



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. CMSB2022: International Conference on Computational Methods in Systems Biology, Sep 2022, Budapest, Romania. hal-02279942v8

HAL Id: hal-02279942

<https://hal.science/hal-02279942v8>

Submitted on 17 May 2022 (v8), last revised 24 Jun 2022 (v9)

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}, Athénaïs Vaginay³, and Cristian Versari¹

¹ BioComputing Team of CRISTAL Lab, Université de Lille, ² Inria Lille, ³ Université de Lorraine, CNRS, CRAN, INRIA, LORIA, F-54000 Nancy.

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 a boolean stating whether the species is present or absent, and the derivation of 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 proof the soundness of our abstract simulation algorithm, discuss how we implemented it, illustrate its usefulness for exact reasoning, and show its applicability to reaction networks in SBML format from the BioModels database.

Keywords: Systems biology · reaction networks · SBML · boolean networks · abstract interpretation · logic · constraint programming.

1 Introduction

Reaction networks [7,3,8,6] are the most prominent formalism for modelling the dynamics of biological system. We consider the deterministic semantics of reaction networks, which describes their dynamics by ordinary differential equations (ODEs). The solutions of ODEs are functions of type $\mathbb{R}_+ \rightarrow \mathbb{R}$ that are called trajectories. The values of trajectories for reaction networks must always be positive since they stand for concentrations of species. In contrast, the derivations of trajectories may be negative. The values of the trajectories at time point 0 are called the initial concentrations. It is well-known that for any assignment of species to some initial concentrations, there exists at most one solution of the ODEs of the reaction network. This solution can be approximated numerically by Euler's deterministic simulation algorithm [5].

The concrete 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 transition graph of bit vectors. The graph can be

enriched, when not only considering the trajectories but also their derivations. 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.

The question that we study in the present paper is whether one can compute the abstract state transition graph of a reaction network. Clearly, abstract state transition graphs are finite, but since they may have $2^{|\mathcal{S}|} + 3^{|\mathcal{S}|}$ many states, where $|\mathcal{S}|$ is the number of species, they are quickly be too big to be enumerated. 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 nontrivial, given that trajectories are infinite objects, and that there are infinitely many trajectories depending on choice of the initial concentrations.

Our idea for abstract simulation is based on the abstract interpretation of ODEs of a reaction networks over the structure of signs \mathbb{S} . This abstraction introduces non-determinism, since $\nearrow +^{\mathbb{S}} \searrow$ may be evaluated to any sign. This abstract interpretation can be proven to provide a sound over-approximation based on John’s theorem [1,13,9]. 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. 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 derivate 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 this 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 Minizinc constraint solver.

While abstract interpretation enable qualitative reasoning, we can 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 derivation of B indicates, whether the concentration of A is above, below or equal to the threshold 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 threshold of $\epsilon = 80$, then the concentration of the product may eventually become bigger than ϵ , and once this happens, it can never become smaller than ϵ again. Most interestingly, the precise initial concentration of the reaction does not matter for this argument, as long as it is above the given number $\epsilon = 80$. In this way, abstract simulation can sometimes show properties of infinitely many concrete simulations.

Last, but not least, we apply our abstract simulation algorithm to a reaction network in the SBML format of the Systems Biology Markup Language [10] supported by the BioModels database [11]. We consider 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 [4]. It has 27 species and 34 reactions and its graph covers one full page (see Fig. 13 of the appendix). Nevertheless, we could apply our abstract simulation algorithm successfully to B448, leading to a very small subgraph of the huge abstract state transition graph with more than 2^{27} states.

Outline. We start with preliminaries in Section 2. We recall the notion of reaction networks and their deterministic semantics via ODEs in Section 3. In Section 4 we recall the first-order logic, which permits to formally capture ODEs in Section 5, lays the foundation of FO-BNNs in Section 6, and enables abstract interpretation in Section 7. We present our compiler from reaction networks to FO-BNNs and prove its soundness in Section 8. The treatment of thresholds is discussed in Section 9. 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 10. The conclusion and future work are given in Section 11. The appendix contains a description of the implementation in Section 12 and further details on B448.

2 Preliminaries

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.

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 $dom(f)$. The restriction of f to a subset $A' \subseteq dom(f)$ is written as $f|_{A'}$. We write $[a_1/b_1, \dots, a_n/b_n]$ for the finite function f with $dom(f) = \{a_1, \dots, a_n\}$ and $f(a_i) = b_i$ for all $1 \leq i \leq n$. For any two sets A, B , the power set $A^B = \{f \mid f : A \rightarrow B\}$ 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 \mathcal{V} be a set of variables. The set of arithmetic expressions $e \in \mathcal{E}_{arith}(\mathcal{V})$ is the least set of terms containing the reals $\rho \in \mathbb{R}$, the variables $x \in \mathcal{V}$, and that is closed under addition, multiplication, subtraction, division, and exponentiation:

$$e, e' \in \mathcal{E}_{arith}(\mathcal{V}) ::= \rho \mid x \mid e + e' \mid e * e' \mid e - e' \mid e/e' \mid exp(e, e')$$

Arithmetic expressions have various usages and interpretations that are sum-

Level	arithmetic expressions	structure	variable assignments
CRN	kinetic expressions	reals	$\alpha : \mathcal{V} \rightarrow \mathbb{R}$
ODE	in equations	real functions	$\beta : \mathcal{V} \rightarrow (\mathbb{R}_+ \rightarrow \mathbb{R})$
BN	in equations	signs	$\gamma : \mathcal{V} \rightarrow \mathbb{S}$

Table 1: Interpretation of arithmetic expressions in various structures.

marize in Table 1. For instance, they can be used as arithmetic expressions in reactions that are interpreted over the reals. For any variable assignment to real numbers $\alpha : \mathcal{V} \rightarrow \mathbb{R}$, an arithmetic expression can be evaluated to a real number if all application of the operations for the operator are well-defined. In our formalization later on, we will define a set $\llbracket e \rrbracket^{\mathbb{R}, \alpha}$ that contains this unique real number if it exists and is empty otherwise. The evaluator over the structure of reals has to interpret any binary operator $\odot \in \{+, *, -, /, exp\}$ as a binary function of the structure of reals $\odot^{\mathbb{R}} : \mathbb{R}^2 \times \mathbb{R}$ that may be partial though. In particular the division operation $/^{\mathbb{R}}$ is only a partial function, since division by zero is not defined. In contrast $+^{\mathbb{R}}$, $*^{\mathbb{R}}$, $-^{\mathbb{R}}$, and $exp^{\mathbb{R}}$ are total functions.

