



**HAL**  
open science

## Reaction Networks to Boolean Networks

Joachim Niehren, Athénaïs Vaginay, Cristian Versari

► **To cite this version:**

Joachim Niehren, Athénaïs Vaginay, Cristian Versari. Reaction Networks to Boolean Networks. 2021. hal-02279942v7

**HAL Id: hal-02279942**

**<https://hal.science/hal-02279942v7>**

Preprint submitted on 15 Sep 2021 (v7), 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.

# Reaction Networks to Boolean Networks

Emilie Allart<sup>1,2</sup>, Joachim Niehren<sup>1,3</sup>, and Cristian Versari<sup>1,2</sup>

<sup>1</sup> BioComputing Team, CRISAL Lab, Lille, <sup>2</sup> Université de Lille, <sup>3</sup> Inria Lille

**Abstract.** We study the problem of how to convert reaction networks into boolean networks with nondeterministic updates. We start from the sign abstraction of the ODE semantics of reaction network, and show that it can be captured by a quasi-boolean network, a generalization of boolean networks that we introduce. We then show that any quasi-boolean networks can be converted into a boolean networks by using John’s overapproximation. We then apply a recent rewrite algorithm that make linear equation systems exact for improving the precision of the conversion of quasi-boolean to boolean networks, also in the general nonlinear case. Thereby we improve our compiler from reaction networks to boolean networks as well.

**Keywords:** Systems biology · reaction networks · boolean networks · abstract interpretation.

## 1 Introduction

Reaction networks [9,5,13,7] are the most prominent language for modeling the dynamics of biological system. We consider the deterministic semantics of reaction networks based on ordinary differential equations (ODEs). We propose to study the sign abstraction of dynamical states of the ODEs. The signs tell us whether a reaction rate at some time point is positive, negative or zero, and whether a species concentration is zero or not. The sign abstraction of a state always has a unique successor, yielding a finite sign graph for an ODE system. This sign graph can be used to prove properties of the dynamical behaviour of reaction networks. The question that we study in the present paper is how to compute the sign graph of an ODE system inferred from a reaction network. This is nontrivial even if the sign graph is always finite, since we cannot compute the infinite object from which it abstracts.

In the first step, we show how to overapproximate the sign graph of the ODE system of a reaction network by a boolean network. The idea is to see a system of ODEs as a system of arithmetic equations with the operators  $+$  and  $*$  that can be interpreted over positive reals. These equations can also be interpreted more abstractly over the booleans, mapping  $+$  to disjunction and  $*$  to conjunction. The soundness of this abstract interpretation can be derived from John’s theorem [1,18,14]. It implies that the set of boolean solutions of an arithmetic formula with  $+$  and  $*$  overapproximates the boolean abstraction of the set of its positive real solutions.

It turns out that the sign reinterpretation of the ODEs of the reaction network yields a boolean network, whose transition relation overapproximates the sign graph of the ODEs of the reaction network. While the real-valued states of the ODE have unique successors, this is not the case for their sign abstractions. Further nondeterminism is introduced by the overapproximation of the possible signs by abstract interpretation. Therefore we will have to deal with boolean networks with nondeterministic updates [19] rather than with their more classical deterministic counterparts.

We then study the question of whether or when the sign graph can be computed exactly. For this, we capture the sign graph of ODEs by the transition relation of quasi-boolean networks, a generalization of boolean networks that we introduce. Whether the quasi-boolean network can be

converted to a boolean network depends on whether the boolean abstraction of the  $\mathbb{R}_+$ -solution sets of system of arithmetic equations with operators  $+$  and  $*$  can be described exactly by some boolean expressions.

A positive answer to question in the case of systems of linear homogeneous equations with integer coefficients was obtained recently [2], based on the computation of elementary modes [17,11,12,23]. It yields first-order formulas, for which John’s overapproximation of the boolean abstraction of the  $\mathbb{R}_+$ -solution set is exact. This rewriting algorithm also permits to compile a reaction network with linear ODEs into a boolean network by capturing exactly the sign graph of the ODEs.

We finally apply exact rewriting to general reaction networks with nonlinear kinetic expressions. For this we replace all nonlinear kinetic subexpressions by fresh variables, rewrite exactly the linear equation system obtained thereby, and then replace the fresh variables back by their nonlinear expressions. The boolean networks obtained in this manner may not always capture the sign graph exactly, but may still considerably reduce the overapproximation coming with abstract interpretation over signs.

The exact rewriting for linear equation systems with respect to boolean abstraction is also key for computing the difference abstraction of steady states of reaction networks [1]. We originally invented the exact rewriting in this context, but did not yet publish it. We discover the relevance of this result only recently, triggered by discussion with Jun Pang and Loïc Paulevé following the PhD defense of Emilie Allart.

*Related Work.* Classical boolean networks have deterministic updates specified by boolean functions. Their application for the modeling of biological systems is not new, and their generation from reaction networks has been discussed before. In [6] for instance, boolean networks are manually produced from ODE models based on bifurcation analysis.

More recent notions of boolean networks by Paulevé and Séné [19] permit nondeterministic updates by boolean relations. Nondeterministic updates are for instance useful to capture the most permissive semantics of a classical boolean network [21]. Recently, Paulevé also developed a most permissive semantics for well-structured reaction networks, that captures the sign graph of its ODEs [20].

The Boolean semantics of reaction networks from Fages and Soliman [8] is different in that it abstracts from the multiset rewriting semantics (see e.g. [16]). Multiset rewriting is asynchronous and discrete, in contrast to the ODEs, whose semantics is synchronous and continuous. Therefore, the Boolean semantics of reaction networks is not directly related to the sign abstraction its ODE semantics.

Another compiler from reaction networks to Boolean networks but with deterministic updates was presented recently by Sutavani et al. [22]. This translation is not intended to be correct with respect to the boolean abstraction of the ODE semantics of the reaction network. Therefore, they do not start from the usual ODEs inferred from the reaction network, but from simpler ODEs inferred differently, ignoring the kinetic expressions and the stoichiometry. Similarly to our approach, they then interpret multiplication as conjunction and addition as disjunction. But in contrast to us, they interpret unary subtraction as negation rather than eliminating subtraction beforehand.

## 2 Preliminaries

Let  $\mathbb{B} = \{0, 1\}$  be the set of booleans,  $\mathbb{N}$  the set 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  $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{R}$ .

$\Sigma$ -Algebras. We next recall the usual notions of  $\Sigma$ -algebras and of homomorphism between  $\Sigma$ -algebras. Let  $\Sigma = \cup_{n \geq 0} F^{(n)} \uplus C$  be a ranked signature. The elements of  $f \in F^{(n)}$  are called the  $n$ -ary function symbols of  $\Sigma$  and the elements in  $c \in C$  its constants.

**Definition 1.** A  $\Sigma$ -algebra  $S = (dom(S), \cdot^S)$  consists of a set  $dom(S)$  and an interpretation  $\cdot^S$  such that  $c^S \in dom(S)$  for all  $c \in C$ , and  $f^S : dom(S)^n \rightarrow dom(S)$  for all  $f \in F^{(n)}$ .

Let  $\Sigma_{arith} = \{+, *, 0, 1, -, /\}$  the arithmetic signature, where 0 and 1 are constants, and all other operator binary function symbols. Let  $\Sigma_{bool} = \{+, *, 0, 1\}$  be the subset of boolean operators.

**Example 2.** The set of positive reals  $\mathbb{R}$  can be turned into a  $\Sigma_{arith}$ -algebra, in which the functions symbols are interpreted as addition of positive reals  $+\mathbb{R}$ , and multiplication of positive reals  $*\mathbb{R}$ . The constants are interpreted by themselves  $0\mathbb{R} = 0$  and  $1\mathbb{R} = 1$ .

**Example 3.** The set of positive reals  $\mathbb{R}_+$  can be turned into a  $\Sigma_{bool}$ -algebra, in which the functions symbols are interpreted as addition of positive reals  $+\mathbb{R}_+$ , and multiplication of positive reals  $*\mathbb{R}_+$ . The constants are interpreted by themselves  $0\mathbb{R}_+ = 0$  and  $1\mathbb{R}_+ = 1$ .

**Example 4.** The set of Booleans  $\mathbb{B} = \{0, 1\} \subseteq \mathbb{R}_+$  equally defines a  $\Sigma_{bool}$ -algebra. There, the function symbols are interpreted as a disjunction  $+\mathbb{B} = \vee^{\mathbb{B}}$  and conjunction  $*\mathbb{B} = \wedge^{\mathbb{B}}$  on Booleans. The constants are interpreted by themselves  $0\mathbb{B} = 0$  and  $1\mathbb{B} = 1$ .

