A Transport Reaction Language: Preliminary Report

Thibaut Henin
IRISA
Campus de Beaulieu
F-35042 Rennes Cedex

Torsten Schaub
Universität Potsdam
August-Bebel-Str. 89
D-14482 Potsdam

Abstract

We present a qualitative approach to represent and reason upon transport reactions in metabolic networks. Our approach is built on action description languages for representing the dynamics of metabolic networks. To begin with, we introduce the transport reaction language $\mathcal{T}$ that is a customized sub-language of action language $\mathcal{C}$. We illustrate the modelling capacities by authentic biological examples. Moreover, we describe the system architecture of our system and detail its current major application.

1 Introduction

Molecular biology has seen a technological revolution with the establishment of high-throughput methods over the last years. This has resulted in a rapid growth of biological knowledge, gathered in web databases such as KEGG[12], Biomodels [14], Reactome [11], or MetaCyc[5]. Although the diverse biological networks are expressed in a computer-readable format, namely the *System Biology Markup Language* (SBML), the encompassing knowledge bases do not provide any general form of reasoning or query-answering.

We address this problem by proposing a reaction description language whose domain descriptions can be generated from networks expressed in SBML. A particular feature of our language is that it takes the location of molecules into account and thus allows for describing the transport of species through compartments. To provide the aforementioned reasoning capacities in a well understood framework, we embed our language into action language $\mathcal{C}$. Apart from clear semantic underpinnings, this approach allows us to benefit from the high performance of modern Answer Set Programming (ASP; [1]) systems via well known mappings of $\mathcal{C}$ into ASP. We illustrate the modelling capacities of our language by authentic biological examples. Moreover, we describe the system architecture of our system and detail its current major application.

2 Background

Action languages use *fluents* to describe the states of a dynamical system and *actions* influence the values of fluents. In $\mathcal{C}$, *static laws* describe properties between fluents that need to be satisfied in every state of the system. *Dynamic laws* describe the effects of actions, that is, how the system evolves when actions are executed.

More formally, we consider *action language* $\mathcal{C}$ [9] over a Boolean *action signature* $\langle B, F, A\rangle$, where B is the set $\{f, t\}$ of truth values, F is a set of *fluent names*, and A is a set of *action names*. In $\mathcal{C}$, an *action description* $D_{\mathcal{C}}$ over a signature $\langle B, F, A\rangle$ consists of *static* and *dynamic laws*:

$$(\mathbf{caused}\ \varphi\ \mathbf{if}\ \psi) \tag{1}$$

$$(\mathbf{caused}\ \varphi\ \mathbf{if}\ \psi\ \mathbf{after}\ \omega) \tag{2}$$

where φ and ψ are propositional combinations of fluent names and ω is a propositional combination of fluent and action names. Every action description $D_{\mathcal{C}}$ induces a unique transition system $\mathcal{T}_{\mathcal{C}}(D_{\mathcal{C}}) = \langle S, V, R\rangle$, where S is a set of *states*, V is a function determining fluents values in state s, and R is a relation containing all possible transitions between states. A *trajectory* $s_0, A_1, s_1, \ldots, s_{n-1}, A_n, s_n$ in a transition system $\langle S, V, R\rangle$ is a sequence of sets of actions $A_i \subseteq A$ and states $s_i \in S$ where $(s_{i-1}, A_i, s_i) \in R$ for $0 \leq i \leq n$. Intuitively, a trajectory represents one possible history (or simply path) within a transition system.

In [9], several syntactic extensions are defined. For instance, the rule $(\omega\ \mathbf{may\ cause}\ \varphi\ \mathbf{if}\ \psi)$ is a shorthand for $(\mathbf{caused}\ \varphi\ \mathbf{if}\ \varphi\ \mathbf{after}\ \psi \wedge \omega)$. Similarly, $(\mathbf{inertial}\ \varphi)$ is a shorthand for $(\mathbf{caused}\ \varphi\ \mathbf{if}\ \varphi\ \mathbf{after}\ \varphi)$. We refer to [9] for more detailed definitions.

$\mathcal{C}$ has an associated *query language*, $\mathcal{Q}$ [9], defined in terms of *axioms* and *queries*. The semantics of a query language is defined in terms of trajectories. The language $\mathcal{Q}$ defines two types of propositions.

