Knockout Prediction for Reaction Networks with Partial Kinetic Information

Mathias John, Mirabelle Nebut, Joachim Niehren

Knockout Prediction for Reaction Networks with Partial Kinetic Information

Mathias John[12], Mirabelle Nebut[12], and Joachim Niehren[13]

[1] BioComputing, LIFL (CNRS UMR8022)
[2] University of Lille
[3] INRIA Lille

Abstract. In synthetic biology, a common application field for computational methods is the prediction of knockout strategies for reaction networks. Thereby, the major challenge is the lack of information on reaction kinetics. In this paper, we propose an approach, based on abstract interpretation, to predict candidates for reaction knockouts, relying only on partial kinetic information. We consider the usual deterministic steady state semantics of reaction networks and a few general properties of reaction kinetics. We introduce a novel abstract domain over pairs of real domain values to compute the differences between steady states that are reached before and after applying some knockout. We show that this abstract domain allows us to predict correct knockout strategy candidates independent of any particular choice of reaction kinetics. Our predictions remain candidates, since our abstract interpretation over-approximates the solution space. We provide an operational semantics for our abstraction in terms of constraint satisfaction problems and illustrate our approach on a realistic network.

Keywords: Abstract interpretation, deterministic semantics, steady state, constraint satisfaction, synthetic biology.

1 Introduction

Synthetic biology aims at creating artificial micro-organisms, either by constructing them from scratch or by modifying existing once [2, 1, 21]. To this end, computational modeling is applied to predict the dynamic behavior of the resulting organisms [24, 32]. Thereby, it is a common task to abstract micro-organisms as sets of chemical reactions and to predict the effects of reaction deletion on the dynamics of the concentrations of chemical species [38, 5]. Such knockout considerations are in particular applied to predict strategies that increase the rate of some target reactions of interest.

Kinetic information is essential to predict the dynamics of chemical systems [22]. In reaction networks, each reaction is endowed with a kinetic rate law, that is a function that defines a rate in dependence of the concentration of the reaction's reactants. Common examples of rate laws are the law of mass action and Michaelis-Menten kinetics. These depend on kinetic rate constants that,

however, remain mostly unknown. Moreover, in practice, many of a model's reactions combine several unknown reaction steps to one, see e.g. [19, 20]. In these cases, only basic properties of the kinetics function are given, e.g. that the rate increases or decreases with the concentration of some chemical species. Thus, the precise prediction of the dynamics of chemical systems remains out of reach.

The commonly accepted dynamic semantics of systems of chemical reactions describes species concentrations in terms of stochastic processes [18]. When dealing with high concentrations, a meaningful approximation is to ignore the moments of probability distributions with an order greater than one [40, 17]. Such, so called, deterministic semantics is then represented by systems of ordinary differential equations (ODEs) that describe the changes in the mean of species concentrations over time. Knockout predictions then usually regard fix points of ODEs when no more changes in concentrations occur (steady state). It is known that in nature systems of chemical reactions exist that have none or more than one steady state [37, 11, 14].

Abstract interpretation [8] has been introduced for the static analysis of program semantics. The idea is to approximate the state space of programs based on approximations of domains and computations. In the realm of chemical reaction systems, abstract interpretation has been applied to obtain different approximations of their dynamic, stochastic semantics [12, 10] or to decrease the size of (infinitely) large sets of ODEs representing their deterministic semantics [6, 13].

In this paper we propose an approach based on abstract interpretation, that predicts candidates for reaction knockout, with only partial kinetic information. We focus on cases that are on one hand simple, since reaction systems are assumed to always reach a steady state as it is usually done when predicting reaction knockouts [39, 5]. On the other hand, they are complicated, since kinetics functions are completely unknown, except for a few basic properties. Our major idea is to compare steady states of reaction systems before and after a reaction knockout is applied. We use abstract interpretation to reason about the effects of knockouts in the absence of kinetic information. Therefore, we introduce a small abstract domain that works on relations of pairs of non negative real numbers and abstract operators that represent an over-approximation of real domain addition and multiplication. We then propose a small set of properties for kinetics functions that are general enough to be also fulfilled by mass action kinetics and Michaelis-Menten kinetics. As our central contribution, we show that the predictions obtained with our abstract interpretation are independent of any particular choice of kinetics functions, as long as these fulfill these properties. Our predictions remain candidates due to the over-approximating nature of our approach. We also present an operational semantics for our approach based on a mapping of the abstract reaction semantics to constraint satisfaction problems [33].

We start off by introducing the deterministic and steady state semantics of reaction systems (Section 2). Then we formalize the reaction knockout task in terms of a real domain constraint satisfaction problem (Section 3). To do so, in

the first instance, we restrict ourselves to reactions with mass action kinetics. Then we define our abstract domain and the abstract interpretation of our mass action knockout constraints yielding sets of finite domain constraints (Section 4). We show that our abstract interpretation is a correct over-approximation in that every solution of the real domain problem is reflected by the solutions of the abstract domain problem. We proceed by introducing properties of kinetics functions and show that these are fulfilled by mass action kinetics and Michaelis-Menten kinetics (Section 5). Then, we prove that every kinetics function with these properties yields the same results in our abstraction, which allows us to generally stick to mass action in our abstract interpretation. We next show how to solve the thus obtained table constraints (Section 6). As this may yield large sets of solutions, we propose to integrate Branch-And-Bound optimization w.r.t. different criteria. We then provide in Section 7 a real-world example based on a model from literature and our own implementation and discuss the obtained solutions. These, on one hand, confirm results already known in literature and on the other hand represent new interesting strategies that are currently evaluated in wet-lab experiments. To finish, we provide a discussion of existing approaches for prediction reaction knockouts in absence of kinetic information (Section 8). An extended version of this paper with appendices containing detailed information on proofs and on the example model is provided online.[4]

2   Reaction Networks

Let M be a finite set of chemical species (molecule types). A chemical solution S with molecules in M is a tuple $(S_m)_{m \in M}$ with $S_m \in \mathbb{R}_{\geq 0}$. We call S_m the concentration of m in S, that is the number of molecules of species m in S divided by the volume of S.

A reaction network over M is a finite set R of chemical reactions with species in M. Thereby, a chemical reaction is a rewrite rule that states how chemical solutions change over time. More formally, a chemical reaction r with species in M is a tuple of the the following form:

$$(r) \; m_1, \ldots, m_j : m_{j+1}, \ldots, m_k \xrightarrow{\kappa} m_{k+1}, \ldots, m_l$$

where $m_1, \ldots, m_l \in M$ and $\kappa : \mathbb{R}_{\geq 0}^k \to \mathbb{R}_{\geq 0}$ is a function. Species $m_{j+1}, \ldots, m_k$ are called the reactants of r, i.e. molecules of this type are consumed when r is applied. Species on the right hand side are called the products of r, which are produced at application time. Species $m_1, \ldots, m_j$ are called enzymes of r. They are neither consumed nor produced but may increase the rate of reaction r, which is defined by the kinetics function κ. We assume that species can only play a single role in every reaction, i.e. they are either enzyme, reactant, or product. However, in every role they might occur more than once. We write $enz_r(m)$, $react_r(m)$, and $prod_r(m)$ for the number of occurrences of m as an enzyme, reactant, or product of r.

---

[4] http://www.lifl.fr/BioComputing/extendedPapers/vmcai13.pdf

$$\text{(SPEC)} \quad \frac{m \in M}{\frac{dS_m}{dt}(t) = \sum_{r \in R}(prod_r(m) - react_r(m)) \cdot rate_{S,r}(t)}$$

$$\text{(RATE)} \quad \frac{r \in R \quad enz(r) = (m_1, \ldots, m_j) \quad react(r) = (m_{j+1}, \ldots, m_k)}{rate_{S,r}(t) = \kappa_r(S_{m_1}(t), \ldots, S_{m_k}(t))}$$

Fig. 1. Deterministic semantics of reaction network R over M.