**Example 5.** The set of function of type  $\mathbb{R}_+ \rightarrow \mathbb{R}$  can be turned into a  $\Sigma_{arith}$ -algebra. Function addition and multiplication are defined pointwise, that is  $(f +^{\mathbb{R}_+ \rightarrow \mathbb{R}} f')(r) = f(r) +^{\mathbb{R}} f'(r)$  and  $(f *^{\mathbb{R}_+ \rightarrow \mathbb{R}} f')(r) = f(r) *^{\mathbb{R}} f'(r)$ . The interpretation  $0^{\mathbb{R}_+ \rightarrow \mathbb{R}}$  is the constant-0 $\mathbb{R}$  function, and similarly  $1^{\mathbb{R}_+ \rightarrow \mathbb{R}}$  is the constant-1 $\mathbb{R}$  function.

$\Sigma$ -Structures. In order to generalize  $\Sigma$ -algebras to  $\Sigma$ -structures, we consider  $n$ -ary function symbols as  $n+1$ -ary relation symbols.

**Definition 6.** A  $\Sigma$ -structure  $\Delta = (dom(\Delta), \cdot^\Delta)$  consists of a set  $dom(\Delta)$  and an interpretation  $\cdot^\Delta$  such that  $c^\Delta \in dom(\Delta)$  for all  $c \in C$  and  $f^\Delta \subseteq dom(\Delta)^{n+1}$  for all  $f \in F^{(n)}$ .

Clearly, any  $\Sigma$ -algebra is also a  $\Sigma$ -structure. Note also that symbols in  $F^{(0)}$  are interpreted as monadic relations, i.e., as subsets of the domain, in contrast to constants in  $C$  that are interpreted as elements of the domain.

**Example 7.** The set of signs  $\{-1, 0, 1\} \subseteq \mathbb{R}$  can be turned into a  $\Sigma_{bool}$ -structure  $\mathbb{S} = (\{-1, 0, 1\}, \cdot^{\mathbb{S}})$  with the interpretation  $+\mathbb{S}$  and  $*\mathbb{S}$  given in Fig. 3. The constants are interpreted by themselves  $0^{\mathbb{S}} = 0$  and  $1^{\mathbb{S}} = 1$ . Note that all  $+\mathbb{S}$  contains  $(-1, 1, -1)$ ,  $(-1, 1, 1)$  and  $(-1, 1, 0)$  meaning that the sum of a strictly negative and a strictly positive real has a sign in  $-1 +^{\mathbb{S}} 1$ , so it may either be strictly positive, strictly negative, or zero. For this reason,  $\mathbb{S}$  is a  $\Sigma$ -structure but not a  $\Sigma$ -algebra.

**Definition 8.** A homomorphism between two  $\Sigma$ -structures  $S$  and  $\Delta$  is a function  $h : dom(S) \rightarrow dom(\Delta)$  such that for  $c \in C$ ,  $n \in \mathbb{N}$ ,  $f \in F^{(n)}$ , and  $s_1, \dots, s_{n+1} \in dom(S)$ :

1.  $h(c^S) = c^\Delta$ , and
2. if  $(s_1, \dots, s_{n+1}) \in f^S$  then  $(h(s_1), \dots, h(s_{n+1})) \in f^\Delta$ .

We can convert any  $n + 1$ -ary relation to a  $n$ -ary set valued functions. In this way any  $n$ -function is converted to a  $n$ -ary set valued  $n$ -functions. In other words, functions of type  $D^n \rightarrow D$  are converted to functions of type  $D^n \rightarrow 2^D$  where  $D = \text{dom}(\Delta)$ . In set-valued notation, the second condition on homomorphism can then be rewritten equivalently as  $h(f^S(s_1, \dots, s_n)) \subseteq f^\Delta(h(s_1), \dots, h(s_n))$ . A homomorphism for  $\Sigma$ -algebras thus satisfies  $h(c^S) = c^\Delta$  and for all function symbols  $f \in F^{(n)}$  and  $s_1, \dots, s_n \in \text{dom}(S)$  it satisfies  $h(f^S(s_1, \dots, s_n)) = f^\Delta(h(s_1), \dots, h(s_n))$ .

*Expressions, Equations, ODEs.* Let  $\mathcal{V}$  be a set of variables and  $\Sigma = C \cup \bigcup_{n \geq 0} F^{(n)}$  a ranked signature with constants and function symbols. The set of  $\Sigma$ -expressions  $e \in \mathcal{E}_\Sigma(\mathcal{V})$  is then given by the following abstract syntax:

$$e_1, \dots, e_n \in \mathcal{E}_\Sigma ::= x \mid c \mid f(e_1, \dots, e_n) \text{ where } c \in C, n \geq 0, f \in F^{(n)}, x \in \mathcal{V}$$

The semantics of an expression  $e \in \mathcal{E}_\Sigma$  can be defined as usual for any  $\Sigma$ -structure  $S$  and variable assignment  $\alpha : V \rightarrow \text{dom}(S)$  with  $\mathcal{V}(e) \subseteq V \subseteq \mathcal{V}$ : it is the subset of the domain  $\llbracket e \rrbracket^{\alpha, S} \subseteq \text{dom}(S)$ . Note that the semantics of expressions is set-valued in order to deal with  $\Sigma$ -structures. For  $\Sigma$ -algebras, it will always be a singleton.

A  $\Sigma$ -equation with variables in  $\mathcal{V}$  is a pair  $e \stackrel{\circ}{=} e'$  where  $e, e' \in \mathcal{E}_\Sigma(\mathcal{V})$ . A system of  $\Sigma$ -equations is a conjunction of equations  $\bigwedge_{i=1}^n e_i \stackrel{\circ}{=} e'_i$ . In the case  $n = 0$  the conjunction is  $\text{true} \stackrel{\text{def}}{=} 1 \stackrel{\circ}{=} 1$ . An equation system  $\phi$  equal to  $\bigwedge_{i=1}^n e_i \stackrel{\circ}{=} e'_i$  is true for a  $\Sigma$ -structure  $S$  and a variable assignment  $\alpha : V \rightarrow \text{dom}(S)$  with  $\mathcal{V}(\phi) \subseteq V$  if  $\llbracket e_i \rrbracket^{\alpha, S} \cap \llbracket e'_i \rrbracket^{\alpha, S} \neq \emptyset$  for all  $1 \leq i \leq n$ . The equality symbol  $\stackrel{\circ}{=}$  is interpreted as nondisjointness, i.e.,  $e \stackrel{\circ}{=} e'$  is true if and only if  $\llbracket e \rrbracket^{\alpha, S} \cap \llbracket e' \rrbracket^{\alpha, S} \neq \emptyset$ . In the case of  $\Sigma$ -algebras, the equality symbol  $\stackrel{\circ}{=}$  is indeed interpreted as equality of singleton. In the case of more general  $\Sigma$ -structures, it will not be interpreted as set equality so.

Let  $\mathcal{S}$  be a finite set of species and  $\dot{\mathcal{S}} = \{S, \dot{S} \mid S \in \mathcal{S}\}$ . An ODE system is a conjunction of  $\Sigma_{arith}$ -equations with variables in  $\dot{\mathcal{S}}$ . We assume for all subexpressions  $e/e'$  that, the value of  $e'$  over  $\mathbb{R}$  is always different from 0. A solution of an ODE system is a function  $\gamma : \dot{\mathcal{S}} \rightarrow \mathbb{R}_+ \rightarrow \mathbb{R}$  that makes all equations true over the structure  $\mathbb{R}_+ \rightarrow \mathbb{R}$  such that  $\frac{d\gamma(S)}{dt}(t) = \gamma(\dot{S})(t)$  for all  $S \in \mathcal{S}$  and  $t \in \mathbb{R}_+$ .

**Lemma 9 Positive Rewriting.** *Any ODE system can be rewritten into an ODE system with boolean operators in  $\Sigma_{bool}$  while preserving the solutions in the  $\Sigma_{bool}$ -algebra  $\mathbb{R}$ .*

**Proof** Subtractions in  $\Sigma_{arith}$ -formulas  $e - e' \stackrel{\circ}{=} e''$  can be eliminated by rewriting to  $e \stackrel{\circ}{=} e'' + e'$ . Division in  $\Sigma_{arith}$ -formula  $e/e' \stackrel{\circ}{=} e''$  can be removed by rewriting to  $e \stackrel{\circ}{=} e'' * e'$ . The latter works since we assumed that the denominator  $e'$  can never become equal to 0. By exhaustive rewriting, we obtain an ODE system with only boolean operators in  $\Sigma_{bool}$ .

*Reaction Networks.* A reaction network with species in  $\mathcal{S}$  is a set of chemical reactions of the following form:  $r : R \xrightarrow{e} P$  where  $r$  is a name,  $R, P : \mathcal{S} \rightarrow \mathbb{N}$  are multisets of species called respectively the reactants and the products, and  $e \in \mathcal{E}_{\Sigma_{arith}}(\mathcal{S})$  is the kinetic expression. The kinetic expression specifies the rate of the reaction at any time point, that is specified by the ODE semantics.

