Qualitative Reasoning about Reaction Networks with Partial Kinetic Information

Joachim Niehren, Mathias John, Cristian Versari, François Coutte, Philippe Jacques

Qualitative Reasoning for Reaction Networks with Partial Kinetic Information

Joachim Niehren[3,2], Mathias John[1,2], Cristian Versari[1,2]
François Coutte[1,4], and Philippe Jacques[1,4]

Université de Lille[1] Inria Lille[3]
Cristal[2] Research Institute Charles Viollette[4]

Abstract. We propose a formal modeling language for reaction networks with partial kinetic information. The language has a graphical syntax reminiscent to Petri nets. The kinetics of reactions need to be described only partially, so that the language can be used to model the regulation of metabolic networks. We present a qualitative reasoning method based on abstract interpretation of the steady state semantics of reaction networks modeled in our language. In particular, we can predict changes of influxes that lead to expected changes of outfluxes.

1 Introduction

Models of reaction networks in systems biology often require full kinetic information, while only partial information on activators and inhibitors is available in practice. In order to become applicable nevertheless, the existing model-based reasoning methods often ignore any kinetic information. Most typically, this holds for flux balance analysis [10,12] when applied to metabolic networks [11,15]. The missing information is then compensated heuristically by the adoption of ad hoc optimization criteria. Alternatively, pathway analysis approaches [12] rely on the structure of reactions networks, but the combinatorial nature of the problem makes difficult their application to densely interconnected networks. To both methods boolean constraints can be added in order to account for inhibitors that block reactions completely [6]. But blocking inhibitors is not appropriate in deterministic semantics, where the average over blocked and unblocked situations is to be considered. The problem therefore is how to model reaction networks with partial kinetic information and how to reason with such models.

In this paper, we propose a modeling language for reaction networks with partial kinetic information. Our language is parameterized by a similarity relation on kinetic functions, so that the rate laws of chemical reactions need only to be specified up to similarity. For instance, we could define two kinetic functions to be similar if they have the same monotonicity behavior. For instance, $2A$ is similar to $5A/(7+A)$ since whenever A increases then both terms increase, and whenever A decreases than both terms decrease.

The models of reaction networks in our language have a graphical syntax that is reminiscent of Petri nets, and also an equivalent XML syntax. To any

model a standard steady state semantics can be assigned, which provides the usual flux balance equations and additional equations with variables for kinetic functions, that are subject to similarity constraints. The steady state semantics ensures that inhibitors slow down reactions, while activators speed them up. In particular, our language can be used to model metabolic networks with complex regulation such as for *B. subtilis* in the Subtiwiki [8]. As an example, we present in Fig. 3 the graphical model of the regulation network of the PIlv-Leu promoter of *B. subtilis*, which regulates the metabolism of the branched-chain amino acids Valine, Leucine, and Isoleucine. Previous models of these metabolic networks as in the Subtiwiki were not given any formal semantics, so that they could not be used for directly for qualitative prediction algorithms.

We then show how to lift the abstract interpretation method from [5] for qualitative reasoning [3] to models of reaction networks in our language. The main technical contribution is to overcome the previous limitation to mass action laws with unknown parameters. By applying abstract interpretation to the steady state semantics, we can abstract away the variables for kinetic functions, and discretize the available partial kinetic information. This yields so called difference constraints [5], which are finite domain constraints that can be solved by finite-domain constraint programming.

As an application of our qualitative reasoning method, we show how to predict changes of influxes when given the expected changes of the outfluxes. This can be done based on the difference constraints obtained from abstract interpretation, either by constraint simplification, by rules that we present in this paper, or else by constraint solving based on the solver from [5]. In particular, constraint simplification can be used for the PIlv-Leu network to predict that any increase of leucine outflux is due to a decrease of either the CodY influx or TnrA influx. For this simple example, a similar reasoning can be done by humans based on the graphical model. This illustrates that our algorithm formalizes a natural kind of qualitative reasoning. In a follow up work [2], the same method is extended to the prediction of gene knockouts leading to the overproduction of some target metabolites [13]. The arguments used there are by far too complicated to be performed manually without any computational support for qualitative reasoning.

2 Reaction Networks

We define reaction networks with complete kinetic information, and show how to compute their steady state semantics. This is basically standard, except for the treatment of inflows and outflows of reaction networks, by which we can model the interaction of the reaction network with its context. In this way, any reaction network can be considered as "module" of a larger biological system, or as part of a chemical experiment that interacts with the network.