A kinetics function of arity k, $\kappa : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$, defines the rate (propensity) at which k-tuples of molecules of S may react, in function of the concentrations of the reactant and enzyme types in the solution. If $(m_1, \ldots, m_k)$ are the enzyme and reactant types of a chemical reaction with kinetics function κ then the reaction rate for a chemical solution S is equal to $\kappa(S_{m_1}, \ldots, S_{m_k})$. We write κ_r for the kinetics function of $r \in R$.

Standard chemical reactions have two reactants, no enzymes, and follow mass action kinetics [18]. The kinetics of such standard reactions is then simply defined as the product of the concentration of its reactants times some rate constant (see the Appendix of the extended version of this paper for more details). However, as a first approximation this reaction model can be extended to arbitrary many enzymes and reactants. We denote this generalized version of the mass action kinetics function as $ma_c : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ for some constant $c \in \mathbb{R}_{\geq 0}$, such that $ma_c(x_1, \ldots, x_k) = c \cdot x_1 \cdot \ldots \cdot x_k$.

Other kinds of kinetics functions give better models of enzymatic reactions. These are justified by compositions of several standard reactions with mass-action kinetics. The most frequent example is Michaelis-Menten kinetics, which accounts for a single-reactant reaction that is triggered by a single enzyme. It is given by the kinetics function $mm_{c_1,c_2} : \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$, such that $mm_{c_1,c_2}(a, e) = c_1 \cdot a \cdot e / (c_2 + a)$, where a, e are the concentrations of the reactant and enzyme, respectively. Rates following Michaelis-Menten kinetics describe a saturation curve that steadily increases with the concentration of the reactant but approaches a limit depending on the enzyme concentration. Yet another interesting alternative are Hill kinetics.

Deterministic Semantics. The deterministic semantics of a reaction network R is a collection of functions $(S_m)_{m \in M}$ of type $S_m : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The value of $S_m(t)$ defines the concentration of m in the solution at time point t, so that the solution at time point t is $(S_m(t))_{m \in M}$. If the initial solution at time point 0 is fixed then the solutions at all later time points $t > 0$ are determined by a collection of ODEs (see below). However, we might not know the initial solution in practice.

The deterministic semantics of a reaction network R over M is defined by applying rule (SPEC) to each species in M. The resulting ODEs compute the change of S_m at time point t by substituting reactants by products for all chem-

$$(\text{SPEC}_{st})\ \frac{m \in M}{\sum_{r \in R} prod_r(m) \cdot rate_{S,r} = \sum_{r' \in R} react_{r'}(m) \cdot rate_{S,r'}}$$

$$(\text{RATE}_{st})\ \frac{r \in R \quad enz(r) = (m_1, \ldots, m_j) \quad react(r) = (m_{j+1}, \ldots, m_k)}{rate_{S,r} = \kappa_r(S_{m_1}, \ldots, S_{m_k})}$$

Fig. 2. Steady state semantics of reaction network R over M.

ical reactions $r \in R$ according to their rate, see Fig. 1. Rule (RATE) defines the rate of a reaction r at time point t by $rate_{S,r}(t) = \kappa_r(S_{m_1}(t), \ldots, S_{m_k}(t))$.
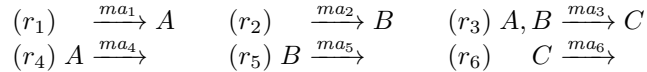
Steady-State Semantics. The steady state semantics assumes that reaction networks reach a fixed point in which all reactions continue to perform with constant speed. This means that the changes for all $m \in M$ become zero:

$$\frac{dS_m}{dt}(t) = 0$$

The amounts $S_m(t)$ will thus become constant for all species $m \in M$, so that we can denote them by S_m. Furthermore, reactions $r \in R$ become constant fluxes with constant rates, so that $rate_{S,r}(t)$ becomes constant and can thus be denoted by $rate_{S,r}$.

The steady state semantics of a reaction network R is given by the system of arithmetic equations in Fig. 2. These equations relate molecule concentrations S_m to reaction rates $rate_{S,r}$. Rule (SPEC$_{st}$) states that the production and consumption must coincide for any molecule type. It should be noticed that these equations are only a necessary condition for steady states of the dynamic system (and that some systems may not have any steady state). However, as mentioned earlier, we assume that reaction networks always reach a steady state.

Example. We consider a reaction network where A's and B's are inputs from the environment that can react to a complex C which is then released into the environment. We assume that all inputs and outputs are done such that an equilibrium must be reached.

$$(r_1)\ \xrightarrow{ma_1} A \qquad (r_2)\ \xrightarrow{ma_2} B \qquad (r_3)\ A, B \xrightarrow{ma_3} C$$
$$(r_4)\ A \xrightarrow{ma_4} \qquad (r_5)\ B \xrightarrow{ma_5} \qquad (r_6)\quad C \xrightarrow{ma_6}$$

Note that we chose artificial rate constants i for reaction r_i. In practice the situation will be even worse in that most rate constants will be unknown. The deterministic semantics is given by the following system of ODEs:

$$\frac{dS_A}{dt}(t) = 1 - 3 \cdot S_A(t) \cdot S_B(t) - 4 \cdot S_A(t),$$
$$\frac{dS_B}{dt}(t) = 2 - 3 \cdot S_A(t) \cdot S_B(t) - 5 \cdot S_B(t),$$
$$\frac{dS_C}{dt}(t) = 3 \cdot S_A(t) \cdot S_B(t) - 6 \cdot S_C(t).$$
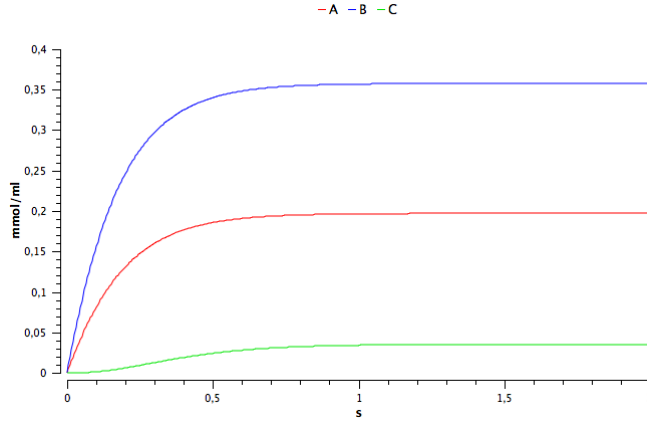
Fig. 3. Dynamics of example reaction network.

In order to determine the functions S_A, S_B, and S_C, it is sufficient to fix the initial solution. We will choose $S_A(0) = S_B(0) = S_C(0) = 0$ for illustration, which leads to the dynamics drawn in Fig. 3. It should be noticed that the concentrations of all molecules stabilize when time t tends to infinity, i.e., a steady state solution is reached. The ODEs induce the following arithmetic equations for this steady state:

$$1 = 3 \cdot S_A \cdot S_B + 4 \cdot S_A, \qquad 2 = 3 \cdot S_A \cdot S_B + 5 \cdot S_B, \qquad 3 \cdot S_A \cdot S_B = 6 \cdot S_C.$$

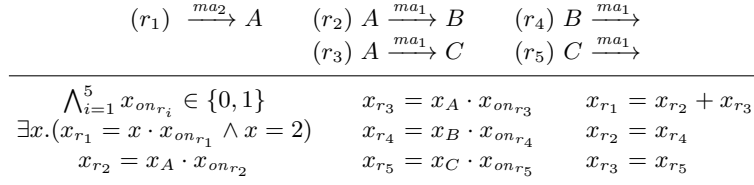One can now solve these quadratic equations to determine two solutions for S_A:

$$(1)\ S_A = \frac{-23 + \sqrt{769}}{24} \qquad (2)\ S_A = \frac{-23 - \sqrt{769}}{24}$$

As only the second solution is positive, we obtain a single steady state at $S_A = 0.197119$, $S_B = 0.357695$, and $S_C = 0.0352542$.