The ODEs of a reaction network  $N$  are inferred as follows. If a species  $S \in \mathcal{S}$  is produced or consumed by reactions  $r_i$  with stoichiometry  $z_i \in \mathbb{Z}$  – the number of  $S$  produced minus number of  $S$  consumed by  $r_i$  – and kinetic expression  $e_i$  then the ODE  $\dot{S} = z_1 e_1 + \dots + z_n e_n$  is added to the ODEs of  $N$ . Consider for instance the acyclic reaction network  $N_{acyc}$  in Fig. 1. It has the molecular species is  $\mathcal{S} = \{A, B, C, D\}$  and two reactions with mass-action kinetics  $r_1 : A \xrightarrow{A} B$  and  $r_2 : A + C \xrightarrow{A * C} D$ . The deterministic semantics of  $N_{acyc}$  is the system of ODEs in Fig. 2. One numerical simulation of the ODEs, corresponding to one solution  $\gamma$ , is shown in Fig. 4.

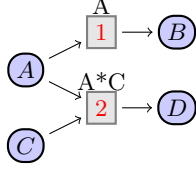


Fig. 1: The acyclic reaction network  $N_{acyc}$ .

$$\begin{aligned} \dot{A} &\stackrel{\circ}{=} -A - A * C & \wedge & \quad \dot{B} \stackrel{\circ}{=} A & \wedge \\ \dot{C} &\stackrel{\circ}{=} -A * C & \wedge & \quad \dot{D} \stackrel{\circ}{=} A * C \end{aligned}$$

Fig. 2: Deterministic semantics by ODEs.

$d$	$d'$	$d +^{\mathbb{S}} d'$	$d *^{\mathbb{S}} d'$	$d$	$d'$	$d +^{\mathbb{S}} d'$	$d *^{\mathbb{S}} d'$	$d$	$d'$	$d +^{\mathbb{S}} d'$	$d *^{\mathbb{S}} d'$
-1	1	{-1, 0, 1}	-1	0	1	1	0	1	1	1	1
-1	0	-1	0	0	0	0	0	1	0	1	0
-1	-1	-1	1	0	-1	-1	0	1	-1	{-1, 0, 1}	-1

Fig. 3: Interpretation of  $\Sigma$ -structure of signs  $\mathbb{S}$ .

*$\Sigma$ -Abstractions.* The boolean abstraction is the function  $h_{\mathbb{B}} : \mathbb{R}_+ \rightarrow \mathbb{B}$  with  $h_{\mathbb{B}}(0) = 0$  and  $h_{\mathbb{B}}(r) = 1$  if  $r > 0$ . The sign abstraction is the function  $h_{\mathbb{S}} : \mathbb{R} \rightarrow \mathbb{S}$  with  $h_{\mathbb{S}}(0) = 0$ ,  $h_{\mathbb{S}}(r) = -1$  for all strictly negative reals  $r < 0$  and  $h_{\mathbb{S}}(r) = 1$  for all strictly positive reals  $r > 0$ . We can generalize the boolean abstraction and the sign abstraction as follows:

**Definition 10.** A  $\Sigma$ -abstraction is a homomorphism  $h: S \rightarrow \Delta$  between  $\Sigma$ -structures such that  $\text{dom}(\Delta) \subseteq \text{dom}(S)$ .

**Lemma 11.** The boolean abstraction  $h_{\mathbb{B}}$  is a  $\Sigma_{bool}$ -abstraction into a  $\Sigma_{bool}$ -algebra.

**Lemma 12.** The sign abstraction is  $h_{\mathbb{S}}$  a  $\Sigma_{bool}$ -abstraction into a  $\Sigma_{bool}$ -structure.

*First-Order Logic.* We call the syntax and semantics of first-order logic. The set of first-order formulas  $\mathcal{F}_{\Sigma}(\mathcal{V})$  is constructed from  $\Sigma$ -equations with variables in  $\mathcal{V}$  and the usual first-order connectives:

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

The set of free variables  $\mathcal{V}(\phi)$  are all those variables of  $\phi$  that occur outside the scope of any quantifier. The semantics of first-order logic can be defined as usual for any  $\Sigma$ -structure  $S$  and variable assignment  $\alpha : V \rightarrow \text{dom}(S)$ . It yields a truth value  $\llbracket \phi \rrbracket^{\alpha, S} \in \mathbb{B}$  for any formula  $\phi \in \mathcal{F}_{\Sigma}(\mathcal{V})$  with  $\mathcal{V}(\phi) \subseteq V$ . See Fig. 11 of the appendix for the details. The set of solutions of a formula  $\phi \in \mathcal{F}_{\Sigma}(\mathcal{V})$  over a  $\Sigma$ -algebra  $S$  with respect to a subset of variables  $V \supseteq \mathcal{V}(\phi)$  is:  $\text{sol}_V^S(\phi) = \{\alpha : V \rightarrow \text{dom}(S) \mid \llbracket \phi \rrbracket^{\alpha, S} = 1\}$ . If  $V = \mathcal{V}(\phi)$  we omit the index  $V$ , i.e.,  $\text{sol}^S(\phi) = \text{sol}_V^S(\phi)$ . We can define the boolean abstraction by a first-order formula with signature  $\Sigma_{bool}$ :

$$y \stackrel{\circ}{=} h_{\mathbb{B}}(x) =_{\text{def}} (y = 0 \wedge x \stackrel{\circ}{=} 0) \vee (y \stackrel{\circ}{=} 1 \wedge \neg x \stackrel{\circ}{=} 0)$$

We need also to define applications of the boolean abstraction to the solution set of a first-order formula  $\phi$  with signature  $\Sigma_{bool}$ . For this, let  $\mathbf{y}$  be vector of distinct variables such that  $\{\mathbf{y}\} = \mathcal{V}(\phi)$ . Let  $\mathbf{x}$  be a vector of fresh variables with the same arity than  $\mathbf{y}$ . We then define the application of  $h_{\mathbb{B}}$  to the solution set of  $\phi$  by the first-order formula  $h_{\mathbb{B}}(\phi)$  as follows:

$$h_{\mathbb{B}}(\phi) =_{\text{def}} \exists \mathbf{x}. \phi[\mathbf{y}/\mathbf{x}] \wedge \bigwedge_i \mathbf{y}_i \stackrel{\circ}{=} h_{\mathbb{B}}(\mathbf{x}_i)$$

*Abstract Interpretation.* We recall John’s theorem [14] on how to overapproximate abstractions of solution sets of first-order formulas by abstract interpretation. Let  $h : S \rightarrow \Delta$  be a  $\Sigma$ -abstraction and  $V \subseteq \mathcal{V}$ . For any subset of assignments  $R$  of type  $V \rightarrow \text{dom}(S)$  we define  $h \circ R = \{h \circ \alpha : V \rightarrow \text{dom}(\Delta) \mid \alpha \in R\}$ .

**Theorem 13 John’s Overapproximation [1,18,14].** *For any  $\Sigma$ -abstraction  $h : S \rightarrow \Delta$  between  $\Sigma$ -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  $\mathcal{V}(e) \subseteq V$  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  $\mathcal{V}(\phi) \subseteq V$  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  $h \circ \text{sol}_V^S(\phi) \subseteq \text{sol}_V^\Delta(\phi)$ .

### 3 Sign Graphs of ODEs

Let  $\mathcal{S}$  be a set of species and  $\dot{\mathcal{S}} = \{S, \dot{S} \mid S \in \mathcal{S}\}$ . Note that the system of ODEs  $E$  of any reaction network with species in  $\mathcal{S}$  is a conjunction of  $\Sigma_{arith}$ -equations with variables in  $\dot{\mathcal{S}}$ , and thus a formula of  $\mathcal{F}_{\Sigma_{arith}}(\dot{\mathcal{S}})$ .

A state of a system of ODE equations is a function  $\alpha : \dot{\mathcal{S}} \rightarrow \mathbb{R}$  that solves all its equations over the algebra  $\mathbb{R}$ . Note that  $\alpha(\dot{S})$  may be negative, while  $\alpha(S) \geq 0$  for all  $S \in \mathcal{S}$  by assumption. Let  $E'$  be the formula  $E \wedge \bigwedge_{S \in \mathcal{S}} S \geq 0$ , where the formula  $S \geq 0$  is defined by  $\exists y. S = y * y$ .

A real assignment  $\alpha : \dot{\mathcal{S}} \rightarrow \mathbb{R}$  can be abstracted to a sign assignment  $h_{\mathbb{S}} \circ \alpha : \dot{\mathcal{S}} \rightarrow \mathbb{S}$ . Given an ODE system  $E$  we define a possibly nondeterministic successor relation on sign assignments  $\beta, \beta' \in h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(E')$ : we call  $\beta'$  a successor of  $\beta$  if

$$\forall S \in \mathcal{S}. \beta'(S) = \beta(S) +^{\mathbb{S}} \max(0, \beta(\dot{S}))$$

The sign graph of  $E$  has as node the elements of  $h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(E')$ , and as edges the elements of the successor relation. Due to the nondeterminism, some sign assignments may have multiple outgoing edges in the sign graph, as for instance in the sign graph in Fig. 8.