- A proposition of form $(A\ \mathbf{occurs\ at}\ t_i)$ is satisfied by a trajectory $s_0, A_1, s_1, \ldots, s_{n-1}, A_n, s_n$, if $A \in A_{i+1}$, where A is an elementary action name and $i < n$.
- A proposition of form $(F\ \mathbf{holds\ at}\ t_i)$ is satisfied by a trajectory $s_0, A_1, s_1, \ldots, s_{n-1}, A_n, s_n$, if $s_i \models F$, where F a fluent name and $i \leq n$.

An axiom is a proposition possibly preceded by $\neg$. A query is a propositional combination of propositions. A query Q is a consequence of a set Φ of axioms, written $\Phi \models_{\mathcal{T}} Q$, if every trajectory of a transition system $\mathcal{T}$ satisfying all axioms in Φ also satisfies Q.

3 Language $\mathcal{T}$

A *reaction* is a process transforming some species, called *reactants* of the reaction, into some other species called *products* of the reaction. In our setting, a *species* is a molecule associated with a compartment, indicating the presence of the molecule in the compartment. Often reactions rely on species, called *enzymes* of the reaction whose presence is mandatory although they are not subject to any transformations. When a reaction deals with species in different compartments, the reaction is called a *transport*.

Language $\mathcal{T}$. We begin with three disjoints nonempty sets of symbols, viz. *molecule names* M, *compartment names* C, and *reaction names* R.

Our language $\mathcal{T}$ for specifying biological *transport networks* consists of the following expressions:

$$m \textbf{ in } c \tag{3}$$
$$r \textbf{ consumes } s_1, \ldots, s_n \tag{4}$$
$$r \textbf{ produces } s_1, \ldots, s_n \tag{5}$$
$$r \textbf{ needs } s_1, \ldots, s_n \tag{6}$$
$$r_1 \textbf{ overtakes } r_2 \tag{7}$$

A *species* is written as in (3) where $m \in M$ is a molecule and $c \in C$ is a compartment name. In the remainder, $r \in R$ is a reaction name and s_i are species for $1 \leq i \leq n$.

A *consume proposition* is an expression of form (4). This means that reaction r needs the presence of reactants $s_1, \ldots, s_n$ and consumes them during its process.

A *produce proposition* is an expression of form (5), meaning that reaction r produces the species $s_1, \ldots, s_n$.

A *need proposition* is an expression of form (6). This means that reaction r requires the presence of enzymes $s_1, \ldots, s_n$ without consuming them.

An *overtake proposition* is an expression of form (7), meaning that reaction r_1 takes priority over reaction r_2.

Example 1 (Detoxification) *Consider a simplistic metabolic pathway dealing with a cell facing hydrogen peroxide (H_2O_2) [16]. The oxidizing capacity of hydrogen peroxide is so strong that the chemical is considered a highly reactive oxygen species, corroding many materials, including human skin as well as DNA, RNA, and proteins.*

We distinguish two compartments, cytosol *and* outside*, and four molecules, viz. hydrogen peroxide H_2O_2 along with the product of detoxification, viz. H_2O,* catalase[1] *(written CAT), and some critical resource (abstracted as the generic molecule $R - OOH$). The presence of H_2O_2 implies some response of the cell, depending on its state:*

1. *If catalase has been secreted and becomes available outside the cell, H_2O_2 is transformed in H_2O extracellularly.*
2. *If not, H_2O_2 enters the cell and if catalase is present, H_2O_2 is transformed in H_2O in the cytosol.*
3. *If H_2O_2 enters the cell and if catalase is not present, then H_2O_2 reacts with $R - OOH$, which is then degraded into $R - OH$, and the cell dies.*

This network contains two transport reactions. One to enter the H_2O_2 in the cytosol (r_1) and one to export the enzyme (r_2). In $\mathcal{T}$, those two reactions are written as follows:

$$r_1 \textbf{ consumes } H_2O_2 \textbf{ in } outside$$
$$r_1 \textbf{ produces } H_2O_2 \textbf{ in } cytosol$$
$$r_2 \textbf{ consumes } CAT \textbf{ in } cytosol$$
$$r_2 \textbf{ produces } CAT \textbf{ in } outside$$