Equations with arithmetic expressions are used ODEs, where they are interpreted over real-valued functions. For any variable assignment $\beta : \mathcal{V} \rightarrow (\mathbb{R}_+ \rightarrow \mathbb{R})$, an arithmetic expression e is evaluated to a real value function if all the application of all the operators for the operations are well-defined. The set defined later on $\llbracket e \rrbracket^{\mathbb{R}_+ \rightarrow \mathbb{R}, \beta}$ will contains this unique real function if it exists and is empty otherwise. For this we interpret the operators $\odot \in \{+, *, -, /, exp\}$ as operations on real valued functions $\odot^{\mathbb{R}_+ \rightarrow \mathbb{R}} : (\mathbb{R}_+ \rightarrow \mathbb{R})^2 \times (\mathbb{R}_+ \rightarrow \mathbb{R})$. Note that the division operation of real-valued functions $/^{\mathbb{R}_+ \rightarrow \mathbb{R}}$ inherits the partiality of the division operation on the reals $/^{\mathbb{R}}$. And finally notice, that the constants $\rho \in \mathbb{R}$ in arithmetic expressions must be reinterpreted as constant functions.

Furthermore, we will abstractly interpret equations with arithmetic expressions over the structure of signs. For any variable assignment to signs $\gamma : \mathcal{V} \rightarrow \mathbb{S}$, an arithmetic expression e is evaluated nondeterministically, that is to a set of possible signs $\llbracket e \rrbracket^{\mathbb{R}_+ \rightarrow \mathbb{R}, \gamma}$. For this we interpret the operators $\odot \in \{+, *, -, /, exp\}$ as the naturally corresponding operations on signs $\odot^{\mathbb{S}} : \mathbb{S}^2 \times \mathbb{S}$. We note that the addition operator is not even functional, give that $1 + (-1) = \nearrow + \searrow$ can be evaluated to any sign. Furthermore, any constant $\rho \in \mathbb{R}$ is now to be reinterpreted as the sign $h_{\mathbb{S}}(\rho)$ where $h_{\mathbb{S}} : \mathbb{R} \rightarrow \mathbb{S}$ so that $h_{\mathbb{S}}(\rho) = 1$ if $\rho > 0$, $h_{\mathbb{S}}(\rho) = 0$ if $\rho = 0$ and $h_{\mathbb{S}}(\rho) = -1$ if $\rho < 0$. The interpretation of \mathbb{S} turns $h_{\mathbb{S}} : \mathbb{R} \rightarrow \mathbb{S}$ into a homomorphism between Σ_{arith} -structures, satisfying $\odot^{\mathbb{S}} = \{(h_{\mathbb{S}}(\rho), h_{\mathbb{S}}(\rho'), h_{\mathbb{S}}(\rho'')) \mid (\rho, \rho', \rho'') \in \odot^{\mathbb{R}}\}$ for any operator $\odot \in \Sigma_{arith}$.

3 Chemical Reaction Networks

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

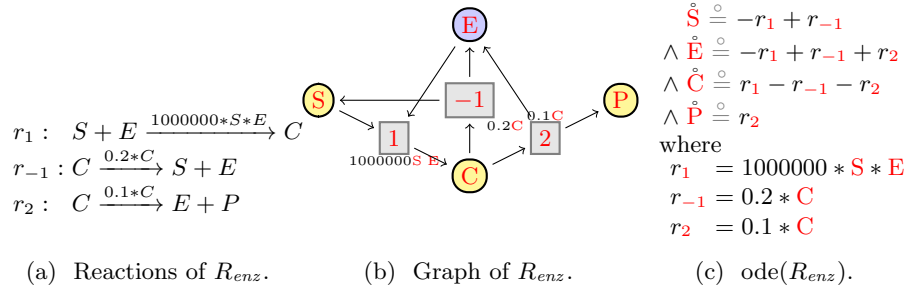


Fig. 1: The enzymatic reaction network R_{enz} .

Definition 1. A (chemical) 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 (chemical) reaction network with species in \mathcal{S} is a subset R of reactions with species in \mathcal{S} .

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 $\llbracket e \rrbracket^{\mathbb{R}, \alpha} = 5.1 *^{\mathbb{R}} (\llbracket A \rrbracket^{\mathbb{R}, \alpha})^2 *^{\mathbb{R}} \llbracket B \rrbracket^{\mathbb{R}, \alpha}$. 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 $A \in \mathcal{S}$ a species, the stoichiometry of A in r is defined by $stoic_r(A) = M'(A) - M(A)$. The ODE of a reaction network R is the following equation system:

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

A formal definition of the syntax and semantics of ODEs will be given in Section 5 based on notions from the first-order logic in Section 4. 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 derivation of the denotation of A if its derivation exists, and is undefined otherwise. Note that derivations 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 for a CRN, we show the network of enzymatic reactions in Fig. 1. It has species $\mathcal{S} = \{S, E, C, P\}$ and the three reactions in Fig. 1a, 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. 1b and its ODEs in Fig. 1c.

4 First-Order Logic

We call the syntax and semantics of first-order logic formulas. It will lay a common foundation for defining ODEs and boolean networks with non-deterministic updates, and serve in all algorithms of the present paper.

A ranked signature is a set $\Sigma = C \cup \bigcup_{n \geq 1} F^{(n)}$ that contains a subset of constants $c \in C$ and subsets of function symbols $F^{(n)}$ of arity n for all $n \geq 1$. We assume that all these subsets are pairwise distinct. For instance, we consider the signature of arithmetics $\Sigma_{arith} = \mathbb{R} \cup F^{(2)}$ where $F^{(2)} = \{+, *, -, /, exp\}$ and $\dot{\Sigma}_{arith} = \Sigma_{arith} \cup F^{(1)}$ where $F^{(1)} = \{\cdot\}$.

A relational Σ -structure is a pair $S = (dom(S), \cdot^S)$ such that the interpretation \odot^S is a relation $f^S \subseteq dom(S)^n \times dom(S)$ for any $\odot \in F^{(n)}$. If all interpretations are total functions, then we call S a Σ -algebra. In concrete examples, we will often confuse a Σ -structure with its domain. Our first example is the set of reals which forms a Σ_{arith} -structure which we call \mathbb{R} , equally to its domain. Note that $/^{\mathbb{R}}$ is only partial function since division by zero not defined. Therefore, \mathbb{R} is not an Σ_{arith} -algebra. The second example is the set of total functions of type $\mathbb{R}_+ \rightarrow \mathbb{R}$. It forms a $\dot{\Sigma}_{arith}$ -structure that we call $\mathbb{R}_+ \rightarrow \mathbb{R}$. The interpretation of the dot operator $\cdot \in F^{(1)}$ in this structure is the operation that maps any real-valued function of type $\mathbb{R}_+ \rightarrow \mathbb{R}$ to its derivative if it exists, and is undefined otherwise. The set of signs forms a Σ_{arith} -structure with domain \mathbb{S} that we also call \mathbb{S} . It is not a Σ -algebra, since $+^{\mathbb{S}}$ is not functional. In particular, $(-1) + 1$ can be evaluated non-deterministically to any sign of \mathbb{S} .