**Example 14.** *Reconsider the ODEs in Fig. 2 inferred from the reaction network  $N_{acyc}$  in Fig. 1. Let  $\gamma$  be the ODE solution in Fig. 4. We consider the following  $\mathbb{R}$ -solutions of the ODE equations:*

$$\begin{aligned} \alpha_1 = \gamma(0) &= [A/2, B/0, C/1, D/0, \dot{A}/-4, \dot{B}/2, \dot{C}/-2, \dot{D}/2] \\ \alpha_2 = \gamma(0.1) &= [A/0.7, B/0.1, C/0.3, D/0.9, \dot{A}/-0.2, \dot{B}/0.1, \dot{C}/-0.1, \dot{D}/0.1] \end{aligned}$$

*The sign abstractions of these solutions are:*

$$\begin{aligned} \beta_1 = h_{\mathbb{S}} \circ \alpha_1 &= [A/1, B/0, C/1, D/0, \dot{A}/-1, \dot{B}/1, \dot{C}/-1, \dot{D}/1] \\ \beta_2 = h_{\mathbb{S}} \circ \alpha_2 &= [A/1, B/1, C/1, D/1, \dot{A}/-1, \dot{B}/1, \dot{C}/-1, \dot{D}/1] \end{aligned}$$

*We can now see that  $\beta_2$  is a successor of  $\beta_1$ . For species  $B$ , for instance, we have  $\beta_2(B) = 1 = 0 + 1 = \beta_1(B) + \max(0, \beta_1(\dot{B}))$ . The whole sign graph for the ODEs of  $N_{acyc}$  is given in Fig. 5.*

### 4 Approximating Sign Graphs by Boolean Networks

We first give a syntax for the boolean networks with nondeterministic updates from [19] based on boolean formulas. We then present a compiler mapping the ODEs of a reaction network to a Boolean network based on boolean interpretation of the ODEs.

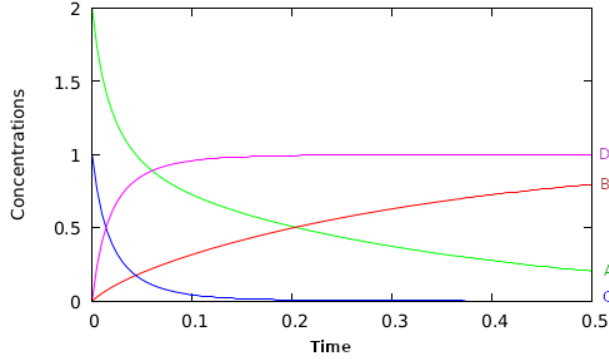


Fig. 4: Solution of ODEs of  $N_{acyc}$  with initial state  $A(0) = 2$ ,  $C(0) = 1$  and  $B(0) = D(0) = 0$ .

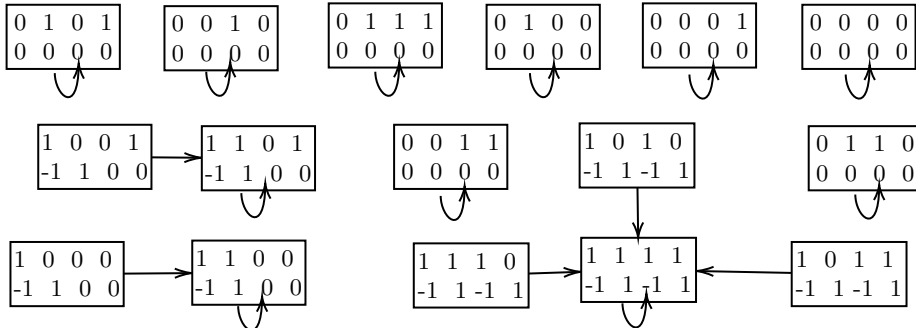


Fig. 5: Sign graph of ODEs for reaction network  $N_{acyc}$ .

The compiler will be done in such a way that the transition relation of the boolean network will overapproximate the sign graph of the ODEs. The overapproximation may introduce further nondeterminism to that of the sign graph, as we will see in the example in Fig. 10 of Section 5. In order to capture the nondeterminism, we need to admit nondeterministic updates for boolean networks too.

Let  $\mathcal{S}$  be a finite set of species. The set  $\mathcal{S} \times \{1, 2\}$  provides two copies of  $\mathcal{S}$ . For any  $S \in \mathcal{S}$ , the first copy  $S_1 = (S, 1)$  will denote the old boolean value of  $S$  before a transition, and  $S_2 = (S, 2)$  the new boolean value of  $S$  after the transition. A boolean network with nondeterministic updates and species in  $\mathcal{S}$  is a formula  $\phi \in \mathcal{F}_{\{+,*,0,1\}}(\mathcal{S} \times \{1, 2\})$ . For illustration, the boolean network  $\phi_{acyc}^{bool}$  is given in Fig. 6. With  $\mathcal{S} = \{A, B, C, D\}$ , the network has the set of species:

$$\dot{\mathcal{S}}^{+/-} = \{S, \dot{S}^+, \dot{S}^- \mid S \in \mathcal{S}\}$$

For each boolean network  $\phi$  we define a transition relation  $\rightarrow_{\phi}^{\mathbb{B}}$  on boolean assignments of species. Given two boolean assignments  $\beta_1, \beta_2 : \mathcal{S} \rightarrow \mathbb{B}$ , let  $\beta : \mathcal{S} \times \{1, 2\}$  be such that  $\beta(S_i) = \beta_i(S)$  for all  $S \in \mathcal{S}$  and  $i \in \{1, 2\}$ . We can then define the transition relation  $\rightarrow_{\phi}^{\mathbb{B}}$  such that:

$$\beta_1 \rightarrow_{\phi}^{\mathbb{B}} \beta_2 \text{ if and only if } \llbracket \phi \rrbracket^{\beta, \mathbb{B}} = 1$$

In this case, we say that  $\beta_2$  is a successor of  $\beta_1$  with respect to the boolean network  $\phi$ . For the boolean network  $\phi_{acyc}^{bool}$  in Fig. 6, the transition relation  $\rightarrow_{\phi_{acyc}^{bool}}^{\mathbb{B}}$  corresponds exactly to the sign graph  $N_{acyc}$  when identifying the sign of  $\dot{S}$  by the equation  $\dot{S} = \dot{S}^+ - \dot{S}^-$ .



$$\begin{aligned}
& \bigwedge_{S \in \{A, B, C, D\}} S_2 \stackrel{\circ}{=} S_1 + \dot{S}_1^+ \wedge \dot{S}_2^- * \dot{S}_2^+ \stackrel{\circ}{=} 0 \\
& \wedge \dot{A}_2^+ + A_2 + A_2 * C_2 \stackrel{\circ}{=} \dot{A}_2^- \quad \wedge \dot{B}_2^+ \stackrel{\circ}{=} \dot{B}_2^- + A_2 \\
& \wedge \dot{C}_2^+ + A_2 * C_2 \stackrel{\circ}{=} \dot{C}_2^- \quad \wedge \dot{D}_2^+ \stackrel{\circ}{=} \dot{D}_2^- + A_2 * C_2
\end{aligned}$$

Fig. 6: Boolean network  $\phi_{acycl}^{bool}$  obtained by boolean interpretation of ODEs of  $N_{acyc}$ .

We next present a compiler from reaction networks to boolean networks with nondeterministic updates. The objective is to overapproximate the sign graph of the ODEs of the reaction network by the transition relation of the boolean network. For instance, the reaction network  $N_{acyc}$  will be mapped to the boolean network  $bn(N_{acyc}) = \phi_{acyc}^{bool}$ .