Let $\mathbb{R}_+$ be the set of non-negative real numbers, S a finite set of species, and $\prec$ an arbitrary total order on S. A kinetic function of arity $k \geq 0$ is a function of type $\kappa : \mathbb{R}_+^k \to \mathbb{R}_+$. Kinetic functions will be used to define the

$$\text{(INFLOWS)} \ \frac{s \in S \setminus I}{x_s = 0} \qquad \text{(OUTFLOWS)} \ \frac{s \in S}{y_\mathrm{s} = O(s) \cdot z_\mathrm{s}}$$

$$\text{(CONS)} \ \frac{s \in S}{c_s = y_\mathrm{s} + \sum_{r \in R} \mathsf{rct}_r(s) \cdot v_r} \qquad \text{(PROD)} \ \frac{s \in S}{p_s = x_s + \sum_{r \in R} \mathsf{prd}_r(s) \cdot v_r}$$

$$\text{(RATE)} \ \frac{r \text{ is } s_1, \ldots, s_k \xrightarrow{\kappa} s_{k+1}, \ldots, s_l}{v_r = \kappa(z_{\mathrm{s}_1}, \ldots, z_{\mathrm{s}_\mathrm{k}})} \qquad \text{(STEADY STATE)} \ \frac{s \in S}{c_s = p_s}$$

Fig. 1: Steady state equations of a reaction network $N = (S, R, I, O)$.

rate laws of chemical reactions. A chemical reaction r is a tuple of the form: $s_1, \ldots, s_k \xrightarrow{\kappa} s_{k+1}, \ldots, s_l$ where $0 \le k \le l$, $s_1, \ldots, s_l \in S$, and $\kappa : \mathbb{R}_+^k \to \mathbb{R}_+$ is a kinetic function. Any reaction has a tuple of reactants $s_1, \ldots, s_k$ and a tuple of products $s_{k+1}, \ldots, s_l$. In order to account for the stoichiometry of a reaction, we write $\mathsf{rct}_r(s)$ for number of occurrences of s in the tuple of reactants of r, and $\mathsf{prd}_r(s)$ for number of occurrences of s in the tuple of products of r. A modifier of a reaction is a species s with $\mathsf{rct}_r(s) = 1 = \mathsf{prd}_r(s)$. Whether a modifier behaves as an activator or as an inhibitor depends on the choice of the rate law κ.

Definition 1. *A reaction network over a species set S is a triple $N = (S, R, I, O)$ where R is a finite set of chemical reactions over S, a set of inflow species $I \subseteq S$, and an outflow function $O : S \to \mathbb{R}_+$.*

A reaction network defines the evolution of a chemical solution in a context. Each inflow species $s \in I$ specifies an inflow that adds s to the chemical solution, and is controlled by the context. An *outflow species* is an element $s \in S$ with $O(s) \ne 0$; for any outflow species there is outflow into the context that consumes s from the chemical solution. The outflow kinetics for s follows the mass-action law with constant $O(s)$. Note that any species may have an inflow and an outflow at the same time.

Under the assumption of deterministic network behavior, for any initial chemical solution a unique limit will be reached that is called a steady state. Since we do not fix any initial chemical solution, many steady states may exist for the same reaction network. The rates of all inflows and and outflows are also assumed to be constant in any steady state, as well as the rates of all reactions and the concentrations of all species of the network.

The steady state semantics of a reaction network is given by a system of arithmetic equations. These use the following variables taking values in $\mathbb{R}_+$. For any species $s \in S$, there is a variable z_s that denotes the concentration of s in a steady state, a variable x_s that denotes the rate of the inflow, which also called the *influx*, and a variable y_s that stands for the rate of the outflow, which also called the *outflux*. For any reaction $r \in R$, variable v_r stands for the rate of reaction r in a steady state.

The steady state equations are inferred from the network by the inference rules in Fig. 1 which are mainly standard. Each inference rule can be seen as an

Fig. 2: An enriched reaction with a partially known rate law $\sim\kappa'$. It has substrate S, inhibitor I, accelerator A', activator A, and one product P beside of the modifiers I, A, and A'.
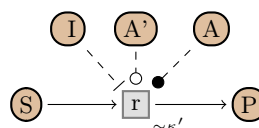
implication, whose condition is written above the line and whose conclusions is written below the line. Rules (INFLOW) states that the influx for any non-inflow species $s \notin I$ is zero. Rule (OUTFLOW) requires for any species $s \in S$ that its outflux is equal to $O(s) \cdot z_s$ according to the mass-action law. The production rate p_s of a species s is defined by rule (PROD) and its consumption rate c_s by rules (CONS). Rule (RATE) provides the rate of reaction r by applying its kinetic function to the concentrations of all its reactants. The (STEADY STATE) states that consumption and production rates are balanced for all species.

Definition 2. *Any reaction network N with n inflow and m outflow species defines a exchange relation $R_N \subseteq \mathbb{R}_+^n \times \mathbb{R}_+^m$, determined by the solutions of the steady state equations for N, when projected to the n-tuple of variables x_s for the inflow species s of N and the m-tuple of variables $y_{s'}$ for the outflow species s' of N. The order of both tuples is given by the order $\prec$ on S.*

3 Modeling Language

We now present a modeling language for reaction networks with partial kinetic information. As first parameter of our language, we assume a similarity relation $\sim$ on kinetic functions. Rather than specifying rate laws of chemical reactions by kinetic functions, we will describe them only up to similarity: a rate law belongs to $\sim\kappa$ if it is similar to the kinetic function κ.