Let \mathcal{V} be a set of variables. The set of Σ -expressions with variables in \mathcal{V} is defined by the following abstract syntax

$$e \in \mathcal{E}_{\Sigma}(\mathcal{V}) ::= x \mid c \mid \odot(e_1, \dots, e_n) \quad \text{where } x \in \mathcal{V}, c \in C \text{ and } \odot \in F^{(n)}$$

By definition the set of arithmetic expressions satisfies $\mathcal{E}_{arith}(\mathcal{S}) = \mathcal{E}_{\Sigma_{arith}}(\mathcal{S})$. For any expression e , the set of free variables $fv(e)$ is the set of all variables that occur in e . The semantics of an expression $e \in \mathcal{E}_{\Sigma}(\mathcal{V})$ is defined in Fig. 2 as a subset of values of the domain $\llbracket e \rrbracket^{\alpha, S} \subseteq dom(S)$. It depends on some relational Σ -structure S for interpreting the operators and a variable assignment into the domain of this structure $\alpha : V \rightarrow dom(S)$ with $fv(e) \subseteq V$.

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

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

We sometimes use shortcuts $e \geq 0$ for the formula $\exists x. e \stackrel{\circ}{=} x^2$ and $e \leq e'$ for $e' - e \geq 0$. We also write $\mathcal{F}_{arith} = \mathcal{F}_{\Sigma_{arith}}$ for the set of arithmetic formulas build from arithmetic expressions. 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 Σ -structure S and variable assignment $\alpha : V \rightarrow dom(S)$ with $fv(\phi) \subseteq V$. 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 Σ -structure S is $sol^S(\phi) = \{\alpha : fv(\phi) \rightarrow dom(S) \mid \llbracket \phi \rrbracket^{\alpha, S} = 1\}$.

$$\begin{aligned} \llbracket e \rrbracket^{\alpha, S} &= c^S & \llbracket x \rrbracket^{\alpha, S} &= \{\alpha(x)\} \\ \llbracket \odot(e_1, \dots, e_n) \rrbracket^{\alpha, S} &= \{s \mid (s_1, \dots, s_n, s) \in \odot^S, s_i \in \llbracket e_i \rrbracket^{\alpha, S} \text{ for all } 1 \leq i \leq n\} \text{ where } \odot \in F^{(n)} \end{aligned}$$

Fig. 2: Interpretation of expressions $e \in \mathcal{E}_\Sigma(\mathcal{V})$ over a relational Σ -structure S with variable assignment $\alpha : V \rightarrow \text{dom}(S)$ where $\text{fv}(e) \subseteq V$.

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

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

5 ODEs

We now define ordinary differential equations (ODEs) as formulas of first-order logic in order to formalize their syntax and semantics in a framework suitable for abstract interpretation.

Definition 2. *An ODE is a formula of first-order logic with signature $\dot{\Sigma}_{arith}$.*

The semantics of an ODE ϕ is the set of its solution over $\dot{\Sigma}_{arith}$ -structure of real-valued functions, i.e., $\text{sol}^{\mathbb{R}^+ \rightarrow \mathbb{R}}(\phi)$. For any choice of initial concentrations $\alpha : \mathcal{S} \rightarrow \mathbb{R}$, there exists at most one solution $\beta \in \text{sol}^{\mathbb{R}^+ \rightarrow \mathbb{R}}(\phi)$, such that $\beta(x)(0) = \alpha(x)$ for all $x \in \text{fv}(\phi)$. This solution can be computed numerically by the usual

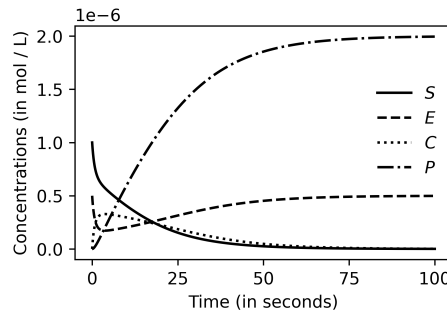


Fig. 4: The deterministic numerical simulation of $ode(R_{enz})$ with initial concentrations $S(0) = 1.0 * 10^{-5}$, $E(0) = 0.5 * 10^{-5}$ and $P(0) = C(0) = 0$ mol/L.

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 of an ODE ϕ is a real solution in $sol^{\mathbb{R}}(\phi)$. An abstract state is a sign solution in $sol^{\mathbb{S}}(\phi)$. We next show how to define a successor relation on abstract states for ODEs.

Definition 3. Let $\gamma_1, \gamma_2 : \mathcal{S} \rightarrow \mathbb{S}$ be two abstract states of ϕ . We call γ_2 a 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 $ode(R_{enz})$, 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 must 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 $next_{ode(R_{enz})}$.

The states of $ode(\phi)$ do not contain information about the values of the derivates. In order to change this, let $\circ : \mathcal{V} \rightarrow \mathcal{V}$ be a bijection such that $\mathring{\mathcal{S}}$ is disjoint from \mathcal{S} for any $\mathcal{S} \subseteq \mathcal{V}$. For any formula ϕ we define a formula $\mathring{\phi}$ assigning the values of the derivation \dot{x} to the variable \mathring{x} :

$$\mathring{\phi} =_{\text{def}} \phi \wedge \bigwedge_{x \in \text{fv}(\phi)} \mathring{x} \overset{\circ}{=} \dot{x}$$