Given a reaction network  $N$ , the compiler starts with the ODEs of  $N$ . As argued earlier, these can be rewritten positively to ODEs with operators in  $\Sigma_{bool}$  only. While the variables  $S \in \mathcal{S}$  must have positive values at every time point, the variables  $\dot{S}$  may still have negative values. In order to get rid of all negative values, the idea is that every real number  $r \in \mathbb{R}$  is equal to a difference of two positive real numbers  $r^+ - r^-$ , where  $r^+ = \max(r, 0)$  and  $r^- = -\min(r, 0)$ . Hence,  $r^+, r^- \in \mathbb{R}_+$  and the decomposition  $r = r^+ - r^-$  is unique such that  $r^+ \geq 0 \wedge r^- \geq 0 \wedge r^+ * r^- = 0$ . Therefore, we replace each variable  $\dot{S}$  by a difference of two fresh variables  $\dot{S}^+ - \dot{S}^-$  that are constrained by  $\dot{S}^+ \geq 0 \wedge \dot{S}^- \geq 0 \wedge \dot{S}^+ * \dot{S}^- = 0$ . We then apply Lemma 9 to rewrite the ODEs positively to a boolean formula with signature  $\Sigma_{bool}$  with variables in  $\dot{\mathcal{S}}^{+/-}$ , while eliminating the subtraction operator. For the example  $N_{acyc}$  we obtain the boolean formula  $ode_{bf}(N_{acyc})$ :

$$\dot{A}^+ + A + A * C = \dot{A}^- \wedge \dot{B}^+ = \dot{B}^- + A \wedge \dot{C}^+ + A * C = \dot{C}^- \wedge \dot{D}^+ = \dot{D}^- + A * C$$

The boolean formula  $ode_{bf}(N)$  can be used to compile the reaction network  $N$  to a boolean network  $bn(N)$ . Suppose that the reaction network  $N$  has the species in  $\mathcal{S}$ . Then  $ode_{bf}(N) \in \mathcal{F}_{\Sigma_{bool}}(\dot{\mathcal{S}}^{+/-})$ . We define the boolean network  $bn(N)$  with species in  $\dot{\mathcal{S}}^{+/-}$  as follows:

$$bn(N) = ode_{bf}(N)[S/S_2 \mid S \in \mathcal{S}] \wedge \bigwedge_{S \in \mathcal{S}} S_2 \stackrel{\circ}{=} S_1 + \dot{S}_1^+ \wedge \dot{S}_2^- * \dot{S}_2^+ \stackrel{\circ}{=} 0$$

For any species  $S \in \mathcal{S}$  the new value described by  $S_2$  is determined by  $S_2 \stackrel{\circ}{=} S_1 + \dot{S}_1^+$  when given the old values. Furthermore, the new values must satisfy the ODEs of  $N$ , that is the expression  $ode_{be}(N)$  in which all  $S$  are replaced by  $S_2$ . For illustration, the boolean network  $bn(N_{acyc})$  is given in Fig. 6. Nicely, it captures precisely the sign graph of  $N_{acyc}$ .

## 5 Overapproximation Example

For general reaction networks  $N$  the boolean network  $bn(N)$  may only overapproximate the sign graph of  $N$ . The overapproximation may be large as we illustrate next by example. For this we consider the cyclic reaction network  $N_{cyc}$  in Fig. 7. The sign graph of  $N_{cyc}$  is given in Fig. 8. The boolean network  $bn(N_{cyc})$  inferred by boolean interpretation of  $ode_{be}(N_{cyc})$  is given in Fig. 9. The approximation of the sign graph made by the boolean network  $bn(N_{cyc})$  is shown in Fig. 10. Note that contains more sign assignments than the sign graph of  $N_{cyc}$ . The question is under how to capture the sign graphs of reaction networks more precisely or even exactly by boolean networks.

## 6 Capturing Sign Graphs by Quasi-Boolean Networks

We propose to generalize boolean networks to quasi-boolean networks. We then show how to compile reaction networks to quasi-boolean networks while capturing the sign graph exactly.

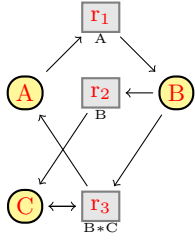


Fig. 7: The reaction network  $N_{cyc}$ .

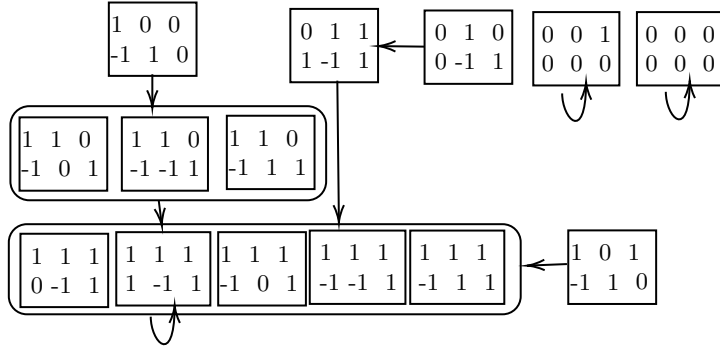


Fig. 8: The sign graph of  $N_{cyc}$  ( $ABC$  over  $\dot{A}\dot{B}\dot{C}$ ).

$$\begin{aligned} \dot{A}_2^+ + A_2 &\stackrel{\circ}{=} \dot{A}_2^- + B_2 * C_2 \wedge \dot{B}_2^+ + B_2 + B_2 * C_2 \stackrel{\circ}{=} \dot{B}_2^- + A_2 \wedge \dot{C}_2^+ \stackrel{\circ}{=} \dot{C}_2^- + B_2 \\ \wedge \bigwedge_{S \in \{A, B, C\}} S_2 &\stackrel{\circ}{=} S_1 + \dot{S}_1^+ \wedge \dot{S}_2^- * \dot{S}_2^+ \stackrel{\circ}{=} 0 \end{aligned}$$

Fig. 9: The boolean network  $bn(N_{cyc})$ .

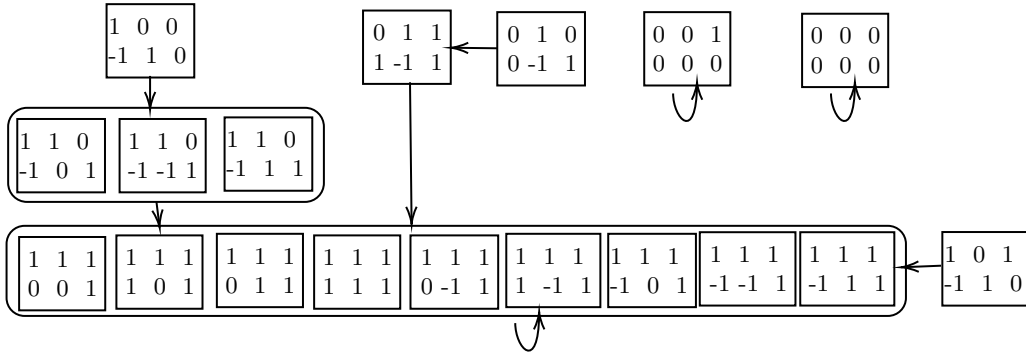


Fig. 10: Overapproximation of the sign graph by  $bn(N_{cyc})$ .

A quasi-boolean network with species in  $\mathcal{S}$  has the same syntax than a boolean network. But for the semantics, the formulas  $\phi \in \mathcal{F}_{\Sigma_{bool}}(\mathcal{S} \times \{1, 2\})$  are interpreted over  $\mathbb{R}_+$  rather than  $\mathbb{B}$ . The transition relation  $\rightarrow_{\phi}^{\mathbb{R}_+}$  is defined such that:

$$\beta_1 \rightarrow_{\phi}^{\mathbb{R}_+} \beta_2 \text{ if and only if } \llbracket \phi \rrbracket^{\beta, \mathbb{R}_+} = 1$$

In this case we say that  $\beta_2$  is a successor of  $\beta_1$  with respect to the quasi-boolean network  $\phi$  on  $\mathbb{R}_+$ .

Given a reaction network  $N$  with species in  $\mathcal{S}$ , let  $ode_{bf}(N)$  be the boolean formula that we obtained from its ODEs. This formula is a system of  $\Sigma_{bool}$ -equation with variables in  $\dot{\mathcal{S}}^{+/-}$ . We define the quasi-boolean network  $qbn(N)$  with species in  $\dot{\mathcal{S}}^{+/-}$  as follows:

$$qbn(N) = h_{\mathbb{B}}(qb(N))$$

Recall from Section 2 that  $h_{\mathbb{B}}(\text{ode}_{bf}(N))$  is the first-order formula that defines the  $h_{\mathbb{B}}$ -abstraction of the solutions of  $\text{ode}_{bf}(N)$ . In  $h_{\mathbb{B}}(\text{ode}_{bf}(N))$ , each variable  $S$  is to be replaced by  $S_2$ , since the ODE equations are to be applied to the values after the transition step.

**Proposition 15 Correctness of  $qbn(N)$ .** *For any reaction network  $N$ , the sign graph of  $N$  is equal to the sign graph encoded by the transition relation of the quasi-boolean network  $qbn(N)$ .*

**Proof** This follows directly from the constructions.

**Proposition 16  $bn(N)$  overapproximates  $qbn(N)$ .** *The transition relation of the quasi-boolean network  $qbn(N)$  is contained in the transition relation of the boolean network  $bn(N)$ .*

**Proof** By John's overapproximation (Theorem 13), we have  $h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}^+}(\phi) \subseteq \text{sol}^{\mathbb{B}}(\phi)$  and thus  $\text{sol}^{\mathbb{R}^+}(h_{\mathbb{B}}(\phi)) \subseteq \text{sol}^{\mathbb{B}}(\phi)$ .