Enriched chemical reactions will be used to describe the chemical reactions of a reaction network. An example is given in Fig. 2. The graph there represents an enriched chemical reaction r with substrate S, activator A, an accelerator A', and inhibitor I and a product P. Please note that the same species may play different roles even in the same reaction, and several times. Both, activators and accelerators speed up a reaction. Activators are like enzymes. The difference is that all activators of a reaction must be present for its application, while the accelerators need not to be there.

For graphical representation, we use conventions similarly to Petri nets. Species are represented by rounded nodes Ⓢ containing the name s of the species, and enriched reactions are graphically represented by boxed nodes r containing the name r of the reaction. More generally, enriched chemical reactions have different kinds of reactants, that are fixed by a finite set of roles *Rol*, which is the second parameter of our language. In our example, there will be substrates – that are consumed – and three kinds of modifiers: inhibitors, activators, and accelerators, so we set *Rol* = {*inh, subs, act, acc*}. For our graphical syntax, we assign to each role an edge type, for edges pointing from the reactant

to the reaction. We will use $\longrightarrow$ for $subs$, $----|$ for inh, $---\bullet$ for act, and $---\circ$ for acc. The products of a reaction – beside of the above modifiers – will be linked by arrows $\longrightarrow$ pointing from the reaction to the product.

Reactant roles serve to order the arguments of the rate law of a enriched chemical reaction. Such a rate law is given by an enriched kinetic function:

$$\kappa' : (Rol \times \mathbb{R}_+)^k \to \mathbb{R}_+$$

We assume that any enriched kinetic function is well-behaved, in that any permutation of arguments with the same role does not change its value. When fixing the order of the arguments, any enriched kinetic function κ' can be replaced by a standard kinetic function κ, for instance such that $\kappa(z_S, z_I, z_{A'}, z_A) = \kappa'(subs{:}z_S, inh{:}z_I, act{:}z_{A'}, acc{:}z_A)$. An enriched chemical reaction can then be replaced by a chemical reaction, in which the kinetic function is replaced by a variable. With the same ordering as for obtaining κ from κ', we obtain for the example from Fig. 2:

$$S, I, A', A \xrightarrow{\sim\kappa} P, I, A', A$$
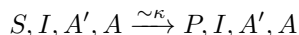
Here, $\sim\kappa$ stands for a fresh variable for a standard kinetic function that is similar to κ. A model in our language is a tuple (S, R, I, O) where R is a set of enriched reactions and $I, O \subseteq S$. Note that we do not require to specify rate constants for outflows. Graphically, inflow species in I and outflow species in O are indicated respectively by ingoing and outgoing arrows $\cdots\cdots\triangleright$. An example model in graphical syntax is given in Fig. 3.

For any model in our language, we can generate a reaction network with variables for kinetic functions that are subject to similarity constraints. Therefore, we can define the steady state equations of any model in the language as before, except that kinetic functions will be represented by variables, as well as rate constants of outflows. An example is worked out in the next section.

Besides the graphical syntax, our language supports an XML syntax, which serves for writing the models, so that the graphs can be generated. We implemented tools for doing this in XSLT. These tools can also compute the steady state equations, and perform abstract interpretation.

4 Example: Regulation of Metabolism of *B. subtilis*

As an example, we model the leucine biosynthesis pathway of *B. subtilis* in our language. This is one of the complex regulation mechanisms of the metabolism of *B. subtilis*, for which informal models are given in the Subtiwiki [8]. The precise similarity relation of the model will be defined in Section 5.

The resulting model in graphical syntax is given in Fig. 3. For clearer visualization, nodes have different colors depending on the type of the species: in this paper we will use proteins $\text{\textcircled{P}}$, metabolites $\text{\textcircled{M}}$, and promoters or binding sites $\text{\textcircled{B}}$. The variable z_B stands for the activity of the promoter or binding site B, while z_P and z_M stand for the concentrations of P and M.
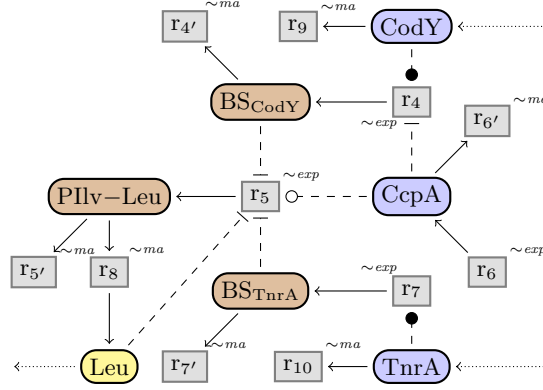
Fig. 3: Reaction network of the regulation of promoter PIlv-Leu in *B. subtilis*.

