



HAL
open science

Abstract Simulation of Reaction Networks via Boolean Networks

Joachim Niehren, Athénaïs Vaginay, Cristian Versari

► **To cite this version:**

Joachim Niehren, Athénaïs Vaginay, Cristian Versari. Abstract Simulation of Reaction Networks via Boolean Networks. CMSB 2022 - 20th International Conference on Computational Methods in Systems Biology, Sep 2022, Bucarest, Romania. hal-02279942v9

HAL Id: hal-02279942

<https://hal.science/hal-02279942v9>

Submitted on 24 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Abstract Simulation of Reaction Networks via Boolean Networks

Joachim Niehren^{1,2}[0000-0002-2611-8950], Athénaïs
Vaginay^{3,4}[0000-0001-5062-7993], and Cristian Versari¹

¹ BioComputing Team of CRIStAL Lab, Université de Lille
² Inria Lille

³ Université de Lorraine, CNRS, CRAN, F-54000 Nancy,

⁴ Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy
{joachim.niehren,cristian.versari}@univ-lille.fr
athenais.vaginay@loria.fr

Abstract. We propose to simulate chemical reaction networks with the deterministic semantics abstractly, without any precise knowledge on the initial concentrations. For this, the concentrations of species are abstracted to Booleans stating whether the species is present or absent, and the derivatives of the concentrations are abstracted to signs saying whether the concentration is increasing, decreasing, or unchanged. We use abstract interpretation over the structure of signs for mapping the ODEs of a reaction network to a Boolean network with nondeterministic updates. The abstract state transition graph of such Boolean networks can be computed by finite domain constraint programming over the finite structure of signs. Constraints on the abstraction of the initial concentrations can be added naturally, leading to an abstract simulation algorithm that produces only the part of the abstract state transition graph that is reachable from the abstraction of the initial state. We prove the soundness of our abstract simulation algorithm, and show its applicability to reaction networks in the SBML format from the BioModels database.

Keywords: Systems Biology · Reaction Networks · SBML · Boolean Networks · Abstract Interpretation · Logic · Constraint Programming.

1 Introduction

Reaction networks [12,6,14,10] are the most prominent formalism for modeling the continuous dynamics of biological systems. Boolean networks [13,28,21,26] are the most prominent formalism for modeling discrete abstractions of the continuous dynamics. Hybrid automata [3] offer a framework for mixed continuous and discrete modeling. How to discretize the continuous semantics of reaction networks or of hybrid automata into Boolean networks is a long standing general question [4,7,27,15,16,5].

Reaction networks can be given different semantics. The continuous dynamics of a reaction network is given by its deterministic semantics: a system of

ordinary differential equations (ODEs), one per species, that is composed from the kinetic expressions of the reactions producing or consuming the species. The non-deterministic rewrite semantics [19], in contrast, ignores the kinetic expressions, while the stochastic semantics uses them differently, for computing the probability of a reaction to happen. Reaction networks also have a Boolean semantics which abstracts from the rewrite semantics [11]. In the present paper, we abstract the continuous semantics, which is notoriously difficult to compare to the rewrite semantics and thus to the Boolean semantics.

Any solution of the ODEs from the deterministic semantics provides a derivable function of type $\mathbb{R}_+ \rightarrow \mathbb{R}$ per species, that is called the trajectory. At any time point, the value of a trajectory must be positive, since it stands for the concentration of the species. In contrast, the value of its derivative may be negative, meaning that the concentration of the species is decreasing. The value of the trajectory at time point 0 is called the initial concentration of the species. It is well-known that for any fixed collection of initial concentrations per species, there exists at most one solution of the ODEs. This solution can be approximated numerically by using Euler’s deterministic simulation algorithm [9].

A concrete continuous state of a reaction network at a given time point is a vector of positive real numbers, one for the concentration of each species. Any concrete state can be abstracted to a vector of Booleans, stating for each species whether its concentration is zero or not. The possible trajectories of a reaction network can thus be abstracted to a state transition graph whose states are bit vectors. The graph can be enriched, when not only considering the trajectories but also their derivatives. Since these may become negative, the concrete states now become vectors of real numbers that can be abstracted to vectors of signs: increasing $\nearrow = 1$, decreasing $\searrow = -1$, and no-change $\rightarrow = 0$. In this way, we obtain an enriched abstract state transition graph between sign vectors.

In the present paper, we study an instance of the general discretization problem, which is whether one can compute the abstract state transition graph from a given reaction network. Clearly, abstract state transition graphs are finite, but may be large, since having $\Theta(2^{|\mathcal{S}|})$ many states where $|\mathcal{S}|$ is the number of species. Computing the abstract state transition graph completely quickly becomes impossible, given that the size grows exponentially in the number of species. Therefore, we propose to study the problem of abstract simulation, which is to compute the part of the abstract state transition graph that is accessible from the abstraction of the initial concentrations. This also has the advantage that the concrete initial concentrations do not need to be known precisely. Nevertheless, the problem remains non-trivial, given that trajectories are infinite objects, and that there are infinitely many trajectories depending on the choice of the initial concentrations.

Our idea for abstract simulation is based on the abstract interpretation of the system of ODEs of the reaction network over the structure of signs $\mathbb{S} = \{\nearrow, \rightarrow, \searrow\}$. This abstraction introduces non-determinism, since $\nearrow +^{\mathbb{S}} \searrow$ may be evaluated to any sign. It can be proven to provide a sound over-approximation based on John’s soundness theorem for the abstract interpretation of logic formulas [1,24,17]. We

show that the sign abstraction of the ODEs of a reaction network can be used to define a Boolean network with non-deterministic updates [25]. It will have rules stating that a species A is present in the next step, if A was already present at the previous step, or if the derivative of A was positive at the previous step. Since such rules can be defined by first-order (FO) formulas, we propose the notion of first-order Boolean networks with non-deterministic semantics (FO-BNNs).

We provide a soundness theorem for abstraction of reaction networks to FO-BNNs. It will rely on a causal next transition relation rather than on a temporal next transition relation inferred from the trajectories, given that concrete simulation algorithms too are based on causality. This may lead to approximation errors for the concrete numerical simulation, so we have to take care of approximation errors for abstract simulation too.

Given that FO-BNNs are first-order formulas that are to be interpreted over the finite structure of signs, we use finite domain constraint programming to compute the abstract state transition graphs of FO-BNNs. Constraints on the abstraction of the initial concentrations can be added naturally, leading to an algorithm for abstract simulation based on constraint programming. We have implemented this algorithm based on the Minizinc constraint solver [23].

While abstract interpretation enables qualitative reasoning, we can still support exact quantitative reasoning about thresholds. We show that whether $A \leq \epsilon$ for some threshold $\epsilon > 0$ can be tested by introducing an artificial species B so that $\dot{B} = A - \epsilon$. In this way, the sign of the derivative of B indicates, whether the concentration of A is above, below or equal ϵ . One can then use exact reasoning with linear equation systems [1] to improve the quality of our abstract simulation algorithm, while taking thresholds into account. For instance, we can show for the usual enzymatic reaction network that if the initial concentration of the substrate is above of a given threshold ϵ , then (1) the concentration of the product may eventually become bigger than ϵ , and (2) once this happens, it can never become smaller than ϵ again. Most interestingly, the precise initial concentration of the substrate does not matter for this argument, as long as it is above ϵ . In this way, abstract simulation can sometimes show properties of infinitely many concrete simulations.

We apply our abstract simulation algorithm to a reaction network represented in the Systems Biology Markup Language (SBML) [18] from the BioModels database [20]. We consider the model <https://www.ebi.ac.uk/biomodels/BIOMD0000000448> that we will call B448 for short. This network describes the insulin signalling in human adipocytes in normal conditions [8]. It has 27 species and 34 reactions, and its graph covers one full page (see Fig. 13). The abstract simulation algorithm successfully yields a very small subgraph of the abstract state transition graph with more than 2^{27} states.

Related Work on Exact Computation of State Transition Graphs.

Mover et al. [22] develop an efficient implicit method for the exact abstraction of dynamical systems, whose abstract state space description and ODE dynamics are restricted to be systems of polynomial equations. While polynomial equations allow the exact description of abstract state spaces that are more general and

fine-grained than the ones used for Boolean networks, they do not provide enough expressiveness to describe the kinetic expressions frequently used for the modeling of chemical reaction networks.

Outline. After a few preliminaries in Section 2, we discuss arithmetic expressions with three different interpretations in Section 3. We recall the notion of reaction networks and their deterministic semantics via ODEs in Section 4. In Section 5, we recall the first-order logic, which permits us to formally capture ODEs in Section 6, enables their abstract interpretation in Section 7, and lays the foundation of FO-BNNs in Section 8. We present our compiler from reaction networks to FO-BNNs and prove its soundness in Section 9. The treatment of thresholds is discussed in Section 10. It illustrates exact reasoning at the example of the enzymatic reaction network. The application of abstract simulation to reaction network B448 of the Biomodels database is shown in Section 11. The conclusion and future work are given in Section 12.

2 Preliminaries

The Cartesian product of sets A_1, \dots, A_n is denoted by $A_1 \times \dots \times A_n$. The domain of a partial function $f \subseteq A \times B$ is denoted by $\text{dom}(f)$. The restriction of f to a subset $A' \subseteq \text{dom}(f)$ is written as $f|_{A'}$. We write $[a_1/b_1, \dots, a_n/b_n]$ for the finite function f with $\text{dom}(f) = \{a_1, \dots, a_n\}$ and $f(a_i) = b_i$ for all $1 \leq i \leq n$. For any two sets A, B , the power set $B^A = A \rightarrow B = \{f \mid f : A \rightarrow B\}$ is the set of total functions from A to B . A multiset M with elements in A is an element $M \in \mathbb{N}^A$. For any $a \in A$, the multiplicity of a in M is $M(a)$.