When interested in derivates we consider the successor relation of the formula $next_{\mathring{\phi}}$. For instance, with respect to $next_{ode(\mathring{R}_{enz})}$, the abstract state $\gamma_1 = [S/1, E/1, C/0, P/0, \mathring{S}/\searrow, \mathring{E}/\searrow, \mathring{C}/\nearrow, \mathring{P}/\rightarrow]$ has the successor $\gamma_2 = [S/1, E/1, C/1, P/1, \mathring{S}/\searrow, \mathring{E}/\searrow, \mathring{C}/\nearrow, \mathring{P}/\nearrow]$. Furthermore, causality can be observed in the signs of the derivates: we have $\gamma_1(\mathring{C}) = \nearrow$ since $\gamma_1(E) = \gamma_1(S) = 1$. In contrast we have $\gamma_1(\mathring{P}) = \rightarrow$ since $\gamma_1(C) = 0$. As a consequence, for any solution $\beta \in sol^{\mathbb{R}+ \rightarrow \mathbb{R}}(ode(\mathring{R}_{enz}))$, 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 4. Let R be a reaction network and $\gamma_1, \gamma_2 : (\mathcal{S} \cup \mathring{\mathcal{S}}) \rightarrow \mathbb{S}$ abstract states. We call γ_2 a causally-next dotted state of γ_1 and write $(\gamma_1, \gamma_2) \in cnext_{ode(R)}$ if $\gamma_2 \in sol^{\mathbb{S}}(ode(\mathring{R}))$ and there exists an abstract state γ'_2 such that $(\gamma_1, \gamma'_2) \in next_{ode(\mathring{R})}$ and for all $A \in \mathcal{S}$: $\gamma_2(A) = \gamma_1(A)$ if $\gamma'_2(\mathring{A}) = 0$ and $\gamma_2(A) = \gamma'_2(A)$ otherwise. We say that $\gamma_{2|\mathcal{S}}$ is a causally-next state of $\gamma_{1|\mathcal{S}}$ and write $(\gamma_{1|\mathcal{S}}, \gamma_{2|\mathcal{S}}) \in cnext_{ode(R)}$ if $(\gamma_1, \gamma_2) \in cnext_{ode(\mathring{R})}$.

For $ode(R_{enz})$, 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 have $next_{ode(R)} \subseteq (cnext_{ode(R)})^*$ for many reaction networks R . We believe

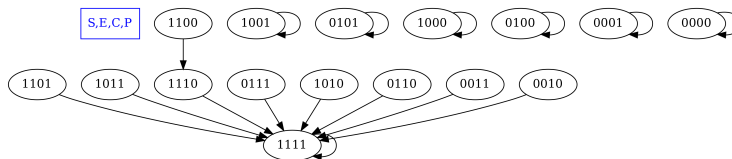


Fig. 5: The full state transition graph of the FO-BNN $bnn(R_{enz})$ in Fig. 6.

that this holds more generally for any reaction network 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 either A or E were absent). In reality, P should be set to a small nonzero value at the next step.

The question of the present paper is how to perform some kind of abstract simulation of a reaction network R , without knowing the exact initial concentrations. Instead, the sign abstractions of the initial concentrations should be given only. These signs are always booleans given that all concentrations are positive. The idea to do so is to over-approximate the relation $next_{ode(R)}$ between abstract states, by using the information about signs of the derivatives in $next_{ode^{\circ}(R)}$. For this we will abstract reaction networks to boolean networks, by abstractly interpreting the ODEs of the reaction network over the structure of signs. This will yield boolean networks with non-deterministic updates, reflecting the non-determinism of the structure of signs introduced by the sign abstraction.

6 Boolean Networks with Non-deterministic Updates

Any (abstract) state in $\mathbb{B}^{\mathcal{S}}$ is a function $\beta : \mathcal{S} \rightarrow \mathbb{B}$ 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 [14], a boolean network B with non-deterministic updates (BNN) and species in \mathcal{S} is generally some kind of definition of a abstract state transition graph $B \subseteq \mathbb{B}^{\mathcal{S}} \times \mathbb{B}^{\mathcal{S}}$ 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} , $\vec{\mathcal{S}}$, $\vec{\mathcal{S}}$, and $\vec{\mathcal{S}}$ 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 derivation at the next time point $\vec{\dot{A}}$.

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

where

$$\begin{aligned}
r_1 &= 1000000 * S * E & \vec{r}_1 &= 1000000 * \vec{S} * \vec{E} \\
r_{-1} &= 0.2 * C & \vec{r}_{-1} &= 0.2 * \vec{C} \\
r_2 &= 0.1 * C & \vec{r}_2 &= 0.1 * \vec{C}
\end{aligned}$$

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

Definition 5. An FO-BNN with variables in $V \subseteq \mathcal{V}$ is a first-order formula $\phi \in \mathcal{F}_{arith}$ with free variables $fv(\phi) = V \cup \vec{V}$.

We notice that variables \dot{A} may occur in ϕ but not any expression with a derivation operator such as \dot{A} since it is not in the signature. Any variable assignment $\gamma : V \cup \vec{V} \rightarrow \mathbb{S}$ yields a abstract state transition:

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

For illustration, a 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 $\dot{\mathcal{S}} \cup \vec{\mathcal{S}}$ 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 \mathbb{S} in Minizinc, which allows us to compute the transition graph of FO-BNNs, i.e., the relation 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, the subgraph accessible by any boolean state contain at most 3 boolean states.

The abstract simulation of the FO-BNN for the reaction network R_{enz} starting with the abstract state $[S/1, E/1, C/0, P/0]$ is given in Fig. 8. This example illustrates, that abstract simulation is related to the causality rather than the temporality of species production. For instance the temporal transition $(1100, 1111) \in next_{ode}(R_{enz})$ is represented by two causal edges $1100 \rightarrow 1110 \rightarrow 1111$ in the abstract simulation. These show the causality of the production: C is produce 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 produce shortly after C there depending on the step width that is admitted. The reachable subgraph can be computed by repeated constraint solving. In each step, the values of the variables in \mathcal{S} is 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. 7. The algorithm ends when no new reachable states are obtained from the available transitions.

```

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

```

Fig. 7: 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} .

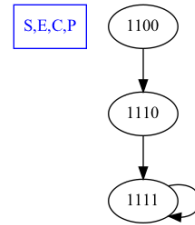


Fig. 8: Abstract simulation of the FO-BNN in Fig. 6 for the reaction network R_{enz} .

7 Abstract Interpretation of Logic Formulas

We discuss how to interpret logic formulas abstractly and recall John’s soundness theorem for this abstract interpretation [9].

Theorem 6 John’s Soundness Theorem [1,13,9]. *For any homomorphism $h : S \rightarrow \Delta$ between Σ -structures and any negation-free formula $\phi \in \mathcal{F}_{\Sigma}(\mathcal{V})$: $h \circ sol^S(\phi) \subseteq sol^{\Delta}(\phi)$.*

Lemma 7. *If $(\gamma_1, \gamma_2) \in \text{next}_{\circlearrowleft}$ then $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^{\mathbb{S}}(\phi)$ and if $(\gamma_1, \gamma_2) \in \text{cnext}_{\text{ode}(R)}$ then $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^{\mathbb{S}}(\text{ode}(R))$.*

8 Abstract Simulation of Reaction Networks

In order 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 $A, \mathring{A}, \vec{A}, \overrightarrow{A}$.