3 Reaction Knockouts

We are now interested in modifying reaction networks, such that the rate of some reactions are increased or decreased in the steady state. The only modifications that we permit are reaction knockouts, i.e. inactivation of some reactions. As mentioned in introduction, we assume in this section that all reactions have mass action kinetics. In Section 5, we extend our approach to a more general, only partially known kinetics, relying on our abstract interpretation given in Section 4.

The knockout problem that we want to study is the following: we are given a reaction network R and some objective O. An objective compares the rate of reactions in steady states that R reaches before and after applying some reaction

6

$$
\begin{array}{lll}
(r_1) \xrightarrow{ma_2} A & (r_2)\ A \xrightarrow{ma_1} B & (r_4)\ B \xrightarrow{ma_1} \\
 & (r_3)\ A \xrightarrow{ma_1} C & (r_5)\ C \xrightarrow{ma_1}
\end{array}
$$

$$
\begin{array}{lll}
\bigwedge_{i=1}^{5} x_{on_{r_i}} \in \{0,1\} & x_{r_3} = x_A \cdot x_{on_{r_3}} & x_{r_1} = x_{r_2} + x_{r_3} \\
\exists x.(x_{r_1} = x \cdot x_{on_{r_1}} \wedge x = 2) & x_{r_4} = x_B \cdot x_{on_{r_4}} & x_{r_2} = x_{r_4} \\
x_{r_2} = x_A \cdot x_{on_{r_2}} & x_{r_5} = x_C \cdot x_{on_{r_5}} & x_{r_3} = x_{r_5}
\end{array}
$$

Fig. 4. Example of a simple knockout problem: the knockout problem on top with its reaction network and objective O, its knockout constraint below.

$$
(\text{SPEC}_{ko})\ \frac{m \in M}{\sum_{r \in R} prod_r(m) \cdot rate^{on}_{S,r} = \sum_{r' \in R} react_{r'}(m) \cdot rate^{on}_{S,r'}}
$$

$$
(\text{RATE}_{ko})\ \frac{r \in R \quad \kappa_r = ma_{c_r} \quad enz(r) = (m_1, \dots, m_j) \quad react(r) = (m_1, \dots, m_k)}{rate^{on}_{S,r} = c_r \cdot S_{m_1} \cdot \dots \cdot S_{m_k} \cdot on_r}
$$

Fig. 5. Knockout steady states semantics of mass action reaction networks R over M.

knockouts. O may state that the rate of some reactions r should be increased (denoted $inc(r)$), decreased ($dec(r)$), that a reaction may not be knocked out ($on(r)$), or conjunctions thereof. We then try to find a set of reactions $R' \subseteq R$ to knockout, i.e. a knockout strategy, such that there exist solutions S, S' that are steady states for R and $R \setminus R'$, respectively, and satisfy O. Notice that this knockout problem could also be defined to compare several steady states that are possibly reached before and after applying some knockout. We leave such extensions to future work, see Section 9.

As a simple example consider the reaction network in Figure 4 and the objective to increase the rate of reaction r_4. Intuitively, this can be achieved by knocking out reaction r_3, such that more of the A molecules produced by reaction r_1 are transformed to B. A higher concentration of B in return leads to an increase in the speed of reaction r_4. Another idea could be to knockout reaction r_5. Then, however, the C molecules produced by reaction r_3 are no more consumed. Consequently, the concentration of C continuously increases, such that a steady state can never be reached.

In order to generally solve the above problem, we have to reason about the steady state semantics of a system of chemical reactions and all its subsets at the same time. Therefore, we first introduce the notion of reaction knockout in the semantics (knockout steady state semantics) and then we reduce the problem to reasoning with a single set of arithmetic constraints.

Knockout Steady State Semantics. We enrich our steady state semantics, such that it supports the knockout of a subset of chemical reactions. The idea is to introduce a Boolean value $on_r \in \{0,1\}$ for all reactions $r \in R$, which expresses whether or not reaction r is switched on (that is $on_r = 1$). This leads us to the knockout steady state semantics in Fig. 5.

$$(\text{ARITHM}_{con})\ \phi_R = \bigwedge_{m \in M} \phi_m \wedge \bigwedge_{r \in R} \phi_r$$

$$(\text{SPEC}_{con})\ \frac{m \in M}{\phi_m = \left(\sum_{r \in R} prod_r(m) \cdot x_r = \sum_{r' \in R} react_{r'}(m) \cdot x_{r'}\right)}$$

$$(\text{RATE}_{con})\ \frac{r \in R \quad \kappa_r = ma_{c_r} \quad enz(r) = (m_1, \dots, m_j) \quad react(r) = (m_1, \dots, m_k)}{\phi_r = (x_r = c_r \cdot x_{m_1} \cdot \ldots \cdot x_{m_k} \cdot x_{on_r} \wedge x_{on_r} \in \{0, 1\})}$$

Fig. 6. Knockout constraint ϕ_R for mass action reaction networks R.

Arithmetic Constraints. We introduce the following set of variables:

$$V = \{x_r, x_{on_r} \mid r \in R\} \cup \{x_m \mid m \in M\}$$

Variables x_m denote the unknown concentration S_m, variables x_r the unknown reaction rate $rate^{on}_{S,r}$, and variables x_{on_r} the unknown Boolean value on_r. We then consider the following language of arithmetic constraints where $x, y, z \in V$ and $c \in \mathbb{R}_{\geq 0}$:

$$\phi \ ::= \ x + y = z \mid x \cdot y = z \mid x = c \mid x \in \{0, 1\} \mid \phi \wedge \phi' \mid \exists x.\phi$$

The conditions of the knockout semantics for a mass action reaction network R can now be expressed by the arithmetic constraint ϕ_R defined in Fig. 6. Notice that the constraint given there can be flattened easily, so that it belongs to the constraint language specified above. Notice further that equalities of the form $x = y$ can be expressed by $\exists z.(x + z = y \wedge z = 0)$. As an exemplary mapping from a reaction network to a knockout constraint, consider the one in Figure 4. One can see that the constraint has no solution when knocking out only reaction r_5, i.e. setting $x_{on_{r_5}} = 0$, since in that case we obtain the following contradiction:

$$
\begin{array}{ll}
x_{r_3} = x_{r_5} = x_C \cdot x_{on_{r_5}} = 0 & \text{by } x_{on_{r_5}} = 0 \\
\Rightarrow x_{r_3} = x_A \cdot x_{on_{r_3}} = 0 = x_A & \text{by } x_{on_{r_3}} = 1 \\
\Rightarrow x_{r_2} = x_A \cdot x_{on_{r_2}} = 0 & \text{by } x_A = 0 \\
\Rightarrow x_{r_1} = 2 \cdot x_{on_1} = 2 = x_{r_2} + x_{r_3} = 0 & \text{by } x_{on_{r_1}} = 1
\end{array}
$$

Let ν be a variable assignment into the domain $\mathbb{R}_{\geq 0}$. The constraint problem that we try to solve is now as follows: given a reaction network R, with its knockout constraint ϕ_R and an objective O, find variable assignments ν, ν', such that ν, ν' satisfy ϕ_R, for all reactions $r \in R$ it holds that $\nu(x_{on_r}) = 1$, and (ν, ν') satisfies objective O. The reactions r for which it holds that $\nu'(x_{on_r}) = 0$ then define our reaction knockout strategy.

4 Abstract Interpretation

The next idea is to reason about changes in concentrations in steady states when switching off reactions. This is done by interpreting arithmetic constraints abstractly into finite domain table constraints.