We consider an acceleration function with $Acc(d) = 1 + d$ and an inhibition function with $Inh(d) = 1/Acc(d)$. We define the enriched kinetic functions *exp* such that for all tuples $t = (r_1 : d_1, \ldots, r_k : d_k) \in (Rol \times \mathbb{R}_+)^k$:

$$exp(t) = \prod_{r_i \in \{subs, act\}} d_i \; \cdot \; \prod_{r_j = acc} Acc(d_j) \; \cdot \; Inh(\textstyle\sum_{r_l = inh} d_l)$$

Note that the order of arguments with the same role is not important, so that function *exp* is well-behaved. When a reaction has the *exp* kinetics, then its inhibitors slow down the reaction but do not block it. Accelerators and activators both speed up the reaction. Furthermore, if one of the activators is missing then the reaction is blocked. One might want to generalize *exp* with parameters defining the strenght of respective accelerations and inhibitions. We do not do so, since these parameters are typically unknown, and since all such generalized expression kinetics will turn out to be similar. Generally, we are only interested in $\sim exp$, so similar definitions would to the job as well. The enriched mass-action kinetics is the special case $ma(t) = exp(t)$ for all $t \in (\{subs\} \times \mathbb{R}_+)^*$.

Leucine biosynthesis is realized by enzymes which are coded by the genes of the *ilv-leu* operon. This operon is under the regulation of the promoter PIlv−Leu. For simplicity, we group the whole reaction network leading to the leucine biosynthesis into reaction r_8. The activation of PIlv−Leu is done by reaction r_5, under regulation by TnrA, CcpA, and CodY. Proteins TnrA and CodY are influx species added by the context and degraded by reactions r_{10} and r_9 respectively. Protein CcpA is expressed by reaction r_6 and degraded by reaction $r_{6'}$. Transcription at the *ilv-leu* promoter is well known to be inhibited by CodY through a binding of this latter on the promoter [7,9,14,16,17]. To model this action of CodY on the promoter PIlv−Leu, we introduce the reaction r_4 which activates the binding side BS_{CodY} of CodY at the promoter, which in turn slows down reaction r_5 and thus reduces the promoter's activity. The binding of CodY to the promoter's binding site BS_{CodY} can be prohibited when CcpA is bound to the promoter. Therefore the presence of CcpA slows down reaction r_4 [1,16]

Flux balance equations:

(Leu) $v_{r_8} = y_{Leu}$

(CcpA) $v_{r_6} = v_{r_{6'}}$

(CodY) $x_{CodY} = v_{r_9}$

(TnrA) $x_{TnrA} = v_{r_{10}}$

(BS_{CodY}) $v_{r_4} = v_{r_{4'}}$

($PIlv-Leu$) $v_{r_5} = v_{r_{5'}} + v_{r_8}$

(BS_{TnrA}) $v_{r_7} = v_{r_{7'}}$

Outfluxes:

$y_{Leu} = ma^{(8)}(subs\colon z_{Leu})$

Reaction rates:

$v_{r_4} = exp^{(1)}(inh\colon z_{CcpA}, act\colon z_{CodY})$

$v_{r_{4'}} = ma^{(1)}(subs\colon z_{BS_{CodY}})$

$v_{r_5} = exp^{(2)}(inh\colon z_{BS_{CodY}}, acc\colon z_{CcpA},$
$\qquad\qquad\qquad inh\colon z_{Leu}, inh\colon z_{BS_{TnrA}})$

$v_{r_{5'}} = ma^{(2)}(subs\colon z_{PIlv-Leu})$

$v_{r_6} = exp^{(3)}()$

$v_{r_{6'}} = ma^{(3)}(subs\colon z_{CcpA})$

$v_{r_7} = exp^{(4)}(act\colon z_{TnrA})$

$v_{r_{7'}} = ma^{(4)}(subs\colon z_{BS_{TnrA}})$

$v_{r_8} = ma^{(5)}(subs\colon z_{PIlv-Leu})$

$v_{r_9} = ma^{(6)}(subs\colon z_{CodY})$

$v_{r_{10}} = ma^{(7)}(subs\colon z_{TnrA})$

Fig. 4: Steady state equations for the PIlv-Leu network.

$v_{r_5} = v_{r_{5'}} + y_{Leu}$

$y_{Leu} = ma^{(8)}(subs\colon z_{Leu})$

$v_{r_4} = exp^{(1)}(inh\colon z_{CcpA}, act\colon z_{CodY})$

$v_{r_4} = ma^{(1)}(z_{BS_{CodY}})$

$v_{r_5} = exp^{(2)}(inh\colon z_{BS_{CodY}}, acc\colon z_{CcpA},$
$\qquad\qquad\qquad inh\colon z_{Leu}, inh\colon z_{BS_{TnrA}})$

$v_{r_{5'}} = ma^{(2)}(subs\colon z_{PIlv-Leu})$

$v_{r_6} = exp^{(3)}()$

$v_{r_6} = ma^{(3)}(subs\colon z_{CcpA})$

$v_{r_7} = exp^{(4)}(act\colon z_{TnrA})$

$v_{r_7} = ma^{(4)}(subs\colon z_{BS_{TnrA}})$

$y_{Leu} = ma^{(5)}(subs\colon z_{PIlv-Leu})$

$x_{CodY} = ma^{(6)}(subs\colon z_{CodY})$

$x_{TnrA} = ma^{(7)}(subs\colon z_{TnrA})$

Fig. 5: Simplified steady state equations for the PIlv-Leu network.

but it does not block it on average in a steady state. The promoter PIlv−Leu is also down-regulated by Leu in terms of a T-box [1,4], which is captured by the negative control of the reaction r_5 by Leu. Protein TnrA forms a further inhibitor whose impact on the PIlv−Leu promoter is represented by the binding side BS_{TnrA} through the reaction r_7. Protein CcpA is also independently up-regulating the *ilv-leu* operon transcription, and thus activating reaction r_5.