Let $\mathbb{B} = \{0, 1\}$ be the set of Booleans, $\mathbb{S} = \{-1, 0, 1\}$ the set of signs, \mathbb{N} the set of natural numbers including 0, \mathbb{Z} the set of integers, \mathbb{R} the set of real numbers, and \mathbb{R}_+ the set of positive real numbers including 0. Note that $\mathbb{B} \subseteq \mathbb{N} \subseteq \mathbb{R}_+ \subseteq \mathbb{R}$ and that $\mathbb{S} \subseteq \mathbb{Z} \subseteq \mathbb{R}$. For signs, we use the symbols $\nearrow = 1$ for increase, $\rightarrow = 0$ for no-change and $\searrow = -1$ for decrease.

3 Arithmetic Expressions

We recall the syntax and semantics of arithmetic expressions while pointing out three different interpretations and usages. Let \mathcal{V} be a set of variables and $\Sigma_{arith}^{(2)}$ be the set of binary operators $\{+, *, -, /, exp\}$. The set of arithmetic expressions $e \in \mathcal{E}_{arith}(\mathcal{V})$ is the least set of terms that can be build from variables $x \in \mathcal{V}$, real numbers $\rho \in \mathbb{R}$, and binary operators $\odot \in \Sigma_{arith}^{(2)}$:

$$e_1, e_2 \in \mathcal{E}_{arith}(\mathcal{V}) ::= \rho \mid x \mid e_1 \odot e_2$$

Definition 1. A relational structure with binary operators in $\Sigma_{arith}^{(2)}$ and constants in \mathbb{R} is a tuple $S = (\text{dom}(S), (\odot^S)_{\odot \in \Sigma_{arith}^{(2)}}, (\rho^S)_{\rho \in \mathbb{R}})$ where $D = \text{dom}(S)$ is a set called the domain, $\odot^S : D^2 \times D$ is the interpretation of \odot , and $\rho^S \in D$ the interpretation of ρ .

$$\begin{aligned} \llbracket \rho \rrbracket^{\alpha, S} &= \{\rho^S\}, & \llbracket x \rrbracket^{\alpha, S} &= \{\alpha(x)\}, \\ \llbracket e_1 \odot e_2 \rrbracket^{\alpha, S} &= \{s \mid s_1 \in \llbracket e_1 \rrbracket^{\alpha, S}, s_2 \in \llbracket e_2 \rrbracket^{\alpha, S}, (s_1, s_2, s) \in \odot^S\} \end{aligned}$$

Fig. 1: Interpretation of arithmetic expressions over a relational structure S and a variable assignment $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$, where $e_1, e_2 \in \mathcal{E}_{arith}(\mathcal{V})$, $\odot \in \Sigma_{arith}^{(2)}$, $\rho \in \mathbb{R}$, and $x \in \mathcal{V}$.

For any relational structure S , variable assignment $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$ and arithmetic expression $e \in \mathcal{E}_{arith}(\mathcal{V})$, Fig. 1 defines the interpretation $\llbracket e \rrbracket^{S, \alpha} \subseteq \text{dom}(S)$. We consider interpretations over three different relational structures \mathbb{R} , $\mathbb{R}_+ \rightarrow \mathbb{R}$, and \mathbb{S} , that we freely confuse with their domain.

First, arithmetic expressions are used as kinetic expressions. They are then interpreted in the relational structure of the reals \mathbb{R} . The binary operators are interpreted as the binary partial functions $\odot^{\mathbb{R}}$ for the addition, multiplication, subtraction, division, and exponentiation of real numbers respectively. Note that $/^{\mathbb{R}}$ is not a total function, since division by zero is not defined. Therefore, $\llbracket x/0 \rrbracket^{\mathbb{R}, \alpha} = \emptyset$ for any $\alpha : \mathcal{V} \rightarrow \mathbb{R}$.

Second, we use arithmetic expressions in ODEs. Then they are interpreted in the structure of real-valued functions $\mathbb{R}_+ \rightarrow \mathbb{R}$. The binary operators now denote the partial functions $\odot^{\mathbb{R}_+ \rightarrow \mathbb{R}}$ for the addition, multiplication, subtraction, division, and exponentiation of real-valued functions respectively. Note that any constant $\rho \in \mathbb{R}$ is interpreted as the constant function with $\rho^{\mathbb{R}_+ \rightarrow \mathbb{R}}(\rho') = \rho$ for all $\rho' \in \mathbb{R}_+$.

Third, we will interpret arithmetic expressions over the structure of signs $\mathbb{S} = \{\nearrow, \rightarrow, \searrow\}$. The binary operators need to be interpreted as ternary relations $\odot^{\mathbb{S}} \subseteq \mathbb{S}^2 \times \mathbb{S}$. Let $h_{\mathbb{S}} : \mathbb{R} \rightarrow \mathbb{S}$ be the unique homomorphism between the structures of reals and of signs. It satisfies for all $\rho \in \mathbb{R}$ that $h_{\mathbb{S}}(\rho) = 1$ if $\rho > 0$, $h_{\mathbb{S}}(\rho) = 0$ if $\rho = 0$ and $h_{\mathbb{S}}(\rho) = -1$ if $\rho < 0$. Then we define for any constant $\rho \in \mathbb{R}$ that $\rho^{\mathbb{S}} = h_{\mathbb{S}}(\rho)$ and for any operator $\odot \in \Sigma_{arith}^{(2)}$ that $\odot^{\mathbb{S}} = \{(h_{\mathbb{S}}(\rho), h_{\mathbb{S}}(\rho'), h_{\mathbb{S}}(\rho'')) \mid (\rho, \rho', \rho'') \in \odot^{\mathbb{R}}\}$. We note that the addition $+^{\mathbb{S}}$ is not even a partial function, since $(\nearrow, \searrow, \sigma) \in +^{\mathbb{S}}$ for all three signs $\sigma \in \mathbb{S}$. For this reason, $\llbracket \nearrow + \searrow \rrbracket^{\mathbb{S}, \gamma} = \mathbb{S}$ for any $\gamma : \mathcal{V} \rightarrow \mathbb{S}$. The intuition is that an arithmetic expression is evaluated non-deterministically to the set of all possible signs.

4 Chemical Reaction Networks

Let \mathcal{S} be a finite set. A chemical solution with species in \mathcal{S} is a multiset $M : \mathcal{S} \rightarrow \mathbb{N}$. The multiset $[A/3, B/2]$ for instance is often written as $3A + 2B$.

Definition 2. *A reaction with species in \mathcal{S} is an element of $\mathbb{N}^{\mathcal{S}} \times \mathcal{E}_{arith}(\mathcal{S}) \times \mathbb{N}^{\mathcal{S}}$. A reaction network with species in \mathcal{S} is a subset of reactions with species in \mathcal{S} .*

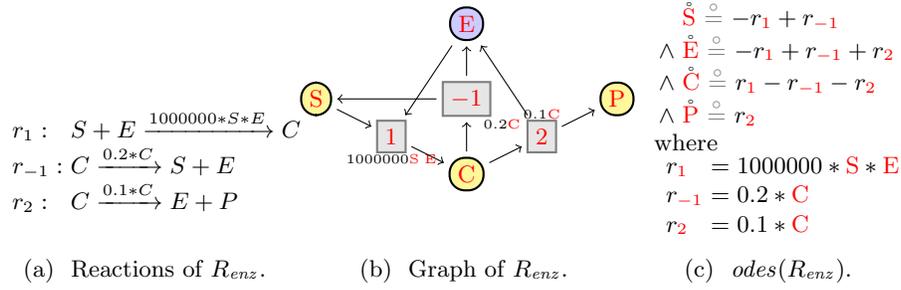


Fig. 2: The enzymatic reaction network R_{enz} .

For instance, if $e = 5.1 * \exp(A, 2) * B$ then $r = (3A + B, e, A + 2C)$ is a chemical reaction, that we denote as usual as $r : 3A + B \xrightarrow{e} A + 2C$. A state α of a reaction network R assigns each species of R a concentration, which is a positive real number, so $\alpha : \mathcal{S} \rightarrow \mathbb{R}_+$. Let $\llbracket e \rrbracket^{\mathbb{R}, \alpha} = 5.1 *^{\mathbb{R}} \exp^{\mathbb{R}}(\llbracket A \rrbracket^{\mathbb{R}, \alpha}, 2) *^{\mathbb{R}} \llbracket B \rrbracket^{\mathbb{R}, \alpha}$. The above reaction states that the concentration of A changes at any time point with state α with speed $-2 *^{\mathbb{R}} \llbracket e \rrbracket^{\mathbb{R}, \alpha}$, the concentration of B with speed $-1 *^{\mathbb{R}} \llbracket e \rrbracket^{\mathbb{R}, \alpha}$, and the concentration of C with speed $2 *^{\mathbb{R}} \llbracket e \rrbracket^{\mathbb{R}, \alpha}$. Negative speeds mean that the species is consumed, while positive speeds mean that the species is produced.