Our network contains also some reactions to transform H_2O_2. Catalase *is involved in two reactions transforming H_2O_2; one outside the cell (r_3) and another in the cytosol (r_4). Finally, there is a reaction transforming hydrogen peroxide into its inactive variant within the cytosol, but consuming the critical resource (r_5).*

$$r_3 \textbf{ consumes } H_2O_2 \textbf{ in } outside$$
$$r_3 \textbf{ produces } H_2O \textbf{ in } outside$$
$$r_3 \textbf{ needs } CAT \textbf{ in } outside$$
$$r_4 \textbf{ consumes } H_2O_2 \textbf{ in } cytosol$$
$$r_4 \textbf{ produces } H_2O \textbf{ in } cytosol$$
$$r_4 \textbf{ needs } CAT \textbf{ in } cytosol$$
$$r_5 \textbf{ consumes } H_2O_2 \textbf{ in } cytosol,\ R - OH \textbf{ in } cytosol$$
$$r_5 \textbf{ produces } R - OOH \textbf{ in } cytosol$$

Furthermore, this pathway contains one priority relation between r_4 and r_5. Indeed, whenever catalase is present in the cell, it transforms H_2O_2 into water before H_2O_2 can react with $R - OOH$. Usually, this last reaction never occurs because of the presence of catalase. However, in some cases, catalase is not fast enough and the cell dies (generally the concentration of H_2O_2 becomes too high).

In $\mathcal{T}$, this relation is expressed as follows:

$$r_4 \textbf{ overtakes } r_5$$

Example 2 (Glycolysis) *Glycolysis is a metabolic pathway converting glucose into pyruvate, while generating high energy compounds (ATP and NADH). In the early stage, glucose is imported inside the cytosol. Then, a series of reactions transform glucose into pyruvate molecules. Pyruvate is then transformed into ethanol which can be transported outside the cell. In the middle of the pathway, an intermediary compound (dihydroxyacetone-phosphate) can be used and lead to the formation of glycerol. This is detailed in Figure 1.*

As the complete glycolysis pathway contains too many reactions, we focus on two reactions that allow for illustrating two kind of reactions. The first reaction of this pathway is a transport reaction which takes some glucose outside the cell and imports it inside the cell. In $\mathcal{T}$, this reaction is written as follows:

$$r_6 \textbf{ consumes } glucose \textbf{ in } outside$$
$$r_6 \textbf{ produces } glucose \textbf{ in } cell$$

[1] Enzyme found in most plant and animal cells that functions as an oxidative catalyst; decomposes hydrogen peroxide into oxygen and water.

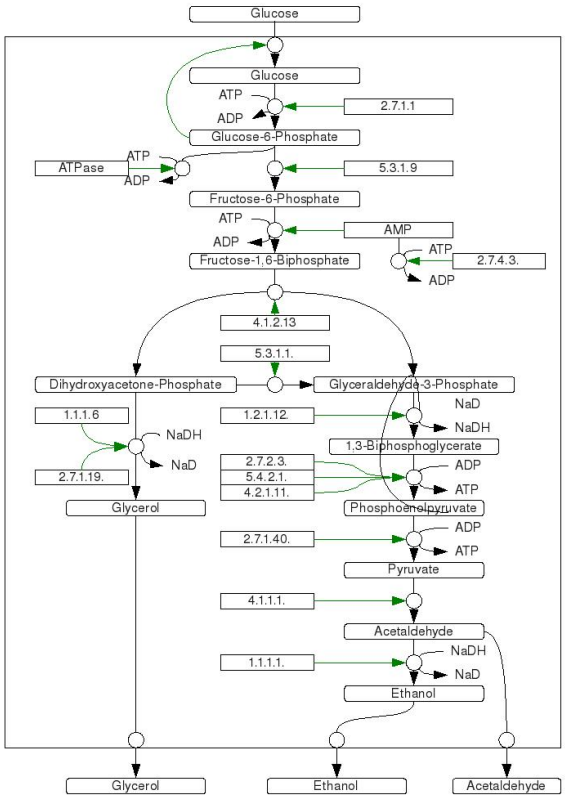

Figure 1: Map of the glycolysis

Another reaction of interest is the transformation of 1,3-Biphosphoglycerate into phosphoenolpyruvate because it needs three enzymes. In $\mathcal{T}$, this reaction is written as follows:

$$
\begin{array}{lll}
r_7 & \textbf{consumes} & 1,3-Biphosphoglycerate \textbf{ in } cell, \\
 & & ADP \textbf{ in } cell \\
r_7 & \textbf{produces} & phosphoenolpyruvate \textbf{ in } cell, \\
 & & ATP \textbf{ in } cell \\
r_7 & \textbf{needs} & 2.7.2.3 \textbf{ in } cell, \\
 & & 5.4.2.1 \textbf{ in } cell, \\
 & & 4.2.1.11 \textbf{ in } cell
\end{array}
$$