$+^\alpha$	$\uparrow$	$\Uparrow$	$\downarrow$	$\Downarrow$	$\sim$	$\approx$
$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow,\downarrow,\sim$	$\uparrow,\downarrow,\sim$	$\uparrow$	$\uparrow$
$\Uparrow$	sy.	$\Uparrow$	$\uparrow,\downarrow,\sim$	$\uparrow,\downarrow,\sim$	$\uparrow$	$\Uparrow$
$\downarrow$	sy.	sy.	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
$\Downarrow$	sy.	sy.	sy.	$\Downarrow$	$\downarrow$	$\Downarrow$
$\sim$	sy.	sy.	sy.	sy.	$\sim$	$\sim$
$\approx$	sy.	sy.	sy.	sy.	sy.	$\approx$

$\cdot^\alpha$	$\uparrow$	$\Uparrow$	$\downarrow$	$\Downarrow$	$\sim$	$\approx$
$\uparrow$	$\uparrow$	$\Uparrow$	$\uparrow,\downarrow,\sim$	$\Downarrow$	$\uparrow$	$\approx$
$\Uparrow$	sy.	$\Uparrow$	$\Uparrow$	$\approx$	$\Uparrow$	$\approx$
$\downarrow$	sy.	sy.	$\downarrow$	$\Downarrow$	$\downarrow$	$\approx$
$\Downarrow$	sy.	sy.	sy.	$\Downarrow$	$\Downarrow$	$\approx$
$\sim$	sy.	sy.	sy.	sy.	$\sim$	$\approx$
$\approx$	sy.	sy.	sy.	sy.	sy.	$\approx$

Fig. 7. Abstraction of binary addition and multiplication functions (sy. = symmetric).

Abstract Domain & Relations. We are interested in capturing the differences between pairs of nonnegative real numbers $(u, u') \in \mathbb{R}^2_{\geq 0}$. We distinguish the cases where $u > u'$, $u < u'$, and $u = u'$, and in addition those cases where u or u' are equal to 0. More formally, we define the following set of difference relations:

$$\Delta = \{\uparrow, \downarrow, \sim, \Uparrow, \Downarrow, \approx\} \subseteq \mathbb{R}^2_{\geq 0}$$

such that the following properties hold for all $u, u' \in \mathbb{R}_{\geq 0}$:

$$u \uparrow u' \Leftrightarrow 0 < u < u' \qquad u \Uparrow u' \Leftrightarrow 0 = u < u'$$
$$u \downarrow u' \Leftrightarrow u > u' > 0 \qquad u \Downarrow u' \Leftrightarrow u > u' = 0$$
$$u \sim u' \Leftrightarrow u = u' > 0 \qquad u \approx u' \Leftrightarrow u = u' = 0$$

Lemma 1. *For any $(u, u') \in \mathbb{R}^2_{\geq 0}$, there exists a unique $\delta \in \Delta$ such that $(u, u') \in \delta$.*

Given an arithmetic relation $p \subseteq \mathbb{R}^n_{\geq 0}$, we next define an abstract relation $p^\alpha \subseteq \Delta^n$, such that for all difference relations $(\delta_1, \ldots, \delta_n) \in \Delta^n$:

$$(\delta_1, \ldots, \delta_n) \in p^\alpha \Leftrightarrow \exists (u_1, \ldots, u_n) \in p\ \exists (u'_1, \ldots, u'_n) \in p.\ \bigwedge_{i=1}^n (u_i, u'_i) \in \delta_i$$

In particular, we define abstract multiplication and addition functions $\cdot^\alpha, +^\alpha \in \Delta^3$ from binary multiplication and addition functions, seen as ternary relations. The tables of these two relations are spelled out in Fig. 7.

Abstract Constraints. Abstract constraints are finite domain table constraints, whose variables have values in Δ, subject to constraints based on the abstract relations $+^\alpha$ and $\cdot^\alpha$. We consider the following language of abstract constraints where $x, y, z \in V$ and $\Delta' \subseteq \Delta$:

$$\psi ::= +^\alpha(x, y, z) \mid \cdot^\alpha(x, y, z) \mid x \in \Delta' \mid \psi \wedge \psi'$$

We first show who to compile objectives to abstract constraints:

$$[\![inc(r)]\!] = x_r \in \{\uparrow, \Uparrow\} \qquad [\![on(r)]\!] = x_{on_r} \in \{\sim\}$$
$$[\![dec(r)]\!] = x_r \in \{\downarrow, \Downarrow\} \qquad [\![O \wedge O']\!] = [\![O]\!] \wedge [\![O']\!]$$

9

The condition that initially all reactions are on is expressed by:

$$\bigwedge_{r \in R} x_{on_r} \in \Delta \setminus \{\approx, \Uparrow\}$$

We next use abstract interpretation in order to map arithmetic constraints to abstract constraints:

$$
\begin{aligned}
[\![x + y = z]\!] &= +^{\alpha}(x, y, z) & [\![x \cdot y = z]\!] &= \cdot^{\alpha}(x, y, z) \\
[\![x = c]\!] &= x \in \{\sim\}, \text{with } c > 0 & [\![x \in \{0, 1\}]\!] &= x \in \Delta \setminus \{\uparrow, \downarrow\} \\
[\![x = c]\!] &= x \in \{\approx\}, \text{with } c = 0 & [\![\phi \wedge \phi']\!] &= [\![\phi]\!] \wedge [\![\phi']\!] \\
[\![\exists x.\phi]\!] &= \exists x.[\![\phi]\!]
\end{aligned}
$$

Consider the constraint $x = c$ which means that x is a constant that cannot be changed. Therefore, it is interpreted as $x \in \{\sim\}$, if $c > 0$ or $x \in \{\approx\}$, else. Or consider the constraint $x + y = z \wedge y = 0$. This is expressed by the corresponding abstract constraint $+^{\alpha}(x, y, z) \wedge y \in \{\approx\}$.

A pair (ν, ν') of two variable assignments into $\mathbb{R}_{\geq 0}$ induces a variable assignment μ into Δ, such that $\mu(x)$ is the difference relation between $\nu(x)$ and $\nu'(x)$ which is unique by Lemma 1. That is for all $\delta \in \Delta$ and $x \in V$:

$$\mu(x) = \delta \Leftrightarrow (\nu(x), \nu'(x)) \in \delta$$

We say that a pair (ν, ν') satisfies ψ if and only if μ is a solution of ψ.

We are now able to compile knockout constraint satisfaction problems into abstract domains. Reconsider the example in Figure 4 with the objective to increase the rate of reaction r_4. Our objective is compiled by $x_{r_4} \in \{\uparrow, \Uparrow\}$ and the condition that initially all reactions are on is expressed by $\bigwedge_{i=1}^{5} x_{on_i} \in \Delta \setminus \{\approx, \Uparrow\}$. Furthermore, the constraint $\bigwedge_{i=1}^{5} x_{on_i} \in \{0, 1\}$ is mapped to $\bigwedge_{i=1}^{5} x_{on_i} \in \Delta \setminus \{\uparrow, \downarrow\}$, such that it holds $\bigwedge_{i=1}^{5} x_{on_i} \in \{\sim, \Downarrow\}$. To complete the translation, it only remains to replace arithmetic relations by their abstract interpretation. Solving the resulting constraint, we obtain that $x_{r_1} \in \{\sim, \Downarrow\}$, such that for any solution μ also satisfying our objective it holds that $\mu(x_{r_1}) = \sim$. Thus, there exist only two solutions, either a knockout of r_3 alone ($\mu(x_{on_3}) = \Downarrow$) or of both reactions r_3 and r_5. As for the real domain constraints, a knockout of only reaction r_5 does not give any solution, since we analogously obtain contradiction $x_{r_1} = \sim = x_{r_2} + x_{r_3} = \Downarrow$.

Correctness. We now show that the abstract interpretation is correct in that every solution of the real domain problem is reflected by the solutions of the abstract domain problem.