Let $r = (M, e, M')$ be a chemical reaction with species in \mathcal{S} . We denote the kinetic expression of r by $kin_r = e$. For any species $A \in \mathcal{S}$, the stoichiometry of A in r is defined by $stoic_r(A) = M'(A) - M(A)$. The ODE system of a reaction network R is the following equation system:

$$odes(R) =_{\text{def}} \bigwedge_{A \in \mathcal{S}} \dot{A} \stackrel{\circ}{=} \sum_{r \in R} stoic_r(A) * kin_r \wedge A \geq 0$$

A formal definition of the syntax and semantics of ODEs will be given in Section 6 based on notions from the first-order logic in Section 5. For now, we just state that all species occurring in an arithmetic expression denote some real valued function of type $\mathbb{R}_+ \rightarrow \mathbb{R}$, that must be positive in addition. An expression \dot{A} denotes the derivative of the denotation of A if its derivative exists, and is undefined otherwise. Note that derivatives may become negative. The arithmetic operators are interpreted as arithmetic operations in the structure of real-valued functions $\mathbb{R}_+ \rightarrow \mathbb{R}$.

As an example of a CRN, we show the network of enzymatic reactions in Fig. 2. It has species $\mathcal{S} = \{S, E, C, P\}$ and the three reactions in Fig. 2a, all with mass action kinetics. Reaction r_1 transforms a pair of a substrate S and an enzyme E to a complex C , reaction r_{-1} does the inverse, and reaction r_2 transforms the complex C into the free enzyme E and the product P . The graph of the CRN R_{enz} is given Fig. 2b and its ODEs in Fig. 2c.

$$\begin{aligned}
\llbracket e \stackrel{\circ}{=} e' \rrbracket^{\alpha, S} &= \begin{cases} 1 & \text{if } \llbracket e \rrbracket^{\alpha, S} \cap \llbracket e' \rrbracket^{\alpha, S} \neq \emptyset \\ 0 & \text{else} \end{cases} & \llbracket \phi \wedge \phi' \rrbracket^{\alpha, S} &= \llbracket \phi \rrbracket^{\alpha, S} \wedge^{\mathbb{B}} \llbracket \phi' \rrbracket^{\alpha, S} \\
\llbracket \neg \phi \rrbracket^{\alpha, S} &= \neg^{\mathbb{B}}(\llbracket \phi \rrbracket^{\alpha, S}) & \llbracket \exists x. \phi \rrbracket^{\alpha, S} &= \begin{cases} 1 & \text{if exists } s \in \text{dom}(S). \\ & \llbracket \phi \rrbracket^{\alpha[x/s], S} = 1 \\ 0 & \text{else} \end{cases}
\end{aligned}$$

Fig. 3: Interpretation of formulas $\phi \in \mathcal{F}_{\Sigma}$ over a relational structure S with signature Σ with respect to a variable assignment $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$.

5 First-Order Logic

We recall the syntax and semantics of first-order logic formulas. Rather than opting for maximal generality, we restrict ourselves to signatures able to capture ODEs and FO-BNNs.

We consider signatures $\Sigma = \Sigma^{(2)} \cup \Sigma^{(1)} \cup \mathbb{R}$ with a set of binary operators in $\Sigma^{(2)}$, unary operators in $\Sigma^{(1)}$, and constants in \mathbb{R} . Arithmetic expressions have the signature $\Sigma_{arith} = \Sigma_{arith}^{(2)} \cup \Sigma_{arith}^{(1)} \cup \mathbb{R}$ with $\Sigma_{arith}^{(1)} = \emptyset$. An expression in $\mathcal{E}_{\Sigma}(\mathcal{V})$ then has the following form where $x \in \mathcal{V}$, $\rho \in \mathbb{R}$, $\odot \in \Sigma^{(2)}$ and $o \in \Sigma^{(1)}$.

$$e, e_1, e_2 \in \mathcal{E}_{\Sigma}(\mathcal{V}) \quad ::= x \mid \rho \mid e_1 \odot e_2 \mid o(e)$$

Clearly, these expressions generalize on arithmetic expressions since $\mathcal{E}_{arith}(\mathcal{V}) = \mathcal{E}_{\Sigma_{arith}}(\mathcal{V})$. The notion of relational structures can be lifted from the signature of arithmetic expressions to expressions with general signatures in the obvious manner, and also the semantics from arithmetic expressions to expressions in $\mathcal{E}_{\Sigma}(\mathcal{V})$.

We are particularly interested in the signature $\dot{\Sigma}_{arith}$ with $\dot{\Sigma}_{arith}^{(2)} = \Sigma_{arith}^{(2)}$ and $\dot{\Sigma}_{arith}^{(1)} = \{ \dot{\ } \}$. The structure of real-valued functions can be extended to give an interpretation of the unary operator $\dot{\ }$, so that $\mathbb{R}_+ \rightarrow \mathbb{R}(f)$ is the derivative of f for any derivable real-valued function $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, and undefined otherwise.

The set of first-order formulas $\mathcal{F}_{\Sigma}(\mathcal{V})$ is constructed from equations between expressions in $e, e' \in \mathcal{E}_{\Sigma}(\mathcal{V})$ and the usual first-order connectives:

$$\phi \in \mathcal{F}_{\Sigma}(\mathcal{V}) \quad ::= e \stackrel{\circ}{=} e' \mid \exists x. \phi \mid \phi \wedge \phi \mid \neg \phi \quad \text{where } x \in \mathcal{V}$$

We sometimes use shortcuts $e \geq 0$ for the formula $\exists x. e \stackrel{\circ}{=} x * x$ and $e \leq e'$ for $e' - e \geq 0$. The set of free variables $fv(\phi)$ contains all those variables of ϕ that occur outside the scope of any occurrence of the existential quantifier.

The semantics of a first-order formula $\phi \in \mathcal{F}_{\Sigma}(\mathcal{V})$ is the truth value $\llbracket \phi \rrbracket^{\alpha, S} \in \mathbb{B}$ defined in Fig. 3. It depends on some relational structure S with signature Σ and variable assignment $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$. An equation $e \stackrel{\circ}{=} e'$ is true if the intersection of the possible values for e and the possible values for e' is non-empty, that is, if $\llbracket e \rrbracket^{\alpha, S} \cap \llbracket e' \rrbracket^{\alpha, S} \neq \emptyset$. The set of solutions of a formula $\phi \in \mathcal{F}_{\Sigma}(\mathcal{V})$ over a relational structure S with the same signature Σ is $\text{sol}^S(\phi) = \{ \alpha|_{fv(\phi)} \mid \alpha : \mathcal{V} \rightarrow \text{dom}(S), \llbracket \phi \rrbracket^{\alpha, S} = 1 \}$.

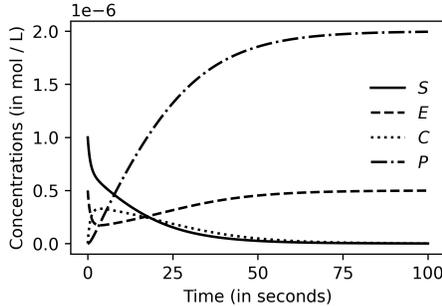


Fig. 4: The deterministic numerical simulation of $odes(R_{enz})$ with initial concentrations $S(0) = 1.0 * 10^{-5}$, $E(0) = 0.5 * 10^{-5}$ and $P(0) = C(0) = 0$ mol/L.

6 Sign Abstraction of ODE Trajectories

We define systems of ODEs as formulas of first-order logic in order to formalize their syntax and semantics in a framework suitable for abstract interpretation. We then introduce a temporal and a causal transition relation on abstract states, by applying the sign abstraction to the trajectories of the ODEs.

Definition 3. *An ODE system is a first-order logic formula $\phi \in \mathcal{F}_{\Sigma_{arith}}(\mathcal{V})$ such that all subexpressions of ϕ that are rooted by the dot-operator have the form \dot{x} for some $x \in \mathcal{V}$ and do not occur below any existential quantifier.*

Note in particular that higher-order derivations \ddot{x} are not permitted. The semantics of an ODE system ϕ is the set of its solutions over the relational structure of real-valued functions, i.e., $sol^{\mathbb{R}^+ \rightarrow \mathbb{R}}(\phi)$. For each such solution β and variable $x \in fv(\phi)$, we call $\beta(x) : \mathbb{R}_+ \rightarrow \mathbb{R}$ a trajectory of x .

For any choice of initial concentrations $\alpha : fv(\phi) \rightarrow \mathbb{R}^+$, there exists at most one solution $\beta \in sol^{\mathbb{R}^+ \rightarrow \mathbb{R}}(\phi)$, such that $\beta(x)(0) = \alpha(x)$ for all $x \in fv(\phi)$. This solution can be computed numerically by the usual integration methods for ODEs starting with the initial concentrations. If some operations of the ODEs are undefined during the integration, no solution exists. For illustration, we show in Fig. 4 the solution of the ODEs of the reaction network R_{enz} with initial concentrations $S(0) = 1.0 * 10^{-5}$, $E(0) = 0.5 * 10^{-5}$ and $P(0) = C(0) = 0$ mol/L.

A (concrete) state with variables $V \subseteq \mathcal{V}$ is a function $\alpha : V \rightarrow \mathbb{R}$, and an abstract state a function $\gamma : V \rightarrow \mathbb{S}$. We next show how to define a successor relation on the abstract states of ODEs.