Our objective is to approximate the relation $\text{cnext}_{\text{ode}(R)}$ 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} :

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

There are several possibilities to define next_spec of which we propose three in Fig. 9. It is not difficult to see that $\text{next_spec}_1(\text{vars}_A)$ satisfy the above re-

$$\begin{aligned} \text{next_spec}_1(\text{vars}_A) &=_{\text{def}} (\vec{A} \stackrel{\circ}{=} 1 \rightarrow (A \stackrel{\circ}{=} 0 \wedge \mathring{A} \stackrel{\circ}{=} 1) \vee A \stackrel{\circ}{=} 1)) \\ &\quad \wedge (\vec{A} \stackrel{\circ}{=} 0 \rightarrow (A \stackrel{\circ}{=} 0 \wedge \mathring{A} \stackrel{\circ}{=} 0) \vee \vec{A} \stackrel{\circ}{=} -1)) \\ \text{next_spec}_2(\text{vars}_A) &=_{\text{def}} \text{next_spec}_1(\text{vars}_A) \\ &\quad \wedge (\vec{A} \stackrel{\circ}{=} 0 \rightarrow ((A \stackrel{\circ}{=} 0 \wedge \mathring{A} \stackrel{\circ}{=} 0) \vee (A \stackrel{\circ}{=} 1 \wedge \mathring{A} \stackrel{\circ}{=} -1 \wedge \vec{A} \stackrel{\circ}{=} 1))) \\ \text{next_spec}_3(\text{vars}_A) &=_{\text{def}} \text{next_spec}_1(\text{vars}_A) \wedge (\vec{A} \stackrel{\circ}{=} 0 \rightarrow A \stackrel{\circ}{=} 0) \end{aligned}$$

Fig. 9: Three possiblility for $\text{next_spec}(\text{vars}_A)$.

quirement 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 then the derivation must requires an increase immediatly after in order to not become negative. If all reactions follows the mass action law then nonzero concentrations can never become zero later on, so $\text{next_spec}_3(\text{vars}_A)$ should satisfy the requirement too.

Let $\text{next_spec}(\text{vars}_A)$ be one of the three formulas $\text{next_spec}_i(\text{vars}_A)$ 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 8. *For any reaction network R with species $\mathcal{S} = \{A_1, \dots, A_n\}$ we define the FO-BNN $\text{bnn}(R)$ depending on the choice of next_spec as follows:*

$$\exists \mathring{A}_1, \dots, \exists \mathring{A}_n. \text{ode}^{\circlearrowleft}(R) \wedge \exists \vec{A}_1, \dots, \exists \vec{A}_n. \text{ode}^{\circlearrowright}(\vec{R}) \wedge \bigwedge_{i=1}^n \text{next_spec}(A_i, \mathring{A}_i, \vec{A}_i, \overrightarrow{A}_i)$$

For illustration, the FO-BNN of the reaction network R_{enz} with $next_spec_3$ for $next_spec$ is given in Fig. 6. The following theorem states the soundness of the construction of $bnn(R)$.

Theorem 9 Soundness. $cnext_{ode(R)} \subseteq trans \circ sol^{\mathbb{S}}(bnn(R))$.

Based on the Soundness Theorem 12 we can simulate any reaction network R abstractly without knowing the exact initial concentrations: it is sufficient to abstractly simulating the boolean network $bnn(R)$. The abstract simulation of $bnn(R_{enz})$ with $next_spec_3$ for $next_spec$, for instance, was shown earlier.

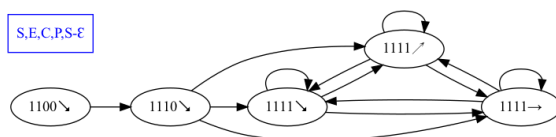
9 Thresholds

Booleans serve us 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:



In this picture, we write $S - \epsilon$ instead of \dot{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 the appendix in Fig. 12. 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. 12, a subset of the inhomogeneous equations was reduced to the homogeneous case by manual rewriting, so that the original algorithm could be applied.

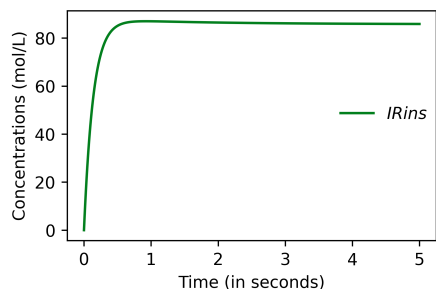


Fig. 10: The numerical simulation of B448 projected to species *IRins*.

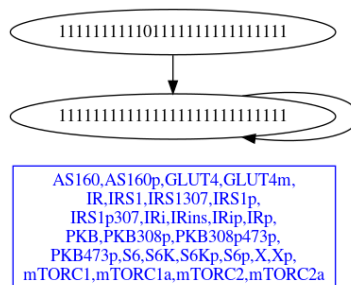


Fig. 11: Abstract simulation of B448

10 Application to Biomodel’s Reaction Networks

Biomodels [11] is an online repository which contains a curated collection of over a thousand published models about various biological systems [11]. Most of these were previously published as ODE, but are now provided as reaction networks in the SBML format [10].

We have implemented a tool chain to perform abstract simulation of reaction networks given in BioComputings XML format that is described in Section 12 of the appendix. We programmed an XSL stylesheet that can convert SBML models to reaction networks in BioComputing’s XML format. In SBML, no x-y-coordinates are available for placing the species and reactions of the graph. So we added them semi-automatically. In the graphs, any species that activates a reaction is connected to the reaction by a dashed arrow with a \bullet head.

We applied abstractly 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 [4]. Once converted in the BioComputing’s XML format, the network involves 27 species and 34 reactions. Its graph in Fig. 13 covers one full page. See Fig. 15 for its ODEs. In the initial (concrete) state given in the SBML model, all the species except *IRins* start with a concentration strictly above 0. The full numerical simulation of the 27 species is given in Fig. 14 of the appendix, 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 bitvector. 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.

11 Conclusion and Future Work

We presented an algorithm that can abstractly simulate a reaction network without exact knowledge on the initial concentrations. We applied it to a real-world reaction network from the BioModels repository with 27 species and 2^{27} abstract spaces. Nevertheless, abstract simulation based on finite domain constraint programming remains feasible, so the question is how far.