Proposition 1 (Correctness of abstract interpretation). *Let ϕ be an arithmetic constraint and ν, ν' variable assignments into $\mathbb{R}_{\geq 0}$. It holds that if ν, ν' satisfy ϕ then (ν, ν') satisfies $[\![\phi]\!]$.*

Proof. By induction on the definition of arithmetic constraints.

$\phi = x + y = z$ Let ν and ν' be both solutions of $x + y = z$. Then we have $\nu(x) + \nu(y) = \nu(z)$ and $\nu'(x) + \nu'(y) = \nu'(z)$. It follows from the definitions of μ and $+^\alpha$ that $(\mu(x), \mu(y), \mu(z)) \in +^\alpha$, i.e., μ is a solution of the abstract constraint $+^\alpha(x, y, z)$, i.e. of $[\![\phi]\!]$

$\phi = x \cdot y = z$ analogous to $+$.

$\phi = (x = c)$ If $c > 0$ then $\nu(x) = c = \nu'(x)$ and thus $\mu(x) = \sim$, i.e., μ satisfies $[\![\phi]\!] = x$. The case of ϕ being $x = 0$ is analogous.

$\phi = x \in \{0, 1\}$ It holds that $\nu(x)$ and $\nu'(x)$ belong to $\{0, 1\}$. There are 4 possible cases, showing that the difference relation $\mu(x)$ between $\nu(x)$ and $\nu'(x)$ must be either of $\{\Uparrow, \Downarrow, \sim, \approx\}$ and thus μ satisfies $[\![\phi]\!] = x \in \{\Uparrow, \Downarrow, \sim, \approx\}$.

$\phi = \exists x.\phi'$ We obtain that $[\![\phi]\!] = \exists x.[\![\phi']\!]$. Since ν, ν' satisfy ϕ, there exist x and $u, u' \in \mathbb{R}_{\geq 0}$, such that $\nu \cup (x, u), \nu' \cup (x, u')$ satisfy ϕ'. Thus, by induction hypothesis, it holds that $(\nu \cup (x, u), \nu' \cup (x, u'))$ satisfy $[\![\phi']\!]$ and thus that (ν, ν') satisfies $[\![\phi]\!]$.

$\phi = \phi_1 \wedge \phi_2$ Since ν, ν' satisfy both ϕ_1 and ϕ_2, the induction hypothesis provides that (ν, ν') satisfies $[\![\phi_1]\!]$ and $[\![\phi_2]\!]$. Thus, (ν, ν') also satisfies $[\![\phi_1]\!] \wedge [\![\phi_2]\!]$, which equals $[\![\phi]\!]$.

Proposition 1 states that every solution of a model in the real domain is reflected by its abstract interpretation. However, the converse does not hold. Consider, e.g., the constraint $\phi = \phi_1 \wedge \phi_2$, with $\phi_1 = (x_1 = x_2)$ and $\phi_2 = (x_1 = x_2 + x_3)$. For all ν that satisfy $\phi \wedge \phi'$, we obtain that $\nu(x_3) = 0$. However, with our abstraction interpretation, μ satisfies $[\![\phi]\!] \wedge [\![\phi']\!]$, even with $\mu(x_1) = \mu(x_2) = \mu(x_3) = \uparrow$. This is a correct approximation, since there exist pairs $(\nu_1, \nu_1'), (\nu_2, \nu_2')$, such that ν_1, ν_1' satisfy ϕ_1 and ν_2, ν_2' satisfy ϕ_2, and that, although differing, correspond both to μ. For example:

$$\nu_1 = \{(x_1, 1), (x_2, 1)\} \qquad \nu_2 = \{(x_1, 2), (x_2, 1), (x_3, 1)\}$$
$$\nu_1' = \{(x_1, 2), (x_2, 2)\} \qquad \nu_2' = \{(x_1, 4), (x_2, 2), (x_3, 2)\}$$
$$\nu_1(x_i) \uparrow \nu_1'(x_i), i \in \{1, 2\} \qquad \nu_2(x_i) \uparrow \nu_2'(x_i), i \in \{1, 2, 3\}$$

Such a constraint results, e.g., from applying rule ($\textsc{subst}_{con}$) to species A and B, considering the following cyclic reaction network:

$$(r_1)\ A \xrightarrow{ma_1} B \quad (r_2)\ B \xrightarrow{ma_1} A \quad (r_3)\ B \xrightarrow{ma_1}$$
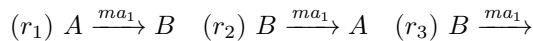
To this end, our approach, can be improved in different ways. On one hand, additional abstract relations could be defined, e.g. for commonly occurring patterns in reaction sets, like cycles of certain length. On the other hand, different methods could be applied to simplify equations. For example, one could use Gaussian elimination to symbolically solve the system of linear equations given by ϕ_1, ϕ_2 and account for the fact that $x_3 = 0$ by adding the constraint $x_3 \in \{\approx\}$. We leave such improvements as subject to future work.

5   Abstract Kinetics Functions

In the following, we extend our approach to more general kinetics. That is, we introduce properties of kinetics functions and show how kinetics functions fulfilling these properties are treated in our abstract interpretation.

The three properties of kinetics functions we consider are continuity, strict monotonicity, and conjunctiveness. Rates following strictly monotonic kinetics increase in the concentration of any reactant or enzyme. More formally, we call a kinetics function $\kappa : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ strictly monotonic if and only if for all $x_1, \ldots, x_n, x \in \mathbb{R}_{\geq 0}$ and all $i \in \{1, \ldots, n\}$ it holds:

$$x_i < x \Rightarrow \kappa(x_1, \ldots, x_n) < \kappa(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n)$$

Reactions that come with conjunctive kinetics can only perform if all its reactants and enzymes are present. More precisely, a kinetics function $\kappa : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is called conjunctive if and only of for all $i \in \{1, \ldots, n\}$ it holds:

$$\bigwedge_{i=1}^n x_i \neq 0 \Leftrightarrow \kappa(x_1, \ldots, x_n) \neq 0$$

In fact, the most widely used kinetics, mass action and Michaelis-Menten, are continuous, strictly monotonic, and conjunctive.

Lemma 2. *Any mass action function ma_c (with $c > 0$) is strictly monotonic, continuous, and conjunctive.*

Proof. Clear, since, by definition, we obtain $ma_c(x_1, \ldots, x_n) = c * \prod_{i=1}^n x_i$ for a reactions with order n. $\qquad\square$

Lemma 3. *The Michaelis-Menten function $mm_{c,c'}$ (with $c, c' > 0$) is strictly monotonic, continuous, and conjunctive.*

Proof. By definition, we obtain $mm_{c,c'}(x_1, x_2) = c * x_1 * x_2 / (c' + x_1)$. Strictly monotonic in x_2, continuous, and conjunctive clear. Strictly monotonic in x_1 becomes obvious by:

$$\frac{c * x_1 * x_2}{c' + x_1} = \frac{c * x_1 * x_2}{x_1 * (\frac{c'}{x_1} + 1)} = \frac{c * x_2}{\frac{c'}{x_1} + 1}$$

$\qquad\square$

We obtain that in our abstract interpretation any two kinetics functions of the same arity provide the same results.

Proposition 2. *Let $\kappa_1, \kappa_2 \subseteq \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be strictly monotonic, continuous, and conjunctive kinetics functions. It holds that $\kappa_1^\alpha = \kappa_2^\alpha$.*

The proof is elaborated in the Appendix of the extended version of this paper.

Since also mass action defines a strictly monotonic, continuous, and conjunctive kinetics function, we can represent any kinetics function by mass action in our abstract interpretation. In this way, we obtain complete independence from any kinetics information, except the three rather general properties listed above.