Definition 4 Temporal next transitions. *Let $\gamma_1, \gamma_2 : \mathcal{S} \rightarrow \mathbb{S}$ be two abstract states and ϕ an ODE system. We call γ_2 a (temporally) next state of γ_1 with respect to ϕ and write $(\gamma_1, \gamma_2) \in next_\phi$ if there exists a real-valued function $\beta \in sol^{\mathbb{R}^+ \rightarrow \mathbb{R}}(\phi)$ and two time points $0 \leq t_1 < t_2$ such that, for all species $A \in \mathcal{S}$ and time points $t'_2 \in]t_1, t_2]$: $\gamma_1(A) = h_{\mathbb{S}}(\beta(A)(t_1))$ and $\gamma_2(A) = h_{\mathbb{S}}(\beta(A)(t'_2))$.*

For instance, for $odes(R_{enz})$, the next state of $[S/1, E/1, C/0, P/0]$ is $[S/1, E/1, C/1, P/1]$, which has itself as next state. For this example, the next states are always unique, but in general this may not be the case.

Interestingly, $[S/1, E/1, C/0, P/0]$ does not have $[S/1, E/1, C/1, P/0]$ as next state. The reason is that instantaneously when C is produced, reaction r_2 starts producing P , so that both C and P will appear at the same time point. Nevertheless, the creation of C causes the creation of P , but this is not observable in the temporal order and thus not in the relation $next_{ode(R_{enz})}$.

So far, the states do not contain any information about the values of the derivatives (since \dot{x} is not a variable but an expression). In order to change this, let $\circ : \mathcal{V} \rightarrow \mathcal{V}$ be a bijection such that \dot{V} is disjoint from V for any $V \subseteq \mathcal{V}$. For any ODE system ϕ we define a formula $\dot{\phi} \in \mathcal{F}_{\Sigma_{arith}}$ without dot operator, by replacing any subexpression of the form \dot{x} in ϕ by \dot{x} :

$$\dot{\phi} =_{\text{def}} \phi[\dot{x}/\dot{x} \mid x \in fv(\phi)]$$

Note that no dot operators remain in $\dot{\phi}$, since all subexpressions of ϕ that are rooted by the dot operator must be of the form \dot{x} by the definition of ODEs. For any assignment $\beta : V \rightarrow (\mathbb{R}_+ \rightarrow \mathbb{R})$ and time point $t \geq 0$ we define an assignment $\beta_t : \{x, \dot{x} \mid x \in V\} \rightarrow \mathbb{R}$ such that: $\beta_t(x) = \beta(x)(t)$ and $\beta_t(\dot{x}) = \mathbb{R}_+ \rightarrow \mathbb{R}(\beta(x))(t)$.

Lemma 5. *If $\beta \in sol^{\mathbb{R}_+ \rightarrow \mathbb{R}}(\phi)$ then $\beta_t \in sol^{\mathbb{R}}(\dot{\phi})$.*

When interested in derivatives, we consider the successor relation of the formula $next_{\dot{\phi}}$. For instance, with respect to $next_{odes(\dot{R}_{enz})}$, the abstract state $\gamma_1 = [S/1, E/1, C/0, P/0, \dot{S}/\searrow, \dot{E}/\searrow, \dot{C}/\nearrow, \dot{P}/\rightarrow]$ has the successor $\gamma_2 = [S/1, E/1, C/1, P/1, \dot{S}/\searrow, \dot{E}/\searrow, \dot{C}/\nearrow, \dot{P}/\nearrow]$. Furthermore, causality can be observed in the signs of the derivatives: we have $\gamma_1(\dot{C}) = \nearrow$ since $\gamma_1(E) = \gamma_1(S) = 1$. In contrast, we have $\gamma_1(\dot{P}) = \rightarrow$ since $\gamma_1(C) = 0$. As a consequence, for any solution $\beta \in sol^{\mathbb{R}_+ \rightarrow \mathbb{R}}(odes(R_{enz}))$, the value of $\lim_{t \rightarrow 0} \beta(C)(t)/t \neq 0$ so the change of $C(t)$ at $t = 0$ can be observed in the limit, while $\lim_{t \rightarrow 0} \beta(P)(t)/t = 0$, so the change of $P(t)$ at $t = 0$ cannot be observed in the limit. Nevertheless $\gamma_2(C) = \gamma_2(P) = 1$, since the successor time point of 0 is not in the limit.

Definition 6 Causal next transitions. *Let ϕ be an ODE system, $V = fv(\phi)$, and $\gamma_1, \gamma_2 : (V \cup \dot{V}) \rightarrow \mathbb{S}$ abstract states. We call γ_2 a causally-next dotted state of γ_1 and write $(\gamma_1, \gamma_2) \in cnext_{\dot{\phi}}$ if $\gamma_2 \in sol^{\mathbb{S}}(\dot{\phi})$ and there exists an abstract state γ'_2 such that $(\gamma_1, \gamma'_2) \in next_{\dot{\phi}}$ and for all $x \in V$: $\gamma_2(x) = \gamma_1(x)$ if $\gamma'_2(\dot{x}) = 0$ and $\gamma_2(x) = \gamma'_2(x)$ otherwise. We say that $\gamma_{2|V}$ is a causally-next state of $\gamma_{1|V}$ and write $(\gamma_{1|V}, \gamma_{2|V}) \in cnext_{\phi}$ if $(\gamma_1, \gamma_2) \in cnext_{\dot{\phi}}$.*

For $odes(R_{enz})$, the causally next state of $[S/1, E/1, C/0, P/0]$ is $[S/1, E/1, C/1, P/0]$, of which the causally next state is $[S/1, E/1, C/1, P/1]$. As here we often have $next_{odes(R)} \subseteq (cnext_{odes(R)})^*$. We believe that this holds more generally for any reaction network R for which the numerical simulation by Euler's algorithm is sound when using exact arithmetics. Euler's algorithm performs simulation steps, which may introduce approximation errors. These errors may lead to arbitrarily false traces in some case, but may also be ignored in many others. For R_{enz} , this is the error of setting the value of P at the next step to 0 if the value of P was 0 at the previous step and $\dot{P} = 0$ (since C was absent). In reality, P should be set to a small non-zero value at the next step.

7 Abstract Interpretation of ODEs

We recall John’s soundness theorem [17] for the abstract interpretation of first-order logic formulas, and apply it for interpreting ODEs abstractly over signs.

Theorem 7 John’s Soundness Theorem [1,24,17]. *For any homomorphism $h : S \rightarrow \Delta$ between relational structures with signature Σ and any negation-free formula $\phi \in \mathcal{F}_\Sigma(\mathcal{V})$: $h \circ \text{sol}^S(\phi) \subseteq \text{sol}^\Delta(\phi)$.*

Proof We only give a sketch of the proof. Let $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$. For any expression $e \in \mathcal{E}_\Sigma(\mathcal{V})$ we can show that $h(\llbracket e \rrbracket^{\alpha, S}) = \llbracket e \rrbracket^{h \circ \alpha, \Delta}$ by induction on the structure of e . It then follows for any positive formula $\phi \in \mathcal{F}_\Sigma(\mathcal{V})$ that $\llbracket \phi \rrbracket^{\alpha, S} \leq \llbracket \phi \rrbracket^{h \circ \alpha, \Delta}$. This is equivalent to that: $\{h \circ \alpha \mid \alpha \in \text{sol}^S(\phi)\} \subseteq \text{sol}^\Delta(\phi)$ and thus $h \circ \text{sol}^S(\phi) \subseteq \text{sol}^\Delta(\phi)$. \square

Recall that $h_\mathbb{S} : \mathbb{R} \rightarrow \mathbb{S}$ is a homomorphism between relational structures with signature Σ_{arith} and that the formula $\dot{\phi}$ has the same signature Σ_{arith} for any ODE ϕ . John’s theorem thus shows:

Corollary 8. *For any ODE ϕ : $h_\mathbb{S} \circ \text{sol}^\mathbb{R}(\dot{\phi}) \subseteq \text{sol}^\mathbb{S}(\dot{\phi})$.*

This corollary states that the set of abstract dotted states of an ODE ϕ can be overapproximated by interpreting $\dot{\phi}$ abstractly in the structure of signs. It can be used to reason about the temporal and causal next transition relations of ODEs as follows:

Lemma 9. *For any ODE ϕ , if $(\gamma_1, \gamma_2) \in \text{next}_{\dot{\phi}} \cup \text{cnext}_\phi$, then $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^\mathbb{S}(\dot{\phi})$.*

Proof

1. Definition 4 of the temporal next relation and Lemma 5 show that for any pair $(\gamma_1, \gamma_2) \in \text{next}_{\dot{\phi}}$ that it satisfies $\gamma_1 = h_\mathbb{S} \circ \alpha_1$ and $\gamma_2 = h_\mathbb{S} \circ \alpha_2$ for some $\alpha_1, \alpha_2 \in \text{sol}^\mathbb{R}(\dot{\phi})$. Corollary 8 of John’s soundness theorem for abstract interpretation of logic formulas applied to ODEs shows that $h_\mathbb{S} \circ \text{sol}^\mathbb{R}(\dot{\phi}) \subseteq \text{sol}^\mathbb{S}(\dot{\phi})$ and thus $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^\mathbb{S}(\dot{\phi})$.
2. If $(\gamma_1, \gamma_2) \in \text{cnext}_\phi$ then $\gamma_2 \in \text{sol}^\mathbb{S}(\dot{\phi})$ by Definition 6. Furthermore, there exists γ'_2 such that $(\gamma_1, \gamma'_2) \in \text{next}_{\dot{\phi}}$. The first property shows that $\gamma_1 \in \text{sol}^\mathbb{S}(\dot{\phi})$ too. \square

8 Boolean Networks with Non-deterministic Updates

Any (abstract) state in \mathbb{B}^V is a function $\beta : V \rightarrow \mathbb{B}$ that we call a bit vector. For instance, the state $[S/1, E/1, C/0, P/1]$ can be identified with the bit vector 1101 when ordering the species as in above. In the pictures of state transition graphs, the states are drawn as bit vectors in oval nodes and the state transitions as arrows linking these nodes. The legends in blue boxes specify the species order.