**Theorem 17  $bn(N)$  overapproximates the sign graph of  $N$ .** *The sign graph encoded by the transition relation of the boolean network  $bn(N)$  subsumes the sign graph of  $N$ .*

**Proof** This is immediate from the correctness of the transition graph of  $qbn(N)$  with respect to the sign graph of  $N$  in Proposition 15 and the overapproximation for the transition relation of  $bn(N)$  relatively to  $qbn(N)$  by Proposition 16.

Our next objective is to rewrite quasi-boolean networks to boolean networks while preserving the transition relation. For this, we propose to rewrite formulas  $\phi \in \mathcal{F}_{\Sigma_{bool}}(\mathcal{V})$  to  $\mathbb{R}_+$ -equivalent formulas  $\phi' \in \mathcal{F}_{\Sigma_{bool}}(\mathcal{V})$  such that  $\text{sol}^{\mathbb{B}}(\phi') = h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}^+}(\phi') = h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}^+}(\phi)$ . The formula  $\phi'$  can then be rewritten into a boolean expression  $e \in \mathcal{E}_{\Sigma_{bool}}(\mathcal{V})$  that holds for the same boolean assignments, as we showed earlier. We provide a solution to the above problem only for the special case, where the formulas  $\phi$  represent a linear homogenous equation system with integer coefficients. But the linear case can be used to obtain better Boolean networks for reaction networks with nonlinear ODEs, simply by replacing nonlinear subexpressions by variables, and putting nonlinear equations for these variables aside. This is the way we computed the sign graph for  $N_{acyc}$  in Fig. 8, even though its ODEs are nonlinear.

## 7 Linear Equation Systems and Elementary Modes

We are interested in systems of  $\Sigma_{bool}$ -equations. The base case is homogeneous linear equations systems with natural coefficients, which capture linear matrix integer equations  $A\mathbf{x} = 0$  interpreted over  $\mathbb{R}_+$ . We will show that elementary modes [17,11,12,23] can be used to transform linear integer matrix equations into equivalent systems which are *quasi-positive* and *strongly-triangular*, two key properties that we will show guarantee the exactness of their interpretation over the booleans. In order to extend our result to linear equation systems interpreted over  $\mathbb{R}$ , we also need systems of polynomial equations, with natural coefficients and no constant term, that are nonlinear.

In the following, let  $e^n = \prod_{i=1}^n e$  and  $ne =_{\text{def}} \sum_{i=1}^n e$ .

**Definition 18.** *A  $\Sigma_{bool}$ -equation is called positive if it has the form  $e \stackrel{\circ}{=} 0$  and quasi-positive if it has the form  $e \stackrel{\circ}{=} ny$ , where  $n \in \mathbb{N}$ ,  $y \in \mathcal{V}$ , and  $e \in \mathcal{E}_{\Sigma_{bool}}(\mathcal{V})$ .*

This definition makes sense, since all constants in  $\Sigma_{bool}$ -expressions are positive and all operators of  $\Sigma_{bool}$ -expressions preserve positivity. Note also that any positive equation is quasi-positive since the constant 0 is equal to the polynomial  $0y$ . A *system of  $\Sigma_{bool}$ -equations* is a conjunctive

formula of the form  $\bigwedge_{i=1}^n e_i \overset{\circ}{=} e'_i$  in  $\mathcal{F}_\Sigma$ . We call a system of  $\Sigma_{bool}$ -equations *positive* respectively *quasi-positive* if all its equations are.

A *polynomial (with natural coefficients)* is an expression of the form  $\sum_{j=1}^l n_j \prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}}$  where  $l$  and  $i_j$  are naturals,  $x_{1,1}, \dots, x_{l,i_l}$  variables, all  $n_j \neq 0$  naturals called the *coefficients*, and all  $m_{j,k} \neq 0$  naturals called the *exponents*. The products  $\prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}}$  are called the *monomials* of the polynomial.

**Definition 19.** A polynomial  $\sum_{j=1}^l n_j \prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}}$  with natural coefficients  $n_j \neq 0$  has no constant term if none of its monomials is equal to 1, i.e.,  $i_j \neq 0$  for all  $1 \leq j \leq l$ . It is linear if all its monomials are variables, i.e.  $i_j = 1$  and  $m_{j,1} = \dots = m_{j,i_j} = 1$  for all  $1 \leq j \leq l$ .

Note that any linear polynomial has the form  $\sum_{j=1}^l n_j x_{j,1}$  where  $l$  and all  $n_j \neq 0$  are naturals and all  $x_{j,1}$  are variables. In particular, linear polynomials do not have a constant term. Furthermore, note that the constant 0 is equal to the linear polynomial with  $l = 0$ . A *polynomial equation* is a  $\Sigma_{bool}$ -equation  $p \overset{\circ}{=} p'$  between polynomials. A (*homogeneous*) *linear equation* is a polynomial equation with linear polynomials, so without constant terms. A *linear equation system* is a system of linear equations.

A (*homogeneous*) *linear integer matrix equation* has the form  $A\mathbf{y} \overset{\circ}{=} 0$  where  $A$  is a  $n \times m$  matrix of integers for some naturals  $m, n$  such that  $\mathbf{y} \in \mathcal{V}^m$ . Any linear integer matrix equation can be turned into a linear equation system with natural coefficients, by bringing the negative coefficients on the right-hand side. For instance, the linear integer matrix equation on the right corresponds to the linear equation system with natural coefficients  $3x \overset{\circ}{=} 0 \wedge 2x \overset{\circ}{=} 5y$ .

$$\begin{pmatrix} 3 & 0 \\ 2 & -5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \overset{\circ}{=} 0$$

This system is quasi-positive, but not positive since  $5y$  appears on a right-hand side. More generally, the linear equation system for an linear integer matrix equation  $A\mathbf{y} \overset{\circ}{=} 0$  is positive if and only if all integers in  $A$  are positive, and quasi-positive, if each line of  $A$  contains at most one negative integer. Furthermore, the above linear equation system is triangular in the following sense, but not strongly triangular:

**Definition 20.** We call a quasi-positive system of  $\Sigma_{bool}$ -equations triangular if it has the form  $\bigwedge_{l=1}^n e_l \overset{\circ}{=} n_l y_l$  such that the variables  $y_l$  are  $l$ -fresh for all  $1 \leq l \leq n$ , i.e.,  $y_l \notin \mathcal{V}(\bigwedge_{i=1}^{l-1} e_i \overset{\circ}{=} e'_i)$  and if  $n_l \neq 0$  then  $y_l \notin \mathcal{V}(e_l)$ . We call the quasi-positive polynomial system strongly-triangular if it is triangular and satisfies  $n_l \neq 0$  for all  $1 \leq l \leq n$ .

Consider a linear integer matrix equation  $A\mathbf{y} \overset{\circ}{=} 0$ . If  $A$  is positive and triangular, then the corresponding linear equation system is positive and triangular too. For being quasi-positive and strongly-triangular, the integers below the diagonal of  $A$  must be negative, those on the diagonal must be strictly negative, and those on the right of the diagonal must be positive.

**Theorem 21 Elementary Modes [2].** For any system of linear equations  $\phi$  with natural coefficients, one can compute in at most exponential time a  $\mathbb{R}_+$ -equivalent formula  $\exists \mathbf{x}.\phi'$  such that  $\phi'$  is a quasi-positive strongly-triangular system of linear equations with natural coefficients and  $\mathbf{x}$  the sequence of variables on the left hand sides of the equations.

The theorem applies in particular to the linear equation systems of integer matrix equations  $A\mathbf{y} \overset{\circ}{=} 0$ . It shows that there exists a matrix  $E$  of naturals, a vector of naturals  $\mathbf{n}$ , and a vector of fresh variables  $\mathbf{x}$ , such that  $A\mathbf{y} \overset{\circ}{=} 0$  is  $\mathbb{R}_+$ -equivalent to  $\exists \mathbf{x}.\mathbf{E}\mathbf{x} \overset{\circ}{=} \mathbf{n}\mathbf{y}$ .

## 8 Boolean Abstractions of Mixed Systems

In this section we prove our main result stating how to compute the  $h_{\mathbb{B}}$ -abstraction of the  $\mathbb{R}_+$ -solution set of a mixed system exactly.

**Definition 22.** *A mixed system is a formula in  $\mathcal{F}_{\Sigma}$  of the form  $\exists \mathbf{z}. \phi \wedge \phi'$  where  $\phi$  is a linear equation system and  $\phi'$  a  $h_{\mathbb{B}}$ -invariant and  $h_{\mathbb{B}}$ -exact first-order formula.*

Note that linear equation systems  $A\mathbf{y} \stackrel{\circ}{=} 0$ , with  $A$  an integer matrix and  $\mathbf{y}$  a sequence of pairwise distinct variables, need not to be  $h_{\mathbb{B}}$ -exact, if  $A$  is not positive. However, any linear equation systems of this shape is  $\mathbb{R}_+$ -equivalent to some quasi-positive strongly-triangular polynomial system, as shown by the Elementary Modes Theorem 21. And quasi-positive strongly-triangular polynomial system were shown to be  $h_{\mathbb{B}}$ -exact by Theorem 4  $h_{\mathbb{B}}$ -exactness of [2].