Corollary 1. *Every strictly monotonic, continuous, and conjunctive, n-ary kinetics function can be abstracted as abstract mass action kinetics ma_c^α of reactions with order n, with any c.*

Proof. By Lemma 2 and Proposition 2. $\qquad\square$

6 Constraint Solving

We apply the usual strategy of constraint programming to first propagate exhaustively and then to distribute. We use the common constraint propagation rules for table constraints. Let p be either of the relations $+^\alpha$ and $\cdot^\alpha$ and assume that we have a constraint $p(x_1, x_2, x_3)$. For all variables x_i we maintain a finite domain $\Delta_i \subseteq \Delta$ of possible values. We can then reduce the domain of variables x_j as follows:

$$\frac{j \in \{1, 2, 3\}}{x_j \in \{\delta_j \mid (\delta_1, \delta_2, \delta_3) \in p, \ \forall i \neq j. \ \delta_i \in \Delta_i\}}$$

The number of solutions would become huge if one tries to enumerate the values of all variables of the constraints $[\![\phi_R]\!] \wedge [\![O]\!]$. The usual method to deal with this problem is to impose a quality measure on solutions and to search only for high quality solutions. First of all, the fewer reactions are knocked out the better, since knockouts in the wet lab impose high costs. Second, the fewer impact the modifications have on the input-output environment of the network, the better. Which reactions are to be considered as inputs and outputs is to be specified (and is usually evident in the applications). The speed of such reactions should change only if required by the objectives, but as few as possible otherwise. This objective can be imposed by giving smaller weights to solutions that assign abstract values $\sim$ or $\approx$ to variables x_r of reactions reactions r.

The performance of a constraint solver largely benefits from such optimality considerations by the usual means of branch and bound. That is, one maintains a lower bound for the quality of the current pre-solution and only searches for solutions that are better or equally well as any solution found previously.

Further reduction of the solution set can be achieved by adding existential quantifiers to the model. Indeed, we are only interested in optimal knockout strategies, that is in the values of the variables x_{on_r} of all reactions and the values of variables x_r of input and output reactions. All other variables define internal fluxes, so that they can be considered as existentially quantified. That is, for every choice of values for these variables we will only verify whether there exists one possible choice for the values of the other variables. From this follows that the performance of our constraint solver largely depends on the number of variables that are considered as part of the optimality criterion.

7 Leucine Overproduction: A Case Study

In this section, we apply our approach to predict knockout strategies for the overproduction of Leucine in *B. subtilis*. Our model forms the current status of our work in progress to extend the efforts of modeling the metabolism of *B. subtilis* as presented in [19].

The reaction network is given in Figure 8. Molecules types are notated by ovals and reactions by boxes, respectively. A reaction's products are denoted by outgoing arrows, reactants by continuous, and enzymes by dashed lines. Red
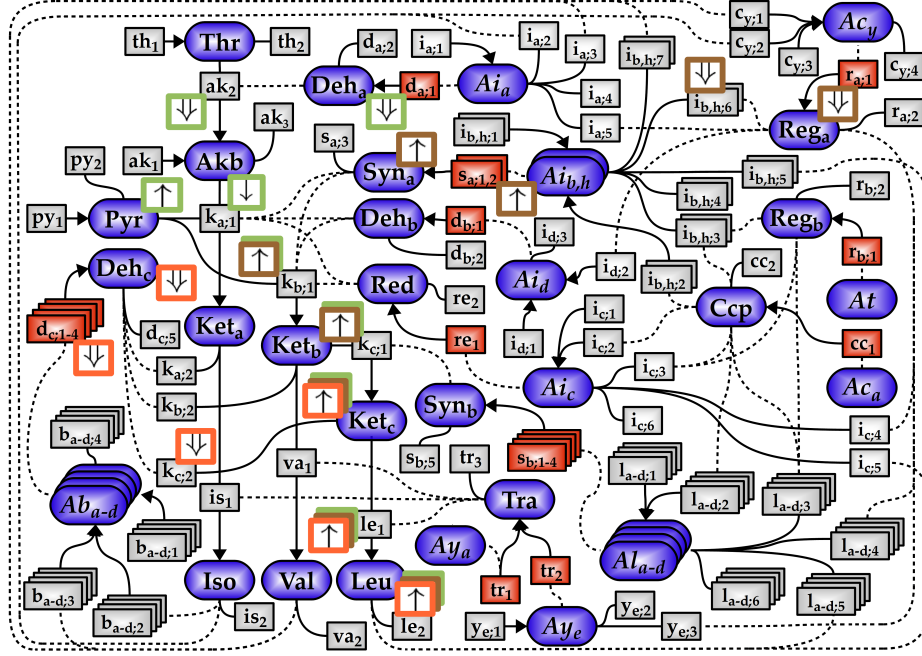
Fig. 8. A model of Leucine production in *B. subtilis* in graphical form. This model forms the current status of our work in progress to extend the efforts of modeling the metabolism of *B. subtilis* as presented in [19]. - Ovals graphically denote molecule types, rectangles reactions, arrows products and continuous and dashed lines reactants and enzymes, respectively. Red boxes mark reactions that may be knocked out. For improved readability, reactions and species are grouped where possible, which is denoted by overlapping boxes and ovals. For example the red overlapping boxes annotated by $(d_{c;1-4})$ on the left hand side denote reactions $(d_{c;1})Ab_a :\to Deh_c$, $(d_{c;2})Ab_b :\to Deh_c$, $(d_{c;3})Ab_c : Deh_c$, and $(d_{c;4})Ab_d :\to Deh_c$. The objective is to increase the rate of reaction (le_2). Arrows in boxes of different colors represent different knockout strategies.

boxes mark reactions that may be knocked out. For improved readability, reactions and species are grouped where possible, which is denoted by overlapping boxes and ovals. For example the red overlapping boxes annotated by $(d_{c;1-4})$ on the left hand side denote reactions $(d_{c;1})Ab_a :\to Deh_c$, $(d_{c;2})Ab_b :\to Deh_c$, $(d_{c;3})Ab_c : Deh_c$, and $(d_{c;4})Ab_d :\to Deh_c$. Our objective is to increase the rate of the Leucine secreting reaction (le_2). The reaction network in textual form and a legend for abbreviated species names is provided in the Appendix of the extended version of this paper.

In a first experiment, the goal was to find knockout strategies that are optimized w.r.t. a minimal number of knockouts. Based on our implementation, we could solve this task in about 5s on a Dell Latitude E6320 machine (Intel Core i7-2640M CPU, 2.8 GHz, 8 GB of memory).

Several single knockout solutions were proposed. Explanations for three of them are graphically annotated in Figure 8 by orange, brown, and green boxed arrows, respectively. The first one is to knock out one of the four reactions $(d_{c;1-4})$ (orange). This leads to the removal of species Deh$_c$ and therefore to a stop of reaction $(k_{c;2})$, i.e. the degradation of species Ket$_c$. In this way the rate of reaction (le_1) increases and so do the concentration of species Leu and the rate of reaction (le_2).

Knocking out reaction $(r_{a;1})$ (brown) leads to a total lack of Reg$_a$. Since this is involved in the degradation of Ai$_b$ and Ai$_h$ (reactions $(i_{b;4})$ and $(r_{a;1})$), the concentrations of Ai$_b$ and Ai$_h$ increase. Consequently, reactions $(s_{a;1})$ and $(s_{a;2})$ are accelerated, such that more Syn$_a$ is produced. This leads to an increase in the speed of reaction $(k_{b;1})$, by this raising the concentrations of Ket$_b$, Ket$_c$, and Leu. This knockout strategy confirms what is presented in literature (cf. [4], knockout of gene $codY$).

The deletion of reaction $(d_{a;1})$ (green) also leads to an acceleration of reaction $(k_{b;1})$. When disabling reaction $(d_{a;1})$, Deh$_a$ is entirely removed from the system, such that the transformation of Thr to Akb (reaction (ak_2)) stops. Thus, the concentration of Akb is decreased, such that reaction $(k_{a;1})$ is slowed down. Therefore, the concentration of Pyr is increased, augmenting the rate of reaction $(k_{b;1})$.