Our algorithm is based on abstract interpretation of logical formulas in the structure of signs. Therefore, we only obtain an over-approximation of the causal next relation of a reaction network. Furthermore, the precision of our algorithm depends on the choice of the formula $next_spec(vars_A)$ in particular. The more restrictive variants of this formula, however, may not be applicable to general reaction networks. One question is whether one can compute the causal next relation exactly in some cases, 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.

Theorem 12 on the soundness of abstract simulation relies on the causal next relation of the reaction network, rather than on its temporal next relation. The precise relationship between these two relations is to be elaborated. It seems, however, that it is related to the irrelevance of approximation errors of Euler’s numeric simulation algorithm for ODEs.

The next step will be to lift our novel 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 [13,1]. 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. Finally, we would like to notice that we avoided any discussions about the various semantics of classical boolean networks (synchronous, asynchronous, most permissive, etc). We did so since we could not see how the traditional boolean networks with deterministic updates could enable a sound abstraction of the deterministic semantics of a reaction network. But may be, it is possible to abstract FO-BNNs further to some classical boolean network with deterministic updates, possibly under some restrictions.

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. 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.
 4. 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.
 5. L. Euler. *Institutionum Calculi Integralis*. Number vol. 1 in *Institutionum Calculi Integralis*. imp. Acad. imp. Saënt., 1768.
 6. F. Fages, S. Gay, and S. Soliman. Inferring reaction systems from ordinary differential equations. *Theor. Comput. Sci.*, 599:64–78, 2015.
 7. 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.
 8. 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.
 9. 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.
 10. S. M. Keating, D. Waltemath, M. König, F. Zhang, A. Dräger, C. Chaouiya, F. T. Bergmann, A. Finney, C. S. Gillespie, T. Helikar, S. Hoops, R. S. Malik-Sheriff, S. L. Moodie, I. I. Moraru, C. J. Myers, A. Naldi, B. G. Olivier, S. Sahle, J. C. Schaff, L. P. Smith, M. J. Swat, D. Thieffry, L. Watanabe, D. J. Wilkinson, M. L. Blinov, K. Begley, J. R. Faeder, H. F. Gómez, T. M. Hamm, Y. Inagaki, W. Liebermeister, A. L. Lister, D. Lucio, E. Mjolsness, C. J. Proctor, K. Raman, N. Rodriguez, C. A. Shaffer, B. E. Shapiro, J. Stelling, N. Swainston, N. Tanimura, J. Wagner, M. Meier-Schellersheim, H. M. Sauro, B. Palsson, H. Bolouri, H. Kitano, A. Funahashi, H. Hermjakob, J. C. Doyle, M. Hucka, 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.
 11. 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.
 12. 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.
 13. 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.
 14. 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.

12 Implementation

We have implemented a whole tool chain to perform abstract simulation of reaction networks extending on BioComputing Reaction-Network tool. The latter can analyse reaction networks in BioComputing’s XML format based on a pipeline of XSL stylesheets. In particular it can draw the graphs reaction networks such as presentd in the present paper. It can also compute the ODEs $ode(R)$ for any reaction network R . The ODEs are represented in an XML format too, so that they can be processed further with XSL stylesheets.

We implemented a new XSL stylesheet for the treatment of thresholds by adding auxiliary species and reaction to a network. We implemented yet another XSL stylesheet that eliminates the arithmetic operators $-$ and $/$ in ODEs, so that they use only the arithmetic operators in $\Sigma = \{+, *, exp\}$. Terms with the derivation operators \dot{A} are replaced by variables A for all species A . We also wrote a stylesheet that maps any first-order Σ -formula ϕ to a Minizinc program [12], by which the solutions of ϕ over the finite structure of signs can be computed.

We implemented the compiler from reaction networks R to FO-BNNs $bnn(R)$ based on $ode(R)$ by yet another XSL stylesheet. It is parameterised by some of the formulas $next_spec_i$ where $i = 1, 2, 3$. We implemented our abstract simulation algorithm for FO-BNNs in Python, based on the constraint solver for the structure of signs. Last but not least, we wrote an XSL stylesheet for drawing the state transition graph obtained from the solution set of constraint over the structure \mathbb{S} via graphviz. All state transition graphs presented in this paper are produced in this manner.

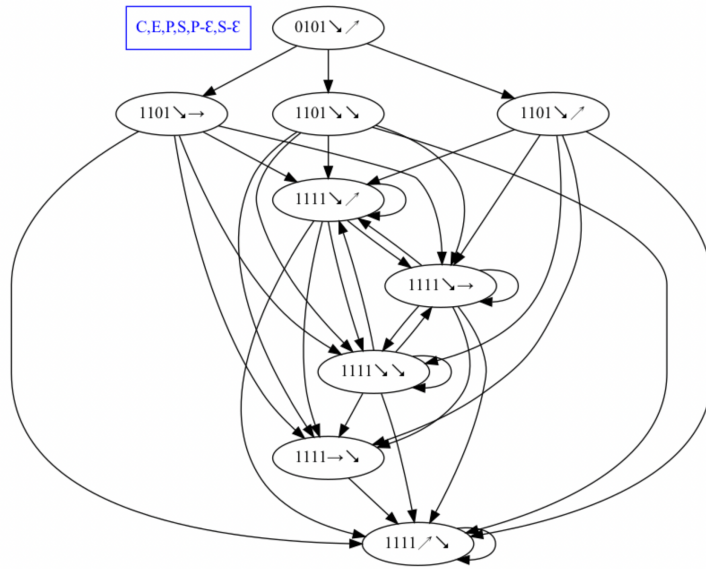


Fig. 12: Abstract simulation for the R_{enz} network with two thresholds and bound on the initial concentration of S, C, P .

