsen, M. Tribastone, M. Tschaikowski, and A. Vandin, "Efficient local computation of differential bisimulations via coupling and up-to methods," in *Symposium on Logic in Computer Science, LICS*, 2021, pp. 1–14.

[38] G. J. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," *IEEE Trans. Automat. Contr.*, vol. 45, no. 6, pp. 1144–1160, 2000.

[39] G. J. Pappas and S. Simic, "Consistent abstractions of affine control systems," *IEEE Trans. Automat. Contr.*, vol. 47, no. 5, pp. 745–756, 2002.

[40] A. J. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE Transactions on Automatic Control*, vol. 49, 2004.

[41] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement," in *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, ser. Lecture Notes in Computer Science, E. A. Emerson and A. P. Sistla, Eds., vol. 1855. Springer, 2000, pp. 154–169. [Online]. Available: https://doi.org/10.1007/10722167_15

[42] S. Mover, A. Cimatti, A. Griggio, A. Irfan, and S. Tonetta, "Implicit semi-algebraic abstraction for polynomial dynamical systems," in *Computer Aided Verification - 33rd International Conference, CAV 2021, Virtual Event, July 20-23, 2021, Proceedings, Part I*, ser. Lecture Notes in Computer Science, A. Silva and K. R. M. Leino, Eds., vol. 12759. Springer, 2021, pp. 529–551. [Online]. Available: https://doi.org/10.1007/978-3-030-81685-8_25

[43] A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya, "Dynamically consistent reduction of logical regulatory graphs," *Theoretical Computer Science*, vol. 412, no. 21, pp. 2207–2218, 2011.

[44] A. Veliz-Cuba, "Reduction of boolean network models," *Journal of theoretical biology*, vol. 289, pp. 167–172, 2011.

[45] J. Milnor, "On the concept of attractor," *Communications in Mathematical Physics*, vol. 99, no. 2, pp. 177–195, 1985.

[46] E. Azpeitia, M. Benítez, I. Vega, C. Villarreal, and E. R. Alvarez-Buylla, "Single-cell and coupled grn models of cell patterning in the arabidopsis

thaliana root stem cell niche," *BMC systems biology*, vol. 4, no. 1, pp. 1–19, 2010.

[47] D. Richardson, "Some undecidable problems involving elementary functions of a real variable," *The Journal of Symbolic Logic*, vol. 33, no. 4, pp. 514–520, 1968.

[48] N. Saxena, "Progress on polynomial identity testing," *Bull. EATCS*, vol. 99, pp. 49–79, 2009.

[49] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1971.

[50] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, 2006.

[51] L. Cardelli, M. Tribastone, and M. Tschaikowski, "From electric circuits to chemical networks," *Nat. Comput.*, vol. 19, no. 1, pp. 237–248, 2020.

[52] P. E. Kloeden and J. Lorenz, "Stable attracting sets in dynamical systems and in their one-step discretizations," *SIAM Journal on Numerical Analysis*, vol. 23, no. 5, pp. 986–995, 1986.

[53] A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya, "Logical modelling of regulatory networks with ginsim 2.3," *Biosystems*, vol. 97, no. 2, pp. 134–139, 2009.

[54] A. Naldi, P. T. Monteiro, and C. Chaouiya, "Efficient handling of large signalling-regulatory networks by focusing on their core control," in *International Conference on Computational Methods in Systems Biology*. Springer, 2012, pp. 288–306.

[55] A. Fauré, B. Vreede, E. Sucena, and C. Chaouiya, "A discrete model of drosophila eggshell patterning reveals cell-autonomous and juxtacrine effects," *PLoS Comput Biol*, vol. 10, p. e1003527, 2014.

[56] I. C. Perez-Verona, M. Tribastone, and A. Vandin, "A large-scale assessment of exact lumping of quantitative models in the biomodels repository," *Theoretical Computer Science*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397521003716

[57] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[58] G. Argyris, A. Lluch-Lafuente, M. Tribastone, M. Tschaikowski, and A. Vandin, "An extension of ERODE to reduce boolean networks by backward boolean equivalence," in *International Conference on Computational Methods in Systems Biology*, ser. LNCS, vol. 13447. Springer, 2022, pp. 294–301.

[59] C. Chaouiya, D. Bérenguier, S. M. Keating, A. Naldi, M. P. Van Iersel, N. Rodriguez, A. Dräger, F. Büchel, T. Cokelaer, B. Kowal *et al.*, "SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools," *BMC systems biology*, vol. 7, no. 1, pp. 1–15, 2013.

[60] A. Naldi, P. T. Monteiro, C. Müssel, C. for Logical Models, Tools, H. A. Kestler, D. Thieffry, I. Xenarios, J. Saez-Rodriguez, T. Helikar, and C. Chaouiya, "Cooperative development of logical modelling standards and tools with colomoto," *Bioinformatics*, vol. 31, no. 7, pp. 1154–1159, 2015.

[61] E. Dubrova and M. Teslenko, "A sat-based algorithm for finding attractors in synchronous boolean networks," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 8, no. 5, pp. 1393–1399, 2011.

[62] S. Raza, K. A. Robertson, P. A. Lacaze, D. Page, A. J. Enright, P. Ghazal, and T. C. Freeman, "A logic-based diagram of signalling pathways central to macrophage activation," *BMC systems biology*, vol. 2, no. 1, pp. 1–15, 2008.

[63] M. Whitby, L. Cardelli, M. Kwiatkowska, L. Laurenti, M. Tribastone, and M. Tschaikowski, "PID control of biochemical reaction networks," *IEEE Trans. Autom. Control.*, vol. 67, no. 2, pp. 1023–1030, 2022.

[64] A. Das, G. V. Merrett, M. Tribastone, and B. M. Al-Hashimi, "Workload change point detection for runtime thermal management of embedded systems," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1358–1371, 2016.

[65] R. M. Hasani, M. Lechner, A. Amini, L. Liebenwein, M. Tschaikowski, G. Teschl, and D. Rus, "Closed-form continuous-depth models," *CoRR*, vol. abs/2106.13898, 2021. [Online]. Available: https://arxiv.org/abs/2106.13898