We performed a second experiment, where we were interested in optimizing additionally subject to the number of side effects w.r.t. to species Thr,Akb,Pyr,Iso, and Val. Therefore, we set reactions (th_2), (ak_3), (py_2), (is_2), and (va_2) to be output reactions. For example the green strategy is likely to have a side effect on species Akb, because it decreases the rate of reaction (ak_2). Since in steady state, the sum of the rates of reactions (ak_1) and (ak_2) has to equal the sum of the rates of reactions (ak_3) and $(k_{a;1})$, this may lead to a decrease of the speed of reaction (ak_3), which in return can only be achieved by lowering the concentration of Akb. Alternatively, the rate of reaction $(k_{a;1})$ could be decreased by reducing the concentrations of Pyr or Syn$_a$. However, this may require additional knockouts, e.g. in case of Syn$_a$, and again lead to different side effects.

The computation of the experiment took about 163s on the same machine. We obtained that the orange strategy and the combination of the brown and the green strategy are valuable candidates.

We further tested the obtained strategies on false solutions by mapping them to finite integer domain problems. This was done based on the knockout semantics in Figure 6 and by encoding the results into additional constraints. That is, given the knockout constraint ϕ of our reaction network with variables $x_1, \ldots, x_n$ and solution μ, we formulated the finite integer domain problem $\phi \wedge \phi' \wedge \bigwedge_{i=1}^n \phi_{x_i}$ over variables $x_1, x_1', \ldots, x_n, x_n'$, where ϕ' is the constraint ϕ but with variables $x_1' \ldots, x_n'$ and ϕ_{x_i} is the encoding of the abstract value $\mu(x_i)$ into a constraint over the pair (x_i, x_i') according to our abstract interpretation of real value pairs. As a result, we obtained, with variable domains $\{1, \ldots, 1000\}$, solutions for all proposed strategies, providing that none of them denotes a false abstract solution.

15

Although this supports the validness of our proposition, in practice the test on false solutions can in general only be considered a side note. We expect the relevance of proposed modification strategies to be more significantly impacted by the validity of models which needs to be continuously improved and should eventually lead to the integration of further available biological information, such as gene expression or flux data [34] (see also Section 9). Currently, the combination of the green and the brown strategy is tested in wet-lab experiments.

8   Related Work

Most existing approaches to solve reaction knockout problems formulate two-level optimization problems [5, 27, 28, 26, 36, 23]. Thereby, the first level of optimization bases on the idea of flux balance analysis [39, 29, 3, 9]. It captures only the equations stating that for any species the sum of the rates of the producing and the consuming reactions equal (Figure 2, rule (SPEC)). The constraints on reaction rates (variables $rate_{S,r}$) as introduced by rule (RATE$_{st}$) are omitted. Since the resulting sets of equations are hopelessly underconstraint, determining the values of variables $rate_{S,r}$ is considered to be an optimization problem with different kinds of optimization goals, e.g. biomass production (optimal growth) [38, 5] or ATP production (optimal energy) [30].

The intuition behind this first level of optimization is that, considering the background of evolution, organisms are assumed to be trimmed in a way, such that they always regulate their metabolism for optimal chances of survival. The second level of optimization is then concerned with finding the gene knockout strategy that yields the highest rate for the given target reaction, as determined by the first optimization level. Such two-level optimization problems are then solved by using e.g. integer linear programming approaches [5, 23] or evolutionary algorithms [26].

Predicting knockout strategies by two-level optimization is appealing, since it projets the problem to a well-founded mathematical domain. However, we see several drawbacks of this approach. On one hand, by dropping the constraints on reaction rates the relations between concentrations and reaction rates are lost. For example, the increase in the rate of a reaction is usually caused by the increase in the concentration of some reactant. If this reactant takes also part in other reactions then it is likely to cause changes in their rates, too. When not considering rate constraints such side effects will not show. This is critical, since in particular negative feed-back loops that are a common theme in reaction networks cannot be taken into account.

On the other hand, whether the assumptions used to define optimization goals are appropriate is controversial [35, 31]. A major problem commonly listed is that artificially created organisms did in fact not face evolutionary pressure, such that they may control their metabolism in unexpected ways. To this end, on one hand, a reasoning is presented in [31] that is based on the maximal and minimal bounds of reaction rates. These are then obtained by a separate optimization procedure for each reaction. On the other hand, in [35], it is proposed to use the

assumption of minimization of metabolic adjustment. This approach is similar to ours in that it compares reaction rates values before and after modification. The reaction rates before modifications are obtained by applying the optimal growth assumption. The rate values resulting from modification are then optimized to diverge as few as possible from their original value.

Finally, by using optimization approaches, it is not possible to apply a local reasoning that considers parts of metabolic networks as in Section 7, where only the production branched-chain amino acids (Leucine, Valine, Isoleucine) is modeled. The reason is that optimization subjects apply to specific parts of metabolisms that thus always need to be considered, such as the Glycolysis pathway or the TCA cycle. This requires a reasoning on rather large models. Different models that aim to capture the entire metabolism of micro-organisms have been proposed so far [20, 25, 16]. However, with the size of models also their uncertainty level increases which in general crucially hampers the application of formal reasoning. Additionally, this approach may favour predictions which apply to different parts of a metabolism and impact the reaction network in a more global manner with unwanted side effects [15]. Furthermore, a more local reasoning is favorable because resulting predictions are easier to explain (cf. explanations given in Figure 8 by arrows in colored boxed that represent traces of in-/decreases). Such kind of explanations are essential in order to communicate prediction results to experts from the domain of biology and to convince them of their validity.

With other methods in the field of analyzing chemical reaction networks based on abstract interpretation [10, 12, 6], our approach has in common the abstraction of value domains. This can be seen by lifting our abstraction to sets of solutions: reconsider the knockout constraint semantics for reaction networks as provided in Figure 6. Let $\wp(A)$ be the power set of set A and let $A \to B$ be the set of all functions from set A to set B. Given a constraint with variable set V, the set of possible solutions of the corresponding knockout problem lies in $\wp((V \to \mathbb{R}_{\geq 0})^2)$, where $\wp$ denotes the power set. Based on this idea and the abstaction of real value pairs to abstract values as given in Section 4, we can define the abstraction function $\alpha : \wp((V \to \mathbb{R}_{\geq 0})^2) \to \wp(V \to \Delta)$ and the concretization function $\gamma : \wp(V \to \Delta) \to \wp((V \to \mathbb{R}_{\geq 0})^2)$. From α, γ, and the partial order given by set inclusion $\subseteq$, we can then build the usual Galois connection [7].

However, the major difference to other approaches denotes their focus on dynamics, i.e. the consideration of changes in molecule amounts over time [10, 12, 6]. The idea is that vectors of species concentrations provide states and occurences of reactions represent state transitions. In [12, 10], e.g., this state space is understood to provide a small step semantics which is then usually abstracted by a collecting semantics. By contrast, we only consider a single state change (between two steady states) that is directly encoded into our abstract domain and waive the idea of abstracting small step semantics.

17

9   Conclusion and Future Work

We have introduced an approach for predicting knockout strategies in reaction networks with partial kinetic information, based on abstract interpretation and constraint solving. We showed that our approach is independent of any particular choice of kinetics functions, as long as these are continuous, strictly monotonic and conjunctive. Our predictions remain candidates due to the over-approximating nature of our abstraction.

A major subject for future work is to find ways to reduce the number of false solutions. On one hand, we plan to integrate methods to simplify systems of equation, linear and non-linear, as they result from our knockout semantics. On the other hand, we would like to come up with less aggressive abstract interpretations, so that one can predict weights of knockout effects. To this end, sources for more detailed kinetic information shall be developed, e.g. gene expression or flux data [34].