From the model, the steady state equations in Fig. 4 were inferred. These contain variables $exp^{(i)}$ for enriched kinetic functions similar to *exp*, and variables $ma^{(i)}$ for enriched kinetic functions similar to the mass-action law *ma* for any i. The equations can be simplified by replacing local variables by equal terms, yielding the equations in Fig. 5.

In order to illustrate the qualitative reasoning methods that we will develop, we consider the overproduction problem of Leu for the PIlv−Leu network. The question is which changes of the influxes may lead to an increase of the Leu outflux? Informally, the problem can be solved as follows. Leu is produced only from PIlv−Leu, which is solely produced by reaction r_5, so the speed of r_5 must be increased. This can be done by either decreasing one of its three inhibitors Leu, BS_{CodY}, or BS_{TnrA}, or by increasing its accelerator CcpA. But CcpA is not connected to any inflow, so it cannot be increased by changing the influxes. And inhibitor Leu cannot be decreased, when we want to increase its outflux. Hence,

either BS_{CodY} or BS_{TnrA} must be decreased. This is possible only by decreasing the influxes of CodY or TnrA.

5 Similarity by Difference Abstraction

We now recall the similarity relation $\sim$ on kinetic functions from [5], which is obtained by abstracting from changes between real numbers.

We are interested in changes of the network raised for example by modification of inflows or outflows. A change of a concentration or a flow rate from one steady state to another is given by a pair of positive real numbers. We now want to abstract the space of all changes in $\mathbb{R}_+ \times \mathbb{R}_+$ into a finite set of *difference relations*. For this, we partition the set $\mathbb{R}_+ \times \mathbb{R}_+$ into a finite collection of subsets $\Delta \subseteq 2^{\mathbb{R}_+ \times \mathbb{R}_+}$, so that we can abstract any change in $\mathbb{R}_+ \times \mathbb{R}_+$ into a *difference relation* of Δ. In the examples that follow, we will use the partition $\Delta = \{<, >, \doteq\}$ where the symbols represent "increase" $< = \{(x, y) \in \mathbb{R}_+^2 \mid x < y\}$, "decrease" $> = \{(x, y) \in \mathbb{R}_+^2 \mid x > y\}$ and "no change" $\doteq = \{(x, x) \mid x \in \mathbb{R}_+\}$.

In general, we assume a relation $R \subseteq \mathbb{R}_+^p$, that may be either a kinetic function κ of arity $p-1$ or the relation R_N of a reaction network with p in- and outflows. We define the set of Δ-differences of the p-ary relation R as follows:

$$R^\Delta = \{(\delta_1, \ldots, \delta_p) \in \Delta^p \mid \forall i.\ (d_i, d_i') \in \delta_i,\ (d_1, ..., d_p) \in R,\ (d_1', ..., d_p') \in R\}$$

So for instance, consider the exchange relation R_N for some reaction network N. Its difference abstraction $R_N{}^\Delta$ then expresses how the tuples in R_N may change when moving from one steady state of N to another.

Definition 3. *Two kinetic functions $\kappa_1, \kappa_2 : (\mathbb{R}_+)^{p-1} \to \mathbb{R}_+$ are similar, written $\kappa_1 \sim \kappa_2$, iff $\kappa_1^\Delta = \kappa_2^\Delta$.*

Example 1. Let $ma_k(subs{:}d_1, subs{:}d_2) = k \cdot d_1 \cdot d_2$ be the mass action law with constant k. As usual, we identify binary functions with ternary relations. The differences abstraction $ma_k{}^\Delta$ is then equal for all choices of parameter k; it contains all triples of difference relations $(\delta_1, \delta_2, \delta_3) \in \Delta^3$ given in the table on the right.

δ_1	δ_2	δ_3
$<$	$<$	$<$
$<$	$>$	$<, \doteq, >$
$<$	$\doteq$	$<$
$>$	$<$	$<, \doteq, >$
$>$	$>$	$>$
$>$	$\doteq$	$>$

Example 2. We consider an enhanced Michaelis-Menten law with an additional activator: $mm_{k_1,k_2}(subs{:}d_1, act{:}d_2) = d_2 \frac{k_1 \cdot d_1}{k_2 + d_1}$. Again, it can be shown that the abstraction $(mm_{k_1,k_2})^\Delta$ is independent of the choice of $k_1, k_2 \in \mathbb{R}_+$. Indeed, it is equal to $ma_k{}^\Delta$ for all parameters k, i.e.: mass-action and the enhanced Michaelis-Menten kinetics are similar with respect to Δ-abstraction, where $\Delta = \{<, >, \doteq\}$. Of course, there exist more precise difference sets Δ for which the two families of kinetics can be distinguished.