**Theorem 23 Main.** *Any mixed system can be converted in exponential time to an  $\mathbb{R}_+$ -equivalent  $\Sigma$ -formula that is  $h_{\mathbb{B}}$ -exact.*

**Proof** The proof comes directly from the Theorem 6 of [2].

**Corollary 24.** [2] *The  $h_{\mathbb{B}}$ -abstraction of the  $\mathbb{R}_+$ -solution set of a mixed system  $\phi$ , that is  $h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}_+}(\phi)$ , can be computed in at most exponential time in the size of the system  $\phi$ .*

The algorithm from the proof Corollary 24 can be improved so that it becomes sufficiently efficient for practical use. For this the two steps with exponential worst case complexity must be made polynomial for the particular instances. First note that the computation of the elementary modes (Theorem 21) is known to be computationally feasible. Various algorithms for this purpose were implemented [10,15,3,4] and applied successfully to problems in systems biology [12]. The second exponential step concerns the enumeration of all boolean variable assignments. This enumeration may be avoided by using constraint programming techniques for computing the solution set  $\text{sol}^{\mathbb{B}}(\phi')$ . For those mixed systems for which both steps can be done in polynomial time, we can compute the boolean abstraction of the  $\mathbb{R}_+$ -solution set in polynomial time too. The practical feasibility of this approach was demonstrated recently at an application to knockout prediction in systems biology [1], where previously only over-approximations could be computed.

## 9 Improving Boolean Networks for Sign Graphs

Let us now show how to obtain better boolean network for the sign graphs of reaction networks than by direct interpretation of ODEs over the booleans. We illustrate the approach for the example reaction network  $N_{cyc}$ . The formula  $bn(N_{cyc})$  can be turned into a mixed-system by replacing the nonlinear kinetic expression  $B_2 * C_2$  by some fresh variable  $v$ . The resulting mixed-system can then be rewritten by Theorem 23 to a  $h_{\mathbb{B}}$ -exact formula that is  $\mathbb{R}_+$ -equivalent. In the exact formula, we replace  $v$  back by  $B_2 * C_2$  yielding the boolean network below. The network indeed captures the sign graph of  $N_{cyc}$  in Fig. 8 exactly. The existentially bound variables  $x_1, \dots, x_n$  for boolean values are introduced by the elementary modes.

$$\begin{aligned} A_2 &\stackrel{\circ}{=} A_1 + \dot{A}_1^+ \wedge B_2 \stackrel{\circ}{=} B_1 + \dot{B}_1^+ \wedge C_2 \stackrel{\circ}{=} C_1 + \dot{C}_1^+ \\ \wedge \dot{A}_2^+ * \dot{A}_2^- &\stackrel{\circ}{=} 0 \wedge \dot{B}_2^+ * \dot{B}_2^- \stackrel{\circ}{=} 0 \wedge \dot{C}_2^+ * \dot{C}_2^- \stackrel{\circ}{=} 0 \\ \wedge \exists x^0 \dots x^7. A_2 &\stackrel{\circ}{=} x^5 + x^6 + x^7 \wedge \dot{A}_2^- \stackrel{\circ}{=} x^4 + x^6 + x^7 \wedge \dot{A}_2^+ \stackrel{\circ}{=} x^3 + x^4 \wedge B_2 \stackrel{\circ}{=} x^2 + x^7 \\ \wedge \dot{B}_2^- &\stackrel{\circ}{=} x^1 + x^2 + x^3 \wedge \dot{B}_2^+ \stackrel{\circ}{=} x^1 + x^6 \wedge \dot{C}^- \stackrel{\circ}{=} x^0 \wedge \dot{C}_2^+ \stackrel{\circ}{=} x^0 + x^2 + x^7 \wedge B_2 * C_2 \stackrel{\circ}{=} x^3 + x^5 \end{aligned}$$

*Acknowledgements.* We thank Jun Pang and L oc Paulev e for the helpful discussions and references on the relation of reaction and boolean networks, as well as our colleagues from BioComputing Maxime Folschette and C edric 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. 2021.
3. D. Avis and C. Jordan. mplrs: A scalable parallel vertex/facet enumeration code. *Mathematical Programming Computation*, 10(2):267–302, 2018.
4. R. Bagnara, P. M. Hill, and E. Zaffanella. The parma polyhedra library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Sci. Comput. Program.*, 72(1-2):3–21, 2008.
5. 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.
6. M. Davidich and S. Bornholdt. The transition from differential equations to boolean networks: A case study in simplifying a regulatory network model. *Journal of Theoretical Biology*, 255(3):269–277, 2008.
7. F. Fages, S. Gay, and S. Soliman. Inferring reaction systems from ordinary differential equations. *Theor. Comput. Sci.*, 599:64–78, 2015.
8. F. Fages and S. Soliman. Abstract interpretation and types for systems biology. *Theor. Comput. Sci.*, 403(1):52–70, 2008.
9. 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.
10. K. Fukuda. cdd: c: C-implementation of the double description method for computing all vertices and extremal rays of a convex polyhedron given by a system of linear inequalities. *Department of Mathematics, Swiss Federal Institute of Technology, Lausanne, Switzerland*, 1993.
11. K. Fukuda and A. Prodon. Double description method revisited. In M. Deza, R. Euler, and I. Manoussakis, editors, *Combinatorics and Computer Science*, pages 91–111, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
12. J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC bioinformatics*, 5(1):1, 2004.
13. 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.
14. 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.
15. S. Klamt, J. Stelling, M. Ginkel, and E. D. Gilles. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics*, 19(2):261–269, 01 2003.
16. 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.
17. T. Motzkin, H. Raiffa, G. Thompson, and R. Thrall. The double description method. In H. Kuhn and A.W.Tucker, editors, *Contributions to theory of games*, volume 2. Princeton University Press, 1953.
18. 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.
19. 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.
20. L. Paulevé. Most permissive reaction networks, 2021.
21. L. Paulevé, J. Kolça, T. Chatain, and S. Haar. Reconciling Qualitative, Abstract, and Scalable Modeling of Biological Networks. *Nature Communications*, 11, 2020.

22. S. Sutavani, K. Sarda, A. Yerudkar, and N. Singh. Interpretation of complex reaction networks in boolean network framework. In *2018 Indian Control Conference (ICC)*, pages 7–11, 2018.
23. D. Zanghellini, D. E. Ruckerbauer, M. Hanscho, and C. Jungreuthmayer. Elementary flux modes in a nutshell: Properties, calculation and applications. *Biotechnology Journal*, pages 1009–1016, 2013.

Interpretation of expressions as sets of elements  $\llbracket e \rrbracket^{\alpha, S} \subseteq \text{dom}(S)$ , where  $S$  is a  $\Sigma$ -structure and  $\alpha : V \rightarrow \text{dom}(S)$  where  $V$  contains all free variables.

$$\llbracket c \rrbracket^{\alpha, S} = c^S \quad \llbracket x \rrbracket^{\alpha, S} = \{\alpha(x)\} \quad \llbracket e \odot e' \rrbracket^{\alpha, S} = \cup\{s \odot^S s' \mid s \in \llbracket e \rrbracket^{\alpha, S}, s' \in \llbracket e' \rrbracket^{\alpha, S}\}$$

Interpretation of formulas as truth values  $\llbracket \phi \rrbracket^{\alpha, S} \in \mathbb{B}$ :

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

Fig. 11: Semantics of  $\Sigma$ -expressions and formulas over a  $\Sigma$ -structure  $S$  with respect to a variable assignment  $\alpha : V \rightarrow \text{dom}(S)$ .

## A Proofs for Section 2 (Preliminaries)

For any set  $A$  and  $n \in \mathbb{N}$ , the set of  $n$ -tuples of elements in  $A$  is denoted by  $A^n$ . The  $i$ -th projection function on  $n$ -tuples of elements in  $A$ , where  $1 \leq i \leq n$  is the function  $\pi_i : A^n \rightarrow A$  such that  $\pi_i(a_1, \dots, a_n) = a_i$  for all  $a_1, \dots, a_n \in A$ . If  $A$  is finite the number of elements of  $A$  is denote by  $|A|$ . *Projections and Pairs*. The projection  $\pi_a(f)$  of a function  $f : A \rightarrow B$  is its restriction  $\alpha|_{A \setminus \{a\}}$ . The projection of a set  $F$  of functions  $f : A \rightarrow B$  is  $\pi_a(F) = \{\pi_a(f) \mid f \in F\}$ . Furthermore, we define the pair function  $f^2 : A^2 \rightarrow B^2$  such that  $f^2(a_1, a_2) = (f(a_1), f(a_2))$ .