Following [25], a Boolean network B with non-deterministic updates (BNN) and species in \mathcal{S} is generally some kind of definition of an abstract state transition

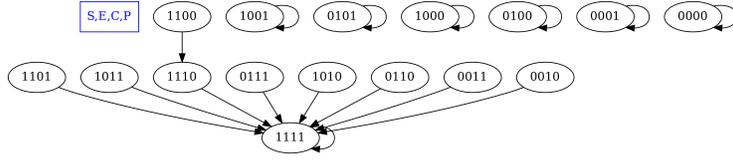


Fig. 5: The state transition graph of the FO-BNN $bnn(R_{enz})$ in Fig. 6.

graph $B \subseteq \mathbb{B}^S \times \mathbb{B}^S$, as for instance in Fig. 5. Definitions of state transition graphs can be expressed in various manners. Here we propose a novel alternative that is based on formulas of first-order logic interpreted over the structure of signs \mathbb{S} . We assume that $\mathcal{S} \subseteq \mathcal{V}$ and fix two bijections $\vec{\cdot} : \mathcal{V} \rightarrow \mathcal{V}$ and $\circ : \mathcal{V} \rightarrow \mathcal{V}$ such that \mathcal{S} , $\vec{\mathcal{S}}$, $\mathring{\mathcal{S}}$, and $\dot{\mathcal{S}}$ are disjoint subsets of \mathcal{V} . Furthermore, we assume that $\vec{\vec{x}} = \vec{x}$ for any $x \in \mathcal{V}$.

From the perspective of the sign abstraction of a reaction network, variable $A \in \mathcal{S}$ states whether species A is present at the current time point and \dot{A} is the sign of \dot{A} . Variable \vec{A} stands for the presence of A at the next time point, and similarly, $\vec{\dot{A}}$ for the sign of the derivative at the next time point $\vec{\dot{A}}$.

Definition 10. A first-order Boolean network with nondeterministic updates (FO-BNN) and variables in $V \subseteq \mathcal{V}$ is a first-order formula $\phi \in \mathcal{F}_{\Sigma_{arith}}$ with free variables $fv(\phi) = V \cup \vec{V}$.

Notice that the dot operator cannot occur in FO-BNNs ϕ . Any variable assignment $\gamma : V \cup \vec{V} \rightarrow \mathbb{S}$ yields an abstract state transition:

$$trans(\gamma) = (\gamma|_V, \gamma|_{\vec{V}} \circ \vec{\cdot}) \in \mathbb{S}^V \times \mathbb{S}^V$$

For illustration, an FO-BNN for R_{enz} with the (free) variables in $\mathcal{S} = \{S, E, C, P\}$ is shown in Fig. 6. It can be inferred from the ODEs of the reaction network as follows: First the formula $ode(R_{enz})$ is added. Second, a copy of $ode(R_{enz})$ is added, in which all variables x are replaced by \vec{x} . All variables in $\vec{\mathcal{S}} \cup \mathring{\mathcal{S}}$ are existentially quantified. Furthermore, for any species $A \in \mathcal{S}$, we relate the variable \vec{A} to the variables A and \dot{A} by the equation $\vec{A} \stackrel{\circ}{=} A + \dot{A}$. This states that \vec{A} can be true only if either A or \dot{A} are true, i.e. if the concentration of A is either present or increasing at the previous time point. Finally, we impose $A \leq \vec{A}$ for stating that species A can never become absent when it was present before. Because the kinetics in R_{enz} are all mass-action laws, this invariant holds here for all the species. It may be false for other reaction networks though.

Furthermore, the solution sets of FO formulas over finite relational structures such as \mathbb{S} can be computed by finite set constraint programming. We have implemented a constraint solver for the relational structure \mathbb{S} in Minizinc [23] which allows us to compute the transition graph of FO-BNNs, i.e., the relation

$$\begin{aligned}
& \exists \vec{S} \exists \vec{E} \exists \vec{C} \exists \vec{P} \exists \vec{S} \exists \vec{E} \exists \vec{C} \exists \vec{P}. \\
& \vec{S} \stackrel{\circ}{=} -r_1 + r_{-1} \quad \wedge \vec{S} \stackrel{\circ}{=} -\vec{r}_1 + \vec{r}_{-1} \quad \wedge \vec{S} \stackrel{\circ}{=} S + \dot{S} \quad \wedge S \leq \vec{S} \\
& \wedge \vec{E} \stackrel{\circ}{=} -r_1 + r_{-1} + r_2 \quad \wedge \vec{E} \stackrel{\circ}{=} -\vec{r}_1 + \vec{r}_{-1} + \vec{r}_2 \quad \wedge \vec{E} \stackrel{\circ}{=} E + \dot{E} \quad \wedge E \leq \vec{E} \\
& \wedge \vec{C} \stackrel{\circ}{=} r_1 - r_{-1} - r_2 \quad \wedge \vec{C} \stackrel{\circ}{=} \vec{r}_1 - \vec{r}_{-1} - \vec{r}_2 \quad \wedge \vec{C} \stackrel{\circ}{=} C + \dot{C} \quad \wedge C \leq \vec{C} \\
& \wedge \vec{P} \stackrel{\circ}{=} r_2 \quad \wedge \vec{P} \stackrel{\circ}{=} \vec{r}_2 \quad \wedge \vec{P} \stackrel{\circ}{=} P + \dot{P} \quad \wedge P \leq \vec{P} \\
& \text{where} \\
& r_1 = 1000000 * S * E \quad r_{-1} = 0.2 * C \quad r_2 = 0.1 * C \\
& \vec{r}_1 = 1000000 * \vec{S} * \vec{E} \quad \vec{r}_{-1} = 0.2 * \vec{C} \quad \vec{r}_2 = 0.1 * \vec{C}
\end{aligned}$$

Fig. 6: An FO-BNN for the reaction network R_{enz} .

of bit vectors that it defines. The transition graph in Fig. 5 is defined by the FO-BNN for the reaction network R_{enz} in Fig. 6.

The set of species of reaction network R_{enz} has cardinality 4. Therefore, the state transition graph of the FO-BNN of R_{enz} has $2^4 = 16$ states. So clearly, the number of states of a FO-BNN may be exponential in the number of its species. Therefore, it is generally advantageous if one does not have to compute the whole state transition graph, but only the needed part of it. Suppose that we know the sign abstraction of the initial state, we can then generate the subgraph of the state transition graph that is accessible from the abstraction of the initial state, without computing any further states or transitions. In this way, much smaller subgraphs can be observed. For R_{enz} , for instance, for any Boolean state the accessible subgraph contains at most 3 Boolean states.

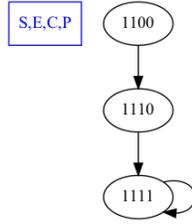


Fig. 7: Abstract simulation of the FO-BNN in Fig. 6 for the reaction network R_{enz} .

```

fun abs_sim( $\phi, V, \gamma_0$ )
   $T_{\text{reach}} := \emptyset$    $S_{\text{reach}} := \{\gamma_0\}$    $S_{\text{new}} := S_{\text{reach}}$ 
  while  $S_{\text{new}} \neq \emptyset$  :
     $T_{\text{new}} := \text{trans} \circ \text{sol}^{\vec{S}}(\phi \wedge$ 
       $\bigvee_{\gamma \in S_{\text{new}}} \bigwedge_{x \in V \cup \vec{V}} x \stackrel{\circ}{=} \gamma(x))$ 
     $S_{\text{new}} := \{\gamma_2 \mid (\gamma_1, \gamma_2) \in T_{\text{new}}\} \setminus S_{\text{reach}}$ 
     $T_{\text{reach}} := T_{\text{reach}} \cup T_{\text{new}}$ 
     $S_{\text{reach}} := S_{\text{reach}} \cup S_{\text{new}}$ 
  return  $T_{\text{reach}}$ 

```

Fig. 8: The abstract simulation of an FO-BNN ϕ with variables in V from an abstract initial state γ_0 computes the set of abstract state transitions T_{reach} .

The abstract simulation of the FO-BNN of R_{enz} , starting with the abstract state $[S/1, E/1, C/0, P/0]$ is given in Fig. 7. This example illustrates, that abstract simulation is related to the causality rather than the temporality of species

production. For instance, the temporal transition $(1100, 1111) \in \text{next}_{ode(R_{enz})}$ is represented by two causal edges $1100 \rightarrow 1110 \rightarrow 1111$ in the abstract simulation. These show the causality of the production: C is produced if S and E are present, and P is produced if C is present. But when C is produced then instantaneously P is produced too, so even though C causally precedes P (as shown by the abstract simulation), they are both produced at the same time (in any solution of the ODEs). We notice that causality also plays for concrete numerical simulation with Euler's method: P will be produced shortly after C , depending on the step size that is admitted. The reachable subgraph can be computed by repeated constraint solving. In each step, the values of the variables in \mathcal{S} are constrained to the Boolean states from which the subgraph is to be explored.