It should be noticed that R^Δ is always a finite relation, since Δ is chosen to be finite. The relation R in contrast, may contain infinitely many tuples. As a consequence, infinitely much information may be abstracted away, in particular

the details about the parameters of kinetic functions. This is why the relations $ma_k{}^{\Delta}$ and $(mm_{k_1,k_2})^{\Delta}$ in the above examples could be computed independently of the parameters. The information that is preserved, however, is still able to distinguish inhibitors and activators.

6 Abstract Interpretation to Difference Constraints

We next show how to interpret steady state equations abstractly as difference constraints, which will then be used for qualitative reasoning about reaction networks in our language in the next section.

The idea is to lift the difference abstraction $.^{\Delta}$ from relations over $\mathbb{R}_+$ to relations over Δ to the level of constraints defining such relations. For instance, the arithmetic equation $x_A = ma_k(z_A, z_B)$ can be abstracted to a difference constraint that defines the relation $ma_k{}^{\Delta}$. We write this difference constraint as $x_A \in ma_k(z_A, z_B)$, since now the variables are interpreted by values of Δ and ma_k is interpreted as the set valued function $ma_k{}^{\Delta}$. It should be noticed that the relation $ma_k{}^{\Delta}$ is finite and independent of the unknown parameter k, i.e., the unknown parameter has been abstracted away successfully.

Arithmetic constraints were used to define the steady state semantics of reaction networks. These are built from a totally ordered set of variables including those from the steady state equations. More formally, an arithmetic constraint is a conjunctive logic formula with existential quantifiers with the following abstract syntax:

$$\phi \;::=\; x{=}\kappa^{(i)}(x_1,\ldots,x_k) \mid x = x_1 + x_2 \mid x_1 = x_2 \mid \phi \wedge \phi' \mid \exists x.\phi$$

where $i \in \mathbb{N}$, $\kappa : \mathbb{R}_+^k \to \mathbb{R}_+$, and all x'es are variables. The expression $\kappa^{(i)}$ is a variable for a kinetic function that is similar to κ, i.e., an implicitly existentially quantified variable that is subject to the similarity constraint $\kappa^{(i)} \sim \kappa$.

A solution of an arithmetic constraint ϕ with n variables can be identified with a tuple in $\mathbb{R}_+^n$ since we assumed a total order on the variables. The solution set $sol(\phi)$ of a formula ϕ satisfies $sol(\phi) \subseteq \mathbb{R}_+^n$. For a reaction network N, the steady state equations are an arithmetic constraint ϕ_N such that $sol(\phi_N) = R_N$.

A difference constraint is a conjunctive logic formula with existential quantifiers with the following abstract syntax, where x'es are variables and $\delta \in \Delta$:

difference relation	$t ::= x \mid \delta$
set of difference relations	$s ::= \{t_1, \ldots, t_n\} \mid s + s' \mid s \cdot s'$
	$\mid \; Inh(s) \mid Acc(s) \mid \kappa(s_1, \ldots, s_k)$
difference constraints	$\psi ::= t \in s \mid t{=}t' \mid \psi \wedge \psi' \mid \exists x.\psi$

In contrast to before, all arithmetic operations now return sets of values in difference constraints. Difference constraints are interpreted over Δ, so that variables x are assigned to elements of Δ (rather than elements of $\mathbb{R}_+$). Arithmetic functions such as $+$ are interpreted as set-valued functions on Δ such as $+^{\Delta}$, and similarly a kinetic function κ is interpreted as the set valued function κ^{Δ}.

$$
\begin{aligned}
v_{\mathrm{r}_5} &\in v_{\mathrm{r}_{5'}} + y_{\mathrm{Leu}} \\
y_{\mathrm{Leu}} &= z_{\mathrm{Leu}} \\
v_{\mathrm{r}_4} &\in z_{\mathrm{CodY}} \cdot Inh(z_{\mathrm{CcpA}}) \\
v_{\mathrm{r}_4} &= z_{\mathrm{BS}_{\mathrm{CodY}}} \\
v_{\mathrm{r}_5} &\in z_{\mathrm{CcpA}} \cdot Inh(z_{\mathrm{BS}_{\mathrm{CodY}}} + z_{\mathrm{Leu}} + z_{\mathrm{BS}_{\mathrm{TnrA}}})
\end{aligned}
\qquad
\begin{aligned}
v_{\mathrm{r}_{5'}} &= z_{\mathrm{PIlv-Leu}} \\
v_{\mathrm{r}_6} &== \!\dot{} \qquad\qquad y_{\mathrm{Leu}} = z_{\mathrm{PIlv-Leu}} \\
v_{\mathrm{r}_6} &= z_{\mathrm{CcpA}} \qquad\quad x_{\mathrm{CodY}} = z_{\mathrm{CodY}} \\
v_{\mathrm{r}_7} &= z_{\mathrm{TnrA}} \qquad\quad x_{\mathrm{TnrA}} = z_{\mathrm{TnrA}} \\
v_{\mathrm{r}_7} &= z_{\mathrm{BS}_{\mathrm{TnrA}}}
\end{aligned}
$$

Fig. 6: Difference constraints for the PIlv-Leu network.