**Lemma 11.** *The boolean abstraction  $h_{\mathbb{B}}$  is a  $\Sigma_{\text{bool}}$ -abstraction into a  $\Sigma_{\text{bool}}$ -algebra.*

*Proof.* For all  $r, r' \in \mathbb{R}_+$  we have:

$$\begin{aligned} h_{\mathbb{B}}(r +^{\mathbb{R}_+} r') &= 1 \Leftrightarrow r +^{\mathbb{R}_+} r' \neq 0 \Leftrightarrow r \neq 0 \vee r' \neq 0 \Leftrightarrow h_{\mathbb{B}}(r) = 1 \vee h_{\mathbb{B}}(r') = 1 \\ h_{\mathbb{B}}(r *^{\mathbb{R}_+} r') &= 1 \Leftrightarrow r *^{\mathbb{R}_+} r' \neq 0 \Leftrightarrow r \neq 0 \wedge r' \neq 0 \Leftrightarrow h_{\mathbb{B}}(r) = 1 \wedge h_{\mathbb{B}}(r') = 1 \end{aligned}$$

Hence  $h_{\mathbb{B}}(r +^{\mathbb{R}_+} r') = h_{\mathbb{B}}(r) +^{\mathbb{B}} h_{\mathbb{B}}(r')$  and  $h_{\mathbb{B}}(r *^{\mathbb{R}_+} r') = h_{\mathbb{B}}(r) *^{\mathbb{B}} h_{\mathbb{B}}(r')$ . Finally, for both constants  $c \in C$  we have that  $h_{\mathbb{B}}(c^{\mathbb{R}_+}) = h_{\mathbb{B}}(c) = c = c^{\mathbb{B}}$ .

**Lemma 12.** *The sign abstraction is  $h_{\mathbb{S}}$  a  $\Sigma_{\text{bool}}$ -abstraction into a  $\Sigma_{\text{bool}}$ -structure.*

*Proof.* For any  $r, r' \in \mathbb{R}$  the second condition for homomorphism follows for all  $\odot \in F^{(2)}$ :  $h_{\mathbb{S}}(r \odot r') \in h_{\mathbb{S}}(r) \odot^{\mathbb{S}} h_{\mathbb{S}}(r')$ . And for all constants  $c \in C$  we have by definition that  $h_{\mathbb{S}}(c^{\mathbb{R}}) = c^{\mathbb{S}}$ .

The following two lemmas are classical. Let  $R$  be a subset of assignments of type  $V' \rightarrow \text{dom}(S)$  and  $V \cap V' = \emptyset$  two subsets of  $\mathcal{V}$ . We define:  $\text{ext}_V^S(R) = \{\alpha \cup \alpha' \mid \alpha' : V' \cup V \rightarrow \text{dom}(S), \alpha \in R\}$ .

**Lemma 25 Cylindrification.** *If  $V \cap \mathcal{V}(\phi) = \emptyset$  then:  $\text{sol}_{V \cup \mathcal{V}(\phi)}^S(\phi) = \text{ext}_V^S(\text{sol}^S(\phi))$ .*

*Proof.* We can show for all expressions  $e \in \mathcal{E}_{\Sigma}$  with  $\mathcal{V}(e)$  disjoint to  $V$  and any variables assignment  $\alpha : \mathcal{V}(e) \cup V \rightarrow \text{dom}(S)$  that  $\llbracket e \rrbracket^{\alpha, S} = \llbracket e \rrbracket^{\alpha|_{\mathcal{V}(e)}, S}$ . This is by induction on the structure of expressions. It follows for all formulas  $\phi \in \mathcal{F}_{\Sigma}(\mathcal{V})$  such that  $\mathcal{V}(\phi)$  disjoint from  $V$  and  $\alpha : \mathcal{V}(\phi) \cup V \rightarrow \text{dom}(S)$  that  $\llbracket \phi \rrbracket^{\alpha, S} = \llbracket \phi \rrbracket^{\alpha|_{\mathcal{V}(\phi)}, S}$ . This is by induction on the structure of formulas. It implies the lemma.

**Lemma 26 Quantification is projection.**  $\text{sol}^S(\exists x. \phi) = \pi_x(\text{sol}^S(\phi))$ .

*Proof.* This is straightforward from the semantics of existential quantifiers:  $\text{sol}^S(\exists x. \phi) = \{\alpha|_{\mathcal{V}(\phi) \setminus \{x\}} \mid \alpha \in \text{sol}^S(\phi)\} = \pi_x(\text{sol}^S(\phi))$ .



## B Proofs for Section 7 (Linear Equation Systems and Elementary Modes)

**Theorem 21 Elementary Modes [2].** *For any system of linear equations  $\phi$  with natural coefficients, one can compute in at most exponential time a  $\mathbb{R}_+$ -equivalent formula  $\exists \mathbf{x}.\phi'$  such that  $\phi'$  is a quasi-positive strongly-triangular system of linear equations with natural coefficients and  $\mathbf{x}$  the sequence of variables on the left hand sides of the equations.*

*Proof.* Consider a system  $\phi$  of linear equations with natural coefficients. Geometrically, the solution space of  $\phi$  over the reals is a linear subspace of  $\mathbb{R}^{\mathcal{V}(\phi)}$ . When restricted to positive reals, as we do, this linear subspace is to be intersected with the positive cone  $\mathbb{R}_+^{\mathcal{V}(\phi)}$ . Therefore,  $\text{sol}^{\mathbb{R}_+}(\phi)$  is a rational cone obtained by intersecting finitely many half-spaces: its H-representation is defined by the inequations  $\mathbf{A}\mathbf{y} \leq 0 \wedge \mathbf{A}\mathbf{y} \geq 0 \wedge \mathbf{y} \geq 0$ . The elementary modes of  $\phi$  are the extreme rays of this cone and allow its V-representation. Up to normalization there are finitely many such extreme rays. Moreover, since the cone is rational and  $\phi$  is homogeneous, the elementary modes can be normalised so that the V-representation contains only integer coefficients. The normalized elementary modes will be vectors of naturals in  $\mathbb{N}^{\mathcal{V}(\phi)}$ . Let  $e_1, \dots, e_n$  be the set of all normalized elementary modes in some arbitrary total order. Let  $E$  be the matrix with columns  $e_1, \dots, e_n$ . Let  $\mathbf{y}$  be the sequence of all variables of  $\mathcal{V}(\phi)$  in some arbitrary total order. By construction, the variable in  $\mathbf{y}$  are pairwise distinct. According to the normalized V-representation of the system, every point of the cone is a positive linear combination of the elementary modes  $\exists \mathbf{x}.\mathbf{E}\mathbf{x} \stackrel{\circ}{=} \mathbf{n}\mathbf{y}$ , where  $\mathbf{n}$  contains the normalization factors. The linear system  $\mathbf{E}\mathbf{x} \stackrel{\circ}{=} \mathbf{n}\mathbf{y}$  is quasi-positive, since  $E$  and  $\mathbf{n}$  are positive. The variables in  $\mathbf{x}$  can be chosen freshly, and thus pairwise distinct with  $\mathbf{y}$ . The linear system  $\mathbf{E}\mathbf{x} \stackrel{\circ}{=} \mathbf{n}\mathbf{y}$  is strongly-triangular, since each variable of  $\mathbf{y}$  occurs in exactly one equation and never on the left. Therefore, we can define  $\phi'$  as  $\mathbf{E}\mathbf{x} \stackrel{\circ}{=} \mathbf{n}\mathbf{y}$ . The computation of the elementary modes and thus of  $E$  can be done in at most exponential time in the size of  $\phi$  by Motzkin's double description method [17,11,12].

## C Proofs for Section 8 (Boolean Abstractions of Mixed Systems)

**Corollary 24.** [2] *The  $h_{\mathbb{B}}$ -abstraction of the  $\mathbb{R}_+$ -solution set of a mixed system  $\phi$ , that is  $h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}_+}(\phi)$ , can be computed in at most exponential time in the size of the system  $\phi$ .*

*Proof.* Given an mixed system  $\phi$ , we can apply Theorem 23 to compute in at most exponential time an  $\mathbb{R}_+$ -equivalent formula  $\phi''$  that is  $h_{\mathbb{B}}$ -exact. It is then sufficient to compute  $\text{sol}^{\mathbb{B}}(\phi'')$  in exponential time in the size of  $\phi$ . This can be done in the naive manner, that is by evaluating the formula  $\phi''$  – which may be of exponential size – over all possible boolean variable assignments – of which there may be exponentially many. For each assignment the evaluation can be done in PSPACE and thus in exponential time. The overall time required is thus a product of two exponentials, which remains exponential.

## D Proofs for Section 9 (References)