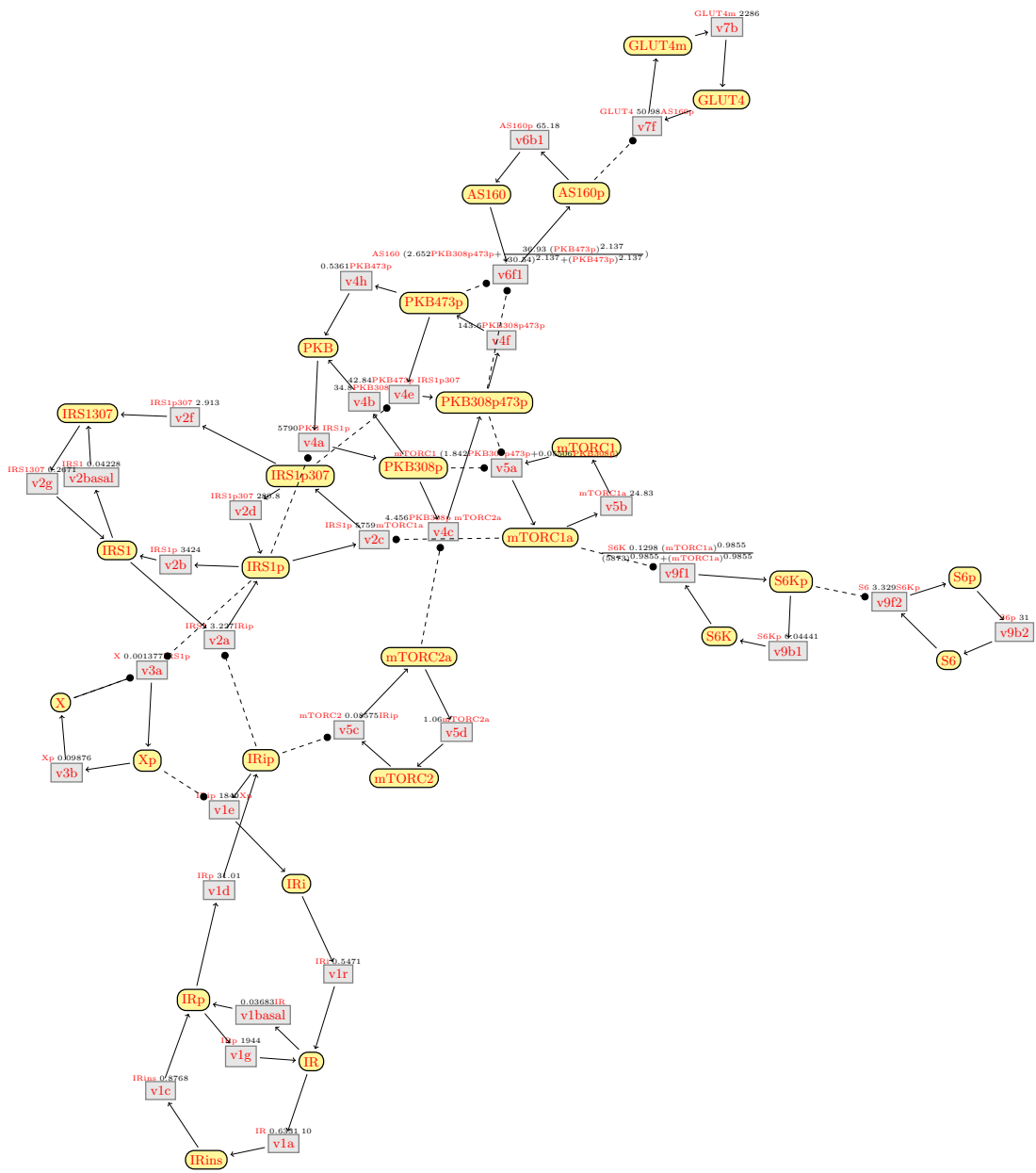


Fig. 13: The graph of reaction network B448

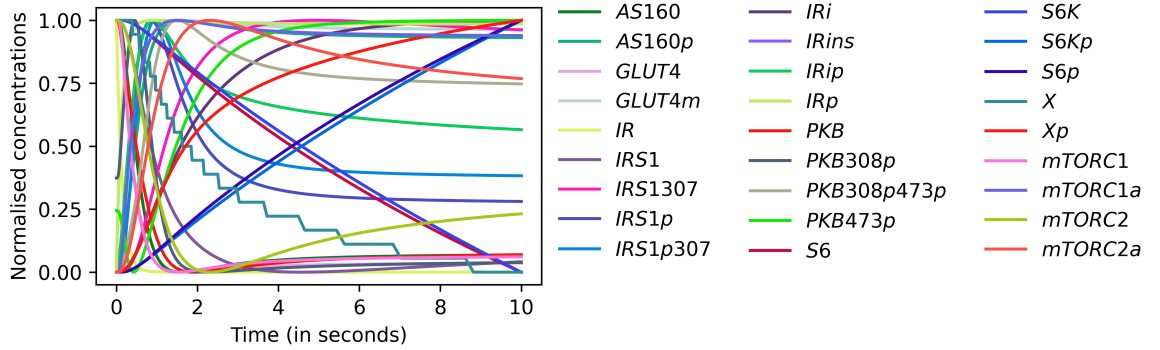


Fig. 14: The numeric simulation of B448 with all species.