Since variables are totally ordered, a solution of a difference constraint can be identified with a tuple in Δ^n, so that the solution set $sol(\psi)$ of any difference constraint ψ satisfied $sol(\psi) \subseteq \Delta^n$.

We can now abstract from arithmetic constraints by interpreting them as difference constraints:

$$
\begin{aligned}
[\![x{=}\kappa^{(i)}(x_1,\dots,x_k)]\!] &= x \in \kappa(x_1,\dots,x_k) \\
[\![x = x_1 + x_2]\!] &= x \in x_1 + x_2 & [\![x_1{=}x_2]\!] &= (x_1{=}x_2) \\
[\![\phi \wedge \phi']\!] &= [\![\phi]\!] \wedge [\![\phi']\!] & [\![\exists x.\phi]\!] &= \exists x.[\![\phi]\!]
\end{aligned}
$$

An important point here is that the variables $\kappa^{(i)}$ for the partially known kinetic functions are replaced by well-known kinetic functions κ. For instance, we can abstract $x = ma^{(i)}(subs{:}x_1)$ to $x = x_1$, $x = ma^{(i)}(subs{:}x_1, subs{:}x_2)$ to $x \in x_1 \cdot x_2$, $x = exp^{(i)}()$ to $x = \dot{=}$, and $x = exp^{(i)}(subs{:}x_1, inh{:}x_2, inh{:}x_3)$ to $x \in x_1 \cdot Inh(x_2 + x_3)$. This way, the simplified steady state equations for the PIlv$-$Leu network are abstracted to the difference constraints in Fig. 6.

Theorem 1 (Soundness of Abstract Interpretation). $sol(\phi)^{\Delta} \subseteq sol([\![\phi]\!])$.

This theorem shows for any reaction network N that the solution set of the abstract interpretation $[\![\phi_N]\!]$ is a correct over-approximation of the abstraction of the exchange relation $R_N{}^{\Delta}$:

Corollary 1. $(R_N)^{\Delta} \subseteq sol([\![\phi_N]\!])$.

Proof. This follows immediately from Theorem 1, since $R_N = sol(\phi_N)$ by construction of ϕ_N.

7 Qualitative Reasoning with Difference Constraints

Since difference constraints have finite domains, we can compute all solutions of difference constraints by using finite domain constraint programming. Or else we can use constraint simplification for qualitative reasoning.

For instance, we can reconsider the question, which changes of the influxes of the PIlv$-$Leu network may increase the outflux of Leu. To find the answers, we can ask for the top-n solutions of the difference constraint $y_{\mathrm{Leu}} = <$ in conjunction with difference constraint inferred for the PIlv$-$Leu network in Fig. 6.

$$
\begin{array}{lll}
(\text{no}_1) & Inh(\doteq) \Rightarrow \doteq & (\text{no}_3) \quad t \cdot \doteq \Rightarrow t \\
\end{array}
$$

(no$_1$) $Inh(\doteq) \Rightarrow \doteq$ (no$_3$) $t \cdot \doteq \Rightarrow t$ (ip) $x + x \Rightarrow x$

(no$_2$) $Acc(\doteq) \Rightarrow \doteq$ (no$_4$) $\doteq \cdot t \Rightarrow t$ (si) $t \in t' \Rightarrow t = t'$

(bv) $\exists x.\ (x = t \land \psi) \Rightarrow \psi[t/x]$ (inh) $t \in Inh(t + s) \Rightarrow t \in Inh(s)$

Fig. 7: Simplification rules.

These solutions can be computed by the solver for difference constraints from [5], but extended with functions Inh and Acc in difference constraints.

There are only 5 solutions for this difference constraint after projection to in- and outflux variables. These solutions are given to the right. The top-2 solutions with the fewest changes (1. and 2.) show that one can either decrease the influx of CodY or TnrA. The next three solutions show that one more change does not change the matter.

	x_{CodY}	x_{TnrA}	y_{Leu}
1.	$>$	$\doteq$	$<$
2.	$\doteq$	$>$	$<$
3.	$>$	$>$	$<$
4.	$>$	$<$	$<$
5.	$<$	$>$	$<$

Since the PIlv−Leu network is quite simple, one can obtain the same predictions based on constraint simplification. The simplification of the difference constraints in Fig. 6 based on the rewrite rules in Fig. 7 yields:

$$y_{\text{Leu}} \in Inh(x_{\text{CodY}} + x_{\text{TnrA}}).$$

When assuming $y_{\text{Leu}} = <$ in addition we can simplify the constraint further to: $< \in Inh(x_{\text{CodY}} + x_{\text{TnrA}})$ which is equivalent to $x_{\text{CodY}} = > \lor x_{\text{TnrA}} = >$. This can be satisfied by decreasing the influx of either CodY or TnrA. Thus, we obtain the same result as before.