In general, for any FO-BNN ϕ with species in \mathcal{S} and abstract initial state $\gamma_0 : V \cup \vec{V} \rightarrow \mathbb{S}$, the abstract simulation represented by the set S_{reach} of all reachable states and the corresponding transition relation T_{reach} can be obtained by iteratively computing the sets of new transitions T_{new} and new reachable states S_{new} starting from the initial state γ_0 , as in Fig. 8. The algorithm ends when no new reachable states are obtained from the available transitions.

9 Abstract Simulation of Reaction Networks

To simulate reaction networks abstractly, we propose to translate them to FO-BNNs based on the abstract interpretation of logic formulas over the structure of signs. For any variable assignment $\alpha : V \rightarrow \mathbb{R}$, we define $\vec{\alpha} : \vec{V} \rightarrow \mathbb{R}$ such that $\alpha(x) = \vec{\alpha}(\vec{x})$ for all $x \in V$. For any reaction network R , let \vec{R} be the reaction network with species in $\vec{\mathcal{S}}$ obtained from R by replacing any species $A \in \mathcal{S}$ by \vec{A} . For any $A \in \mathcal{S}$, let vars_A be the sequence of the four variables $A, \overset{\circ}{A}, \vec{A}, \vec{\vec{A}}$.

Our objective is to approximate the relation $\text{cnext}_{ode(R)}$ on abstract states by a Boolean network. For any $A \in \mathcal{S}$, we consider formulas $\text{next_spec}(\text{vars}_A)$ with the following property. For all $\gamma_1, \gamma_2 : \mathcal{S} \cup \vec{\mathcal{S}} \rightarrow \mathbb{S}$ and all reaction networks R with species in \mathcal{S} :

$$(\gamma_1, \gamma_2) \in \text{cnext}_{odes(R)} \Rightarrow (\gamma_1 \cup \vec{\gamma}_2)|_{\{\text{vars}_A\}} \in \text{sol}^{\mathbb{S}}(\text{next_spec}(\text{vars}_A))$$

There are several possibilities to define $\text{next_spec}(\text{vars}_A)$, of which we propose:

- $\text{next_spec}_1(\text{vars}_A) \stackrel{\vec{\rightarrow}}{=}_{\text{def}} (\vec{A} \overset{\circ}{=} 1 \rightarrow (A \overset{\circ}{=} 0 \wedge \overset{\circ}{A} \overset{\circ}{=} 1) \vee A \overset{\circ}{=} 1) \wedge (\vec{A} \overset{\circ}{=} 0 \rightarrow (A \overset{\circ}{=} 0 \wedge \overset{\circ}{A} \overset{\circ}{=} 0) \vee \vec{A} \overset{\circ}{=} -1)$
- $\text{next_spec}_2(\text{vars}_A) \stackrel{\vec{\rightarrow}}{=}_{\text{def}} \text{next_spec}_1(\text{vars}_A) \wedge (\vec{A} \overset{\circ}{=} 0 \rightarrow ((A \overset{\circ}{=} 0 \wedge \overset{\circ}{A} \overset{\circ}{=} 0) \vee (A \overset{\circ}{=} 1 \wedge \overset{\circ}{A} \overset{\circ}{=} -1 \wedge \vec{A} \overset{\circ}{=} 1)))$
- $\text{next_spec}_3(\text{vars}_A) \stackrel{\vec{\rightarrow}}{=}_{\text{def}} \text{next_spec}_1(\text{vars}_A) \wedge (\vec{A} \overset{\circ}{=} 0 \rightarrow A \overset{\circ}{=} 0)$

It is not difficult to see that $\text{next_spec}_1(\text{vars}_A)$ satisfies the above requirement since using causally-next relation (but this would not hold for the temporal-next relation). If all kinetic expressions are infinitely derivable, then, when a concentration becomes 0, the derivative requires an increase immediately after,

in order to not become negative. If all reactions follow the mass action law then non-zero concentrations can never become zero later on, so $next_spec_3(vars_A)$ should satisfy the requirement too.

Let $next_spec(vars_A)$ be one of the three formulas $next_spec_i(vars_A)$ above or any other formula satisfying the above property. Which of these choices is applicable or best depends on properties of the reaction network.

Definition 11. For any reaction network R with species $\mathcal{S} = \{A_1, \dots, A_n\}$, we define the FO-BNN $bnn(R)$ depending on the choice of $next_spec$ as follows: $\exists \dot{A}_1, \dots, \exists \dot{A}_n. odes(\dot{R}) \wedge \exists \vec{A}_1, \dots, \exists \vec{A}_n. odes(\vec{R}) \wedge \bigwedge_{i=1}^n next_spec(A_i, \dot{A}_i, \vec{A}_i, \vec{A}_i)$

Fig. 6 gives the FO-BNN of the reaction network R_{enz} with $next_spec_3$.

Theorem 12 Soundness of $bnn(R)$. $cnext_{odes(R)} \subseteq trans \circ sol^{\mathbb{S}}(bnn(R))$.

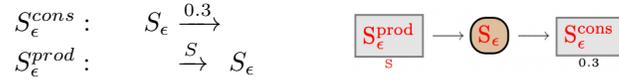
Proof Let $(\gamma_1, \gamma_2) \in cnext_{odes(R)}$. Then there exists $(\gamma'_1, \gamma'_2) \in cnext_{odes(R)}$ such that $\gamma'_1 = \gamma_1|_{\mathcal{S}}$ and $\gamma'_2 = \gamma_2|_{\mathcal{S}}$. By assumption on $next_spec(vars_A)$, this implies for all $A \in \mathcal{S}$ that $\gamma'_1 \cup \gamma'_2 \in sol^{\mathbb{S}}(\bigwedge_{A \in \mathcal{S}} next_spec(vars_A))$. Lemma 9 shows that $\gamma'_1, \gamma'_2 \in sol^{\mathbb{S}}(odes(\dot{R}))$ so that $\vec{\gamma}'_2 \in sol^{\mathbb{S}}(odes(\vec{R}))$. By definition of $bnn(R)$, we obtain $\gamma'_1|_{\mathcal{S}} \cup \gamma'_2|_{\mathcal{S}} \in sol^{\mathbb{S}}(bnn(R))$. Hence, $(\gamma_1, \gamma_2) = (\gamma'_1|_{\mathcal{S}}, \gamma'_2|_{\mathcal{S}}) \in trans \circ sol^{\mathbb{S}}(bnn(R))$ as stated by the theorem. \square

Based on the Soundness Theorem 12 we can simulate any reaction network R by abstractly simulate the Boolean network $bnn(R)$. The abstract simulation of $bnn(R_{enz})$ with $next_spec_3$ for $next_spec$, for instance, was shown earlier.

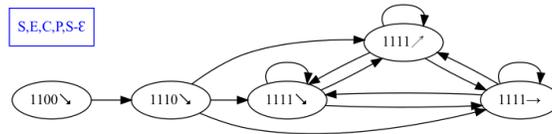
10 Thresholds

We use Booleans to distinguish whether the concentration of a species is zero or not. It often happens, though, that we would like to know whether the concentration of a species is above or below a given threshold. We now show that this can be treated with the above techniques.

Suppose we are given a species $S \in \mathcal{S}$ and a threshold $\epsilon > 0$, say $\epsilon = 0.3$, and we want to know whether the concentration of S is above, equal, or below ϵ , so whether $S - \epsilon < 0$. The idea is to add an artificial species S_ϵ to the network, such that $\dot{S}_\epsilon = S - \epsilon$. This can be done by adding the following two reactions:



The ODEs of the so extended reaction network contain the expected equation. We can thus run the abstract simulation algorithm on the extended reaction network. When applied to the reaction network R_{enz} with the same initial concentrations than above, this yields the following accessible transition graph:



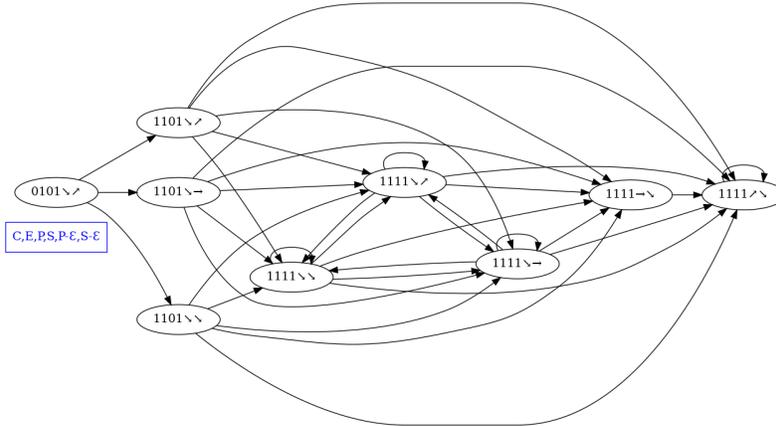


Fig. 9: Abstract simulation for the R_{enz} network with two thresholds.

In this picture, we write $S-\epsilon$ instead of \hat{S}_ϵ . The negative sign $S-\epsilon = \searrow$ means $S < \epsilon$, and a positive sign $S-\epsilon = \nearrow$ means $S > \epsilon$, and $S-\epsilon = \rightarrow$ means $S = \epsilon$.

The proper addition of thresholds, combined with the utilisation of the exact Boolean abstraction algorithm [2] for the set of linear equations of the extended reaction network, provides a considerably more fine-grained abstract simulation of the network. For example, with the addition of a further threshold for P in R_{enz} and an upper bound on the sum of the initial concentrations of S, C, P , the abstract simulation allows us to conclude that only one final state may be reachable during the abstract simulation, where the concentration of S is below the given threshold, as shown in Fig. 9. The automatic application of the exact Boolean abstraction algorithm to the simulation of Boolean networks with thresholds requires however an extension of the algorithm to the inhomogeneous case, which is under implementation. For the simulation shown in Fig. 9, a subset of the inhomogeneous equations was reduced to the homogeneous case by manual rewriting, so that the original algorithm could be applied.