$$\begin{aligned}
\dot{IR} &= -r_{v1a} - r_{v1basal} + r_{v1g} + r_{v1r} & r_{v1g} &= IRp \ 1944 \\
\dot{IRp} &= r_{v1basal} + r_{v1c} - r_{v1d} - r_{v1g} & r_{v1r} &= IRI \ 0.5471 \\
\dot{IRins} &= r_{v1a} - r_{v1c} & r_{v2a} &= IRS1 \ 3.227IRip \\
\dot{IRip} &= r_{v1d} - r_{v1e} & r_{v2b} &= IRS1p \ 3424 \\
\dot{IRi} &= r_{v1e} - r_{v1r} & r_{v2c} &= IRS1p \ 5759mTORC1a \\
\dot{IRS1} &= -r_{v2a} + r_{v2b} - r_{v2basal} + r_{v2g} & r_{v2d} &= IRS1p307 \ 280.8 \\
\dot{IRS1p} &= r_{v2a} - r_{v2b} - r_{v2c} + r_{v2d} & r_{v2f} &= IRS1p307 \ 2.913 \\
\dot{IRS1p307} &= r_{v2c} - r_{v2d} - r_{v2f} & r_{v2basal} &= IRS1 \ 0.04228 \\
\dot{IRS1307} &= r_{v2f} + r_{v2basal} - r_{v2g} & r_{v2g} &= IRS1307 \ 0.2671 \\
\dot{X} &= -r_{v3a} + r_{v3b} & r_{v3a} &= X \ 0.001377IRS1p \\
\dot{Xp} &= r_{v3a} - r_{v3b} & r_{v3b} &= Xp \ 0.09876 \\
\dot{PKB} &= -r_{v4a} + r_{v4b} + r_{v4h} & r_{v5a} &= mTORC1 \ (1.842PKB308p473p + \\
\dot{PKB308p} &= r_{v4a} - r_{v4b} - r_{v4c} & & \ 0.05506PKB308p) \\
\dot{PKB473p} &= -r_{v4e} + r_{v4f} - r_{v4h} & r_{v5b} &= mTORC1a \ 24.83 \\
\dot{PKB308p473p} &= r_{v4c} + r_{v4e} - r_{v4f} & r_{v5c} &= mTORC2 \ 0.08575IRip \\
\dot{mTORC1} &= -r_{v5a} + r_{v5b} & r_{v5d} &= 1.06mTORC2a \\
\dot{mTORC1a} &= r_{v5a} - r_{v5b} & r_{v4a} &= 5790PKB \ IRS1p \\
\dot{mTORC2} &= -r_{v5c} + r_{v5d} & r_{v4b} &= 34.8PKB308p \\
\dot{mTORC2a} &= r_{v5c} - r_{v5d} & r_{v4c} &= 4.456PKB308p \ mTORC2a \\
\dot{AS160} &= -r_{v6f1} + r_{v6b1} & r_{v4e} &= 42.84PKB473p \ IRS1p307 \\
\dot{AS160p} &= r_{v6f1} - r_{v6b1} & r_{v4f} &= 143.6PKB308p473p \\
\dot{GLUT4m} &= r_{v7f} - r_{v7b} & r_{v4h} &= 0.5361PKB473p \\
\dot{GLUT4} &= -r_{v7f} + r_{v7b} & r_{v6f1} &= AS160 \ (2.652PKB308p473p + \\
\dot{S6K} &= -r_{v9f1} + r_{v9b1} & & \ \frac{36.93 \ (PKB473p)^{2.137}}{(30.54)^{2.137} + (PKB473p)^{2.137}}) \\
\dot{S6Kp} &= r_{v9f1} - r_{v9b1} & r_{v6b1} &= AS160p \ 65.18 \\
\dot{S6} &= -r_{v9f2} + r_{v9b2} & r_{v7f} &= GLUT4 \ 50.98AS160p \\
\dot{S6p} &= r_{v9f2} - r_{v9b2} & r_{v7b} &= GLUT4m \ 2286 \\
r_{v1a} &= IR \ 0.6331 \ 10 & r_{v9f1} &= \frac{S6K \ 0.1298 \ (mTORC1a)^{0.9855}}{(5873)^{0.9855} + (mTORC1a)^{0.9855}} \\
r_{v1basal} &= 0.03683IR & r_{v9b1} &= S6Kp \ 0.04441 \\
r_{v1c} &= IRins \ 0.8768 & r_{v9f2} &= S6 \ 3.329S6Kp \\
r_{v1d} &= IRp \ 31.01 & r_{v9b2} &= S6p \ 31 \\
r_{v1e} &= IRip \ 1840Xp & &
\end{aligned}$$

Fig. 15: The ODEs of reaction network B448.

A Proofs for Section 7 (Abstract Interpretation of Logic Formulas)

Theorem 10 John's Soundness Theorem [1,13,9]. *For any homomorphism $h : S \rightarrow \Delta$ between Σ -structures and any negation-free formula $\phi \in \mathcal{F}_\Sigma(\mathcal{V})$: $h \circ \text{sol}^S(\phi) \subseteq \text{sol}^\Delta(\phi)$.*

We only give a sketch of the proof. Let $\alpha : V \rightarrow \text{dom}(S)$. For any expression $e \in \mathcal{E}_\Sigma(\mathcal{V})$ such that $\text{fv}(e) \subseteq V$ 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 $\phi \in \mathcal{F}_\Sigma(\mathcal{V})$ with $\text{fv}(\phi) \subseteq V$ that $\llbracket \phi \rrbracket^{\alpha, S} \leq \llbracket \phi \rrbracket^{h \circ \alpha, \Delta}$. This is equivalent to that: $\{h \circ \alpha \mid \alpha \in \text{sol}_V^S(\phi)\} \subseteq \text{sol}_V^\Delta(\phi)$ and thus $h \circ \text{sol}_V^S(\phi) \subseteq \text{sol}_V^\Delta(\phi)$.

Lemma 11. *If $(\gamma_1, \gamma_2) \in \text{next}_\phi$ then $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^S(\phi)$ and if $(\gamma_1, \gamma_2) \in \text{cnext}_{\text{ode}(R)}$ then $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^S(\text{ode}(R))$.*

Proof. The definition of next_ϕ shows that any pair $(\gamma_1, \gamma_2) \in \text{next}_\phi$ has the form $(h_S \circ \alpha_1, h_S \circ \alpha_2)$ for some $\alpha_1, \alpha_2 \in \text{sol}^R(\phi)$. John's Soundness Theorem for abstract interpretation of logic formulas (Theorem 10) implies $h_S \circ \text{sol}^R(\phi) \subseteq \text{sol}^S(\phi)$ and thus $\{\gamma_1, \gamma_2\} \subseteq \text{sol}^S(\phi)$. If $(\gamma_1, \gamma_2) \in \text{cnext}_{\text{ode}(R)}$ then $\gamma_2 \in \text{sol}^S(\text{ode}(R))$ by definition. Furthermore there exists γ'_2 such that $(\gamma_1, \gamma'_2) \in \text{next}_{\text{ode}(R)}$. The first property shows that $\gamma_1 \in \text{sol}^S(\text{ode}(R))$ too.

B Proofs for Section 8 (Abstract Simulation of Reaction Networks)

Theorem 12 Soundness. $\text{cnext}_{\text{ode}(R)} \subseteq \text{trans} \circ \text{sol}^S(\text{bnn}(R))$.

Proof. Let $(\gamma_1, \gamma_2) \in \text{cnext}_{\text{ode}(R)}$. Then there exists $(\gamma'_1, \gamma'_2) \in \text{cnext}_{\text{ode}(R)}$ such that $\gamma'_1 = \gamma_1|_S$ and $\gamma'_2 = \gamma_2|_S$. By assumption on $\text{next_spec}(\text{vars}_A)$, this implies for all $A \in \mathcal{S}$ that $\gamma'_1 \cup \vec{\gamma}'_2 \in \text{sol}^S(\bigwedge_{A \in \mathcal{S}} \text{next_spec}(\text{vars}_A))$. Lemma 11 shows that $\gamma'_1, \gamma'_2 \in \text{sol}^S(\text{ode}(R))$ so that $\vec{\gamma}'_2 \in \text{sol}^S(\text{ode}(\vec{R}))$. By definition of $\text{bnn}(R)$ we obtain $\gamma'_1|_S \cup \vec{\gamma}'_2|_S \in \text{sol}^S(\text{bnn}(R))$. Hence, $(\gamma_1, \gamma_2) = (\gamma'_1|_S, \gamma'_2|_S) \in \text{trans} \circ \text{sol}^S(\text{bnn}(R))$ as stated by the theorem. \square