As a further subject, we also plan to integrate new optimization criteria for solutions. For example, one could consider, instead of only one, sets of solutions of the constraint satisfaction problem that correspond to the same knockout strategy. As each solution possibly represents a pair of steady states that are reached before and after a knockout is applied, it would make sense to favor, e.g., those knockout strategies that provide an optimal ratio between the numbers of solutions that fulfill an objective and those that do not.

We hope that the provided methods will help us to obtain better knockout results for wet-lab engineering. Concrete case studies are on the way. We also hope that better prediction methods will increase the interest in improving the quality of existing reactions networks in synthetic biology.

References

1. Ernesto Andrianantoandro, Subhayu Basu, David K. Karig, and Ron Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Molecular systems biology*, 2(1):msb4100073–E1–msb4100073–E14, May 2006.
2. Steven A. Benner and A. Michael Sismour. Synthetic biology. *Nature Reviews Genetics*, 6(7):533–543, July 2005.
3. H. P. J. Bonarius, G. Schmid, and J. Tramper. Flux analysis of underdetermined metabolic networks: the quest for the missing constraints. *Trends in Biotechnology*, 15(8):308–314, August 1997.
4. Shaun R. Brinsmade, Roelco J. Kleijn, Uwe Sauer, and Abraham L. Sonenshein. Regulation of CodY Activity through Modulation of Intracellular Branched-Chain Amino Acid Pools. *J. Bacteriol.*, 192(24):6357–6368, December 2010.

5. Anthony P. Burgard, Priti Pharkya, and Costas D. Maranas. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and Bioengineering*, 84(6):647–657, 2003.

6. Ferdinanda Camporesi and Jérôme Feret. Formal reduction for rule-based models. In Michael Mislove and Joël Ouaknine, editors, *the 27th Conference on the Mathematical Foundations of Programming Semantics - MFPS 2011*, volume 276 of *Electronic Notes in Theoretical Computer Science*, pages 29–59, Pittsburgh, États-Unis, September 2011. Elsevier.

7. P. Cousot. The calculational design of a generic abstract interpreter. In M. Broy and R. Steinbrüggen, editors, *Calculational System Design*. NATO ASI Series F. IOS Press, Amsterdam, 1999.

8. Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. In *POPL*, pages 269–282, 1979.

9. M. W. Covert, C. H. Schilling, and B. Palsson. Regulation of gene expression in flux balance models of metabolism. *Journal of theoretical biology*, 213(1):73–88, November 2001.

10. Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Abstracting the differential semantics of rule-based models: Exact and automated model reduction. In *LICS*, pages 362–381. IEEE Computer Society, 2010.

11. Michael B. Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, January 2000.

12. François Fages and Sylvain Soliman. Abstract interpretation and types for systems biology. *Theor. Comput. Sci.*, 403(1):52–70, 2008.

13. Jérôme Feret, Thomas Henzinger, Heinz Koeppl, and Tatjana Petrov. Lumpability abstractions of rule-based systems. *Theoretical Computer Science*, 2012.

14. James E. Ferrell. Feedback regulation of opposing enzymes generates robust, all-or-none bistable responses. *Current biology : CB*, 18(6), March 2008.

15. Lope Florez, Katrin Gunka, Rafael Polania, Stefan Tholen, and Jorg Stulke. SPAB-BATS: A pathway-discovery method based on Boolean satisfiability that facilitates the characterization of suppressor mutants. *BMC Systems Biology*, 5(1):5+, 2011.

16. Jochen Förster, Iman Famili, Patrick Fu, Bernhard Ø. Palsson, and Jens Nielsen. Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. *Genome research*, 13(2):244–253, February 2003.

17. C. S. Gillespie. Moment-closure approximations for mass-action models. *IET systems biology*, 3(1):52–58, January 2009.

18. Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977.

19. Anne Goelzer, Fadia B. Brikci, Isabelle M. Verstraete, Philippe Noirot, Philippe Bessieres, Stephane Aymerich, and Vincent Fromion. Reconstruction and analysis of the genetic and metabolic regulatory networks of the central metabolism of Bacillus subtilis. *BMC Systems Biology*, 2(1):20+, 2008.

20. Christopher S. Henry, Jenifer F. Zinner, Matthew P. Cohoon, and Rick L. Stevens. iBsu1103: a new genome-scale metabolic model of Bacillus subtilis based on SEED annotations. *Genome biology*, 10(6):R69+, 2009.

21. Jay D. Keasling. Synthetic biology for synthetic chemistry. *ACS chemical biology*, 3(1):64–76, January 2008.

22. Boris N. Kholodenko. Cell-signalling dynamics in time and space. *Nature Reviews Molecular Cell Biology*, 7:165–176, 2006.

23. Joonhoon Kim and Jennifer Reed. OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains. *BMC Systems Biology*, 4(1):53+, 2010.

24. Tie Koide, Wyming Lee L. Pang, and Nitin S. Baliga. The role of predictive modelling in rationally re-engineering biological systems. *Nature reviews. Microbiology*, 7(4):297–305, April 2009.

25. You-Kwan Oh, Bernhard O. Palsson, Sung M. Park, Christophe H. Schilling, and Radhakrishnan Mahadevan. Genome-scale Reconstruction of Metabolic Network in Bacillus subtilis Based on High-throughput Phenotyping and Gene Essentiality Data. *Journal of Biological Chemistry*, 282(39):28791–28799, September 2007.

26. Kiran Raosaheb R. Patil, Isabel Rocha, Jochen Förster, and Jens Nielsen. Evolutionary programming as a platform for in silico metabolic engineering. *BMC bioinformatics*, 6(1):308+, 2005.

27. Priti Pharkya, Anthony P. Burgard, and Costas D. Maranas. OptStrain: A computational framework for redesign of microbial production systems. *Genome Research*, 14(11):2367–2376, November 2004.

28. Priti Pharkya and Costas D. Maranas. An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metabolic engineering*, 8(1):1–13, January 2006.

29. Nathan D. Price, Jennifer L. Reed, and Bernhard Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature reviews. Microbiology*, 2(11):886–897, November 2004.

30. R. Ramakrishna, J. S. Edwards, A. McCulloch, and B. O. Palsson. Flux-balance analysis of mitochondrial energy metabolism: consequences of systemic stoichiometric constraints. *American journal of physiology. Regulatory, integrative and comparative physiology*, 280(3):R695–704, March 2001.

31. Sridhar Ranganathan, Patrick F. Suthers, and Costas D. Maranas. OptForce: An Optimization Procedure for Identifying All Genetic Manipulations Leading to Targeted Overproductions. *PLoS Comput Biol*, 6(4):e1000744+, April 2010.

32. Guillermo Rodrigo, Javier Carrera, Thomas E. Landrain, and Alfonso Jaramillo. Perspectives on the automatic design of regulatory systems for synthetic biology. *FEBS Letters*, 586(15):2037–2042, July 2012.

33. F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming*. Elsevier, 2006.

34. Uwe Sauer. Metabolic networks in motion: 13C-based flux analysis. *Molecular systems biology*, 2(1), November 2006.

35. Daniel Segrè, Dennis Vitkup, and George M. Church. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences*, 99(23):15112–15117, November 2002.

36. Naama Tepper and Tomer Shlomi. Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, 26(4):536–543, February 2010.

37. R. Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585, December 1973.

38. Amit Varma and Bernhard O. Palsson. Metabolic Capabilities of Escherichia coli II. Optimal Growth Patterns. *Journal of Theoretical Biology*, 165(4):503–522, December 1993.

39. Amit Varma and Bernhard O. Palsson. Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. *Nature Biotechnology*, 12(10):994–998, October 1994.

40. Olaf Wolkenhauer, Mukhtar Ullah, Walter Kolch, and Kwang-Hyun H. Cho. Modeling and simulation of intracellular dynamics: choosing an appropriate framework. *IEEE transactions on nanobioscience*, 3(3):200–207, September 2004.