11 Application to Biomodel’s Reaction Networks

Biomodels [20] is an online repository which contains a curated collection of over a thousand published models about various biological systems [20]. Most of these were previously published as ODE, but are now provided as reaction networks in the SBML format [18]. We applied abstract simulation to reaction network B448 of the Biomodels database at <https://www.ebi.ac.uk/biomodels/BIOMD000000448>. It models insulin signalling in human adipocytes in normal conditions [8]. Once converted to the BioComputing’s XML format, the network involves 27 species and 34 reactions. Its graph in Fig. 13 covers one full page, and its ODEs are given in Fig. 14. In the initial (concrete) state given in the SBML model, all the species except IRins start with a concentration strictly

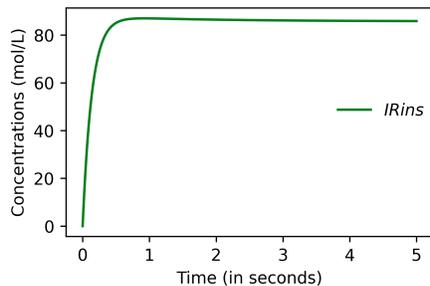


Fig. 10: The numerical simulation of B448 projected to species *IRins*.

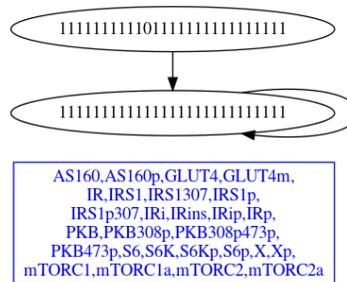


Fig. 11: Abstract simulation of B448

above 0. The full numerical simulation of the 27 species is given in Fig. 12. while Fig. 10 focuses on the concentration of *IRins* over time.

The total number of abstract states is 2^{27} . However, by starting from the abstraction of the initial concentrations in the SBML model (all species are present except *IRins*), the state transition graph is reduced to 2 edges between 2 states (Fig. 11). One of these states being the initial abstract state mentioned above, and the other is the 1-only bit vector. The latter is an attractor, as its exiting edge is making a self-loop. This is consistent with the concrete simulation of the model (Fig. 10) and the steady-state computed numerically, but independent of the precise initial concentrations chosen in the SBML model.

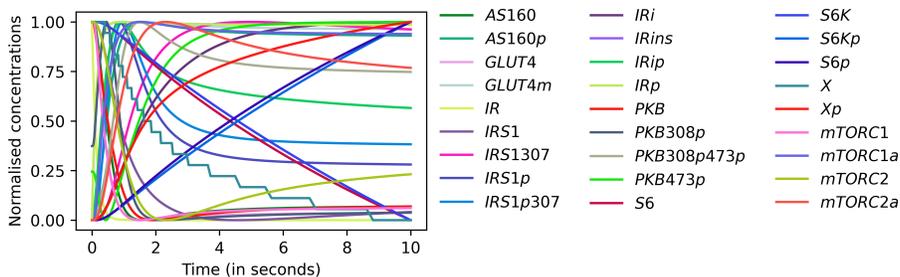


Fig. 12: The numeric simulation of B448 with all species.