In Fig. 7 we present simplification rules for difference constraints over the specific domain $\Delta = \{<, >, \doteq\}$. Rule (bv) replaces equal by equal while eliminating existentially bound variables (all variables z_{A} and $v_{\text{r}_{\text{i}}}$ are implicitly existentially quantified). The simplification rules (no$_i$) remove the nochange value $\doteq$. The third rule (si) simplifies membership in singletons to equality. Rule (ip) expresses the idempotence of addition.

8 Conclusion

We have presented a formal modeling language for chemical reaction networks with partial kinetic information, and shown how to abstract away from the unknowns thanks abstract interpretation. We have illustrated that this allows us to reason qualitatively about such networks at the example of influx-change prediction. The same reasoning techniques are lifted to predict gene knockout strategies in follow-up work [2]. An important question for future work is how to develop finer abstractions for quantitative predictions.

References

1. Brinsmade, S.R., Kleijn, R.J., Sauer, U., Sonenshein, A.L.: Regulation of CodY Activity through Modulation of Intracellular Branched-Chain Amino Acid Pools. J. Bacteriol. 192(24), 6357–6368, 2010.
2. Coutte, F., Niehren, J., Dhali, D., John, M., Versari, C., Jacques, P.: Modeling Leucine's Metabolic Pathway and Knockout Prediction Improving the Production of Surfactin, a Biosurfactant from Bacilus Subtilis. J. Biotechnology. To appear.
3. Forbus, K.D.: Qualitative reasoning. In: Tucker, A.B. (ed.) The Computer Science and Engineering Handbook, pp. 715–733. CRC Press, 1997.
4. Grandoni, J.A., Zahler, S.A., Calvo, J.M.: Transcriptional regulation of the ilv-leu operon of Bacillus subtilis. Journal of bacteriology 174(10), 3212–3219, 1992.
5. John, M., Nebut, M., Niehren, J.: Knockout Prediction for Reaction Networks with Partial Kinetic Information. In: 14th International Conference on Verification, Model Checking, and Abstract Interpretation. LNCS, 338-357, 2013.
6. Jungreuthmayer, C., Zanghellini, J.: Designing optimal cell factories: integer programming couples elementary mode analysis with regulation. BMC systems biology 6(1), 103, 2012.
7. Mäder, U., Hennig, S., Hecker, M., Homuth, G.: Transcriptional organization and posttranscriptional regulation of the Bacillus subtilis branched-chain amino acid biosynthesis genes. Journal of bacteriology 186(8), 2240–2252, 2004.
8. Mäder, U., Schmeisky, A.G., Flórez, L.A., Stülke, J.: Subtiwiki — a comprehensive community resource for the model organism bacillus subtilis. Nucleic acids research: 40(D1). 278-287, 2012.
9. Molle, V., Nakaura, Y., Shivers, R.P., Yamaguchi, H., Losick, R., Fujita, Y., Sonenshein, A.L.: Additional targets of the Bacillus subtilis global regulator CodY identified by chromatin immunoprecipitation and genome-wide transcript analysis. Journal of bacteriology 185(6), 1911–1922, Mar 2003.
10. Orth, J.D., Thiele, I., Palsson, B.O.: What is flux balance analysis? Nature biotechnology 28(3), 245–248, 2010.
11. Otero, J.M., Nielsen, J.: Industrial systems biology. Industrial Biotechnology: Sustainable Growth and Economic Success, 2010.
12. Papin, J.A., Stelling, J., Price, N.D., Klamt, S., Schuster, S., Palsson, B.O.: Comparison of network-based pathway analysis methods. Trends in biotechnology 22(8), 400–405, 2004.
13. Price, N.D., Reed, J.L., Palsson, B.O.: Genome-scale models of microbial cells: evaluating the consequences of constraints. Nature reviews. Microbiology 2(11), 886–897, 2004.
14. Shivers, R.P., Sonenshein, A.L.: Activation of the Bacillus subtilis global regulator CodY by direct interaction with branched-chain amino acids. Molecular microbiology 53(2), 599–611, 2004.
15. Sohn, S.B., Kim, T.Y., Park, J.M., Lee, S.Y.: In silico genome-scale metabolic analysis of *Pseudomonas putida* kt2440 for polyhydroxyalkanoate synthesis, degradation of aromatics and anaerobic survival. Biotechnol. J. 5(7), 739–750. 2010.
16. Tojo, S., Satomura, T., Morisaki, K., Deutscher, J., Hirooka, K., Fujita, Y.: Elaborate transcription regulation of the Bacillus subtilis ilv-leu operon involved in the biosynthesis of branched-chain amino acids through global regulators of CcpA, CodY and TnrA. Molecular Microbiology 56(6), 1560–1573, 2005.
17. Villapakkam, A.C., Handke, L.D., Belitsky, B.R., Levdikov, V.M., Wilkinson, A.J., Sonenshein, A.L.: Genetic and Biochemical Analysis of the Interaction of B. subtilis CodY with Branched-Chain Amino Acids. J. Bacteriol. 191(22), 6865–6876, 2009.