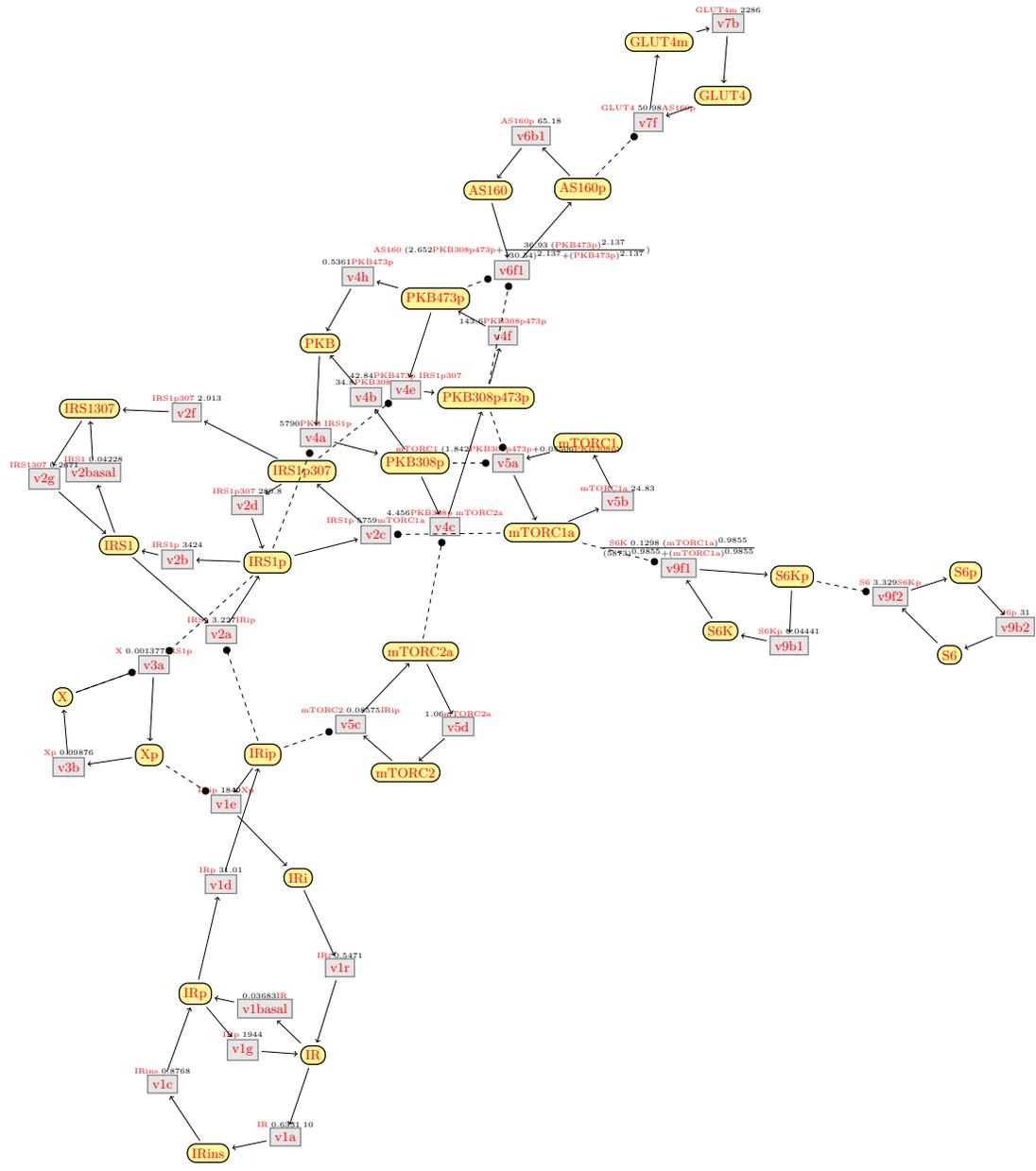


Fig. 13: The graph of reaction network B448

$$\begin{aligned}
\dot{\text{IR}} &= -r_{v1a} - r_{v1\text{basal}} + r_{v1g} + r_{v1r} & r_{v1g} &= \text{IRp } 1944 \\
\dot{\text{IRp}} &= r_{v1\text{basal}} + r_{v1c} - r_{v1d} - r_{v1g} & r_{v1r} &= \text{IRi } 0.5471 \\
\dot{\text{IRins}} &= r_{v1a} - r_{v1c} & r_{v2a} &= \text{IRS1 } 3.227\text{IRip} \\
\dot{\text{IRip}} &= r_{v1d} - r_{v1e} & r_{v2b} &= \text{IRS1p } 3424 \\
\dot{\text{IRi}} &= r_{v1e} - r_{v1r} & r_{v2c} &= \text{IRS1p } 5759\text{mTORC1a} \\
\dot{\text{IRS1}} &= -r_{v2a} + r_{v2b} - r_{v2\text{basal}} + r_{v2g} & r_{v2d} &= \text{IRS1p307 } 280.8 \\
\dot{\text{IRS1p}} &= r_{v2a} - r_{v2b} - r_{v2c} + r_{v2d} & r_{v2f} &= \text{IRS1p307 } 2.913 \\
\dot{\text{IRS1p307}} &= r_{v2c} - r_{v2d} - r_{v2f} & r_{v2\text{basal}} &= \text{IRS1 } 0.04228 \\
\dot{\text{IRS1307}} &= r_{v2f} + r_{v2\text{basal}} - r_{v2g} & r_{v2g} &= \text{IRS1307 } 0.2671 \\
\dot{\text{X}} &= -r_{v3a} + r_{v3b} & r_{v3a} &= \text{X } 0.001377\text{IRS1p} \\
\dot{\text{Xp}} &= r_{v3a} - r_{v3b} & r_{v3b} &= \text{Xp } 0.09876 \\
\dot{\text{PKB}} &= -r_{v4a} + r_{v4b} + r_{v4h} & r_{v5a} &= \text{mTORC1 } (1.842\text{PKB308p473p} + \\
\dot{\text{PKB308p}} &= r_{v4a} - r_{v4b} - r_{v4c} & & 0.05506\text{PKB308p}) \\
\dot{\text{PKB473p}} &= -r_{v4e} + r_{v4f} - r_{v4h} & r_{v5b} &= \text{mTORC1a } 24.83 \\
\dot{\text{PKB308p473p}} &= r_{v4c} + r_{v4e} - r_{v4f} & r_{v5c} &= \text{mTORC2 } 0.08575\text{IRip} \\
\dot{\text{mTORC1}} &= -r_{v5a} + r_{v5b} & r_{v5d} &= 1.06\text{mTORC2a} \\
\dot{\text{mTORC1a}} &= r_{v5a} - r_{v5b} & r_{v4a} &= 5790\text{PKB IRS1p} \\
\dot{\text{mTORC2}} &= -r_{v5c} + r_{v5d} & r_{v4b} &= 34.8\text{PKB308p} \\
\dot{\text{mTORC2a}} &= r_{v5c} - r_{v5d} & r_{v4c} &= 4.456\text{PKB308p mTORC2a} \\
\dot{\text{AS160}} &= -r_{v6f1} + r_{v6b1} & r_{v4e} &= 42.84\text{PKB473p IRS1p307} \\
\dot{\text{AS160p}} &= r_{v6f1} - r_{v6b1} & r_{v4f} &= 143.6\text{PKB308p473p} \\
\dot{\text{GLUT4m}} &= r_{v7f} - r_{v7b} & r_{v4h} &= 0.5361\text{PKB473p} \\
\dot{\text{GLUT4}} &= -r_{v7f} + r_{v7b} & r_{v6f1} &= \text{AS160 } (2.652\text{PKB308p473p} + \\
\dot{\text{S6K}} &= -r_{v9f1} + r_{v9b1} & & \frac{36.93 (\text{PKB473p})^{2.137}}{(30.54)^{2.137} + (\text{PKB473p})^{2.137}}) \\
\dot{\text{S6Kp}} &= r_{v9f1} - r_{v9b1} & r_{v6b1} &= \text{AS160p } 65.18 \\
\dot{\text{S6}} &= -r_{v9f2} + r_{v9b2} & r_{v7f} &= \text{GLUT4 } 50.98\text{AS160p} \\
\dot{\text{S6p}} &= r_{v9f2} - r_{v9b2} & r_{v7b} &= \text{GLUT4m } 2286 \\
r_{v1a} &= \text{IR } 0.6331 \text{ 10} & r_{v9f1} &= \frac{\text{S6K } 0.1298 (\text{mTORC1a})^{0.9855}}{(5873)^{0.9855} + (\text{mTORC1a})^{0.9855}} \\
r_{v1\text{basal}} &= 0.03683\text{IR} & r_{v9b1} &= \text{S6Kp } 0.04441 \\
r_{v1c} &= \text{IRins } 0.8768 & r_{v9f2} &= \text{S6 } 3.329\text{S6Kp} \\
r_{v1d} &= \text{IRp } 31.01 & r_{v9b2} &= \text{S6p } 31 \\
r_{v1e} &= \text{IRip } 1840\text{Xp} & &
\end{aligned}$$

Fig. 14: The ODEs of reaction network B448.

12 Conclusion and Future Work

We presented an algorithm that simulates a reaction network abstractly without any exact knowledge on the initial concentrations. The soundness Theorem 12 relies on the causal next relation of the reaction network, rather than on its temporal next relation. The precise relationship between these two relations seems to be related to approximation errors of Euler's numeric simulation.

One open question is whether one can compute the causal next relation exactly, similarly to the exact computation of the Boolean abstraction of linear equation systems from [2]. Another question is whether more accurate approximations may be possible than those presented here.

The next step will be to lift our abstract simulation algorithm to reaction networks for which the kinetic expressions are only partially known. The most frequent case is that some parameters of the kinetic expressions are unknown. Alternatively, the form of the kinetic expressions may be known only up to similarity [24,1]. Such networks cannot even be simulated concretely without estimating the missing kinetic information from data, so abstract simulation may provide an interesting alternative for the qualitative analysis of such networks.

We hope that the present work could be of interest to the research community of Boolean networks. We believe that FO-BNNs offer an interesting alternative to classical Boolean networks with deterministic updates. So the classical questions for Boolean networks should be reconsidered for FO-BNNs.

Acknowledgements. We thank Jun Pang and Loïc Paulevé for the helpful discussions and references on the relation of reaction and Boolean networks as well as our colleagues from BioComputing Emilie Allart, Maxime Folschette and Cédric Lhoussaine.

References

1. E. Allart, J. Niehren, and C. Versari. Computing difference abstractions of metabolic networks under kinetic constraints. In L. Bortolussi and G. Sanguinetti, editors, *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings*, volume 11773 of *Lecture Notes in Computer Science*, pages 266–285. Springer, 2019.
2. E. Allart, J. Niehren, and C. Versari. Exact Boolean Abstraction of Linear Equation Systems. *Computation*, 9(11):32, Oct. 2021.
3. R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid systems*, pages 209–229. Springer, 1992.
4. G. Batt, H. de Jong, M. Page, and J. Geiselman. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4):982–989, Apr. 2008.
5. N. Benes, L. Brim, J. Drazanova, S. Pastva, and D. Safránek. Facetal abstraction for non-linear dynamical systems based on δ -decidable SMT. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 99–108, Montreal Quebec Canada, Apr. 2019. ACM.
6. L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, July 2006.
7. P. J. Collins, L. Habets, J. H. van Schuppen, I. Cerná, J. Fabriková, and D. Safránek. Abstraction of Biochemical Reaction Systems on Polytopes. *IFAC Proceedings Volumes*, 44(1):14869–14875, Jan. 2011.
8. G. M. Di Guglielmo, P. G. Drake, P. C. Baass, F. Authier, B. I. Posner, and J. J. Bergeron. Insulin receptor internalization and signalling. *Molecular and Cellular Biochemistry*, 182(1-2):59–63, May 1998.
9. L. Euler. *Institutionum Calculi Integralis*. Number vol. 1 in *Institutionum Calculi Integralis*. imp. Acad. imp. Saënt., 1768.
10. F. Fages, S. Gay, and S. Soliman. Inferring reaction systems from ordinary differential equations. *Theor. Comput. Sci.*, 599:64–78, 2015.

11. F. Fages and S. Soliman. Abstract interpretation and types for systems biology. *Theor. Comput. Sci.*, 403(1):52–70, 2008.
12. M. Feinberg. Chemical reaction network structure and the stability of complex isothermal reactors—I. the deficiency zero and deficiency one theorems. *Chemical Engineering Science*, 42(10):2229 – 2268, 1987.
13. L. Glass and S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1):103–129, 1973.
14. S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.
15. S. Jamshidi, H. Siebert, and A. Bockmayr. Comparing Discrete and Piecewise Affine Differential Equation Models of Gene Regulatory Networks (Long Version). page 20, 2012.
16. S. Jamshidi, H. Siebert, and A. Bockmayr. Preservation of dynamic properties in qualitative modeling frameworks for gene regulatory networks. *Biosystems*, 112(2):171–179, May 2013.
17. M. John, M. Nebut, and J. Niehren. Knockout Prediction for Reaction Networks with Partial Kinetic Information. In *14th International Conference on Verification, Model Checking, and Abstract Interpretation*, pages 355–374, Rom, Italy, Jan. 2013.
18. S. M. Keating, et al. and SBML Level 3 Community members. SBML Level 3: An extensible format for the exchange and reuse of biological models. *Molecular Systems Biology*, 16(8):e9110, Aug. 2020.
19. G. Madelaine, C. Lhoussaine, and J. Niehren. Attractor Equivalence: An Observational Semantics for Reaction Networks. In *First International Conference on Formal Methods in Macro-Biology*, Lecture Notes in Bioinformatics, pages 82–101, Nouméa, New Caledonia, Sept. 2014. Springer-Verlag.
20. 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*, 48(D1):D407–D415, 2020.
21. A. Mizera, J. Pang, H. Qu, and Q. Yuan. Taming asynchrony for attractor detection in large boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1):31–42, 2019.
22. S. Mover, A. Cimatti, A. Griggio, A. Irfan, and S. Tonetta. Implicit Semi-Algebraic Abstraction for Polynomial Dynamical Systems. In A. Silva and K. R. M. Leino, editors, *Computer Aided Verification*, volume 12759, pages 529–551. Springer International Publishing, Cham, 2021.
23. N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack. Minizinc: Towards a standard CP modelling language. In C. Bessiere, editor, *Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007, Proceedings*, volume 4741 of *Lecture Notes in Computer Science*, pages 529–543. Springer, 2007.
24. J. Niehren, C. Versari, M. John, F. Coutte, and P. Jacques. Predicting Changes of Reaction Networks with Partial Kinetic Information. *BioSystems*, 149:113–124, July 2016.
25. L. Paulevé and S. Sené. Non-deterministic updates of Boolean networks. In *AUTOMATA 2021 (27th International Workshop on Cellular Automata and Discrete Complex Systems)*, Marseille, France, 2021.

26. L. Paulevé, J. Kolça, T. Chatain, and S. Haar. Reconciling Qualitative, Abstract, and Scalable Modeling of Biological Networks. *Nature Communications*, 11, 2020.
27. S. Sankaranarayanan and A. Tiwari. Relational Abstractions for Continuous and Hybrid Systems. In G. Gopalakrishnan and S. Qadeer, editors, *Computer Aided Verification*, volume 6806, pages 686–702. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
28. R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563–585, 1973.