Translation into $\mathcal{C}$. The meaning of a transport network described in $\mathcal{T}$ is fixed through a translation into action language $\mathcal{C}$. Apart from providing a formal semantics, this allows us to draw upon the greatly elaborated framework of $\mathcal{C}$, offering query and observation languages, extensions such as additive fluents, needed for expressing resources, and finally an efficient implementation through off-the-shelf $\mathcal{C}$ solvers.

To this end, we map a transport network $N_{\mathcal{T}}$ in $\mathcal{T}$ over signature (M, C, R) into a (definite) action description $\mathbf{T}(N_{\mathcal{T}})$ in $\mathcal{C}$ over a Boolean signature (B, F, A), where $B = \{t, f\}$, $A = R$ and $F = \{present(m, c) \mid (m, c) \in M \times C\} \cup \{\bot\} \cup \{possible(r) \mid r \in R\}$ where $\bot$ is interpreted by f. In what follows, we detail our translation $\mathbf{T}$ by giving the translation of each expression in $\mathcal{T}$ into propositions of $\mathcal{C}$.

For each reactant $(m \textbf{ in } c)$ occurring in a consume proposition as in (4), we define one dynamic law.

$$\textbf{caused } \neg present(m, c) \textbf{ if } \neg present(m, c) \textbf{ after } r \qquad (8)$$

It expresses that the reactant may but must not be consumed.

For each product $(m \textbf{ in } c)$ in a product proposition of form (5), we define a dynamic law expressing that the species is produced by the reaction.

$$\textbf{caused } present(m, c) \textbf{ after } r \qquad (9)$$

For each enzyme $(e_i \textbf{ in } c_i)$ in a need proposition in (6) of each reaction, plus each reactant $(r_j \textbf{ in } c_j)$ in a consume proposition as in (4) of the same reaction we define a static law and a dynamic law to express that those species are mandatory for the reaction.

$$possible(r) \textbf{ if } \bigwedge_i present(e_i, c_i) \bigwedge_j present(r_j, c_j) \qquad (10)$$

$$\textbf{caused } \bot \textbf{ after } r \wedge \neg possible(r) \qquad (11)$$

For each overtakes expression of the form (7), we define a static law expressing that the less prior reaction cannot occur alone when both reaction can occur.

$$\textbf{caused } \bot \textbf{ after } r_2 \wedge \neg r_1 \wedge possible(r_1) \qquad (12)$$

Action query language $\mathcal{Q}_{\mathcal{T}}$ We adapt and extend action query language $\mathcal{Q}$ in order to use it with action language $\mathcal{T}$ by defining a new axiom dealing with species instead of fluents and another one providing confidence levels. As with $\mathcal{Q}$, the semantics of query language $\mathcal{Q}_{\mathcal{T}}$ is given in terms of trajectories.

$\mathcal{Q}_{\mathcal{T}}$ is build from three types of propositions:

- A proposition of form

$$(m \textbf{ in } c \textbf{ is present at } t_i) \qquad (13)$$

  is satisfied by a trajectory $s_0, A_1, s_1, \ldots, s_{n-1}, A_n, s_n$ if $present(m, c) \in s_i$ where $(m \textbf{ in } c)$ is a species and $i \leq n$.

- A proposition of form

$$(r \textbf{ occurs at } t_i) \qquad (14)$$

  is satisfied by a trajectory $s_0, A_1, s_1, \ldots, s_{n-1}, A_n, s_n$ if $r \in A_{i+1}$ where r is a reaction name and $i < n$.

- A proposition of form

$$(r \textbf{ has confidence level } l) \qquad (15)$$

  where r is a reaction name and l is an integer. Every trajectory satisfies this proposition. If no confidence level is given for a reaction, we give it the highest among all reactions. We denote by $L(r)$ the confidence level attributed to r. The confidence level of a trajectory $T = s_0, A_1, s_1, \ldots, s_{n-1}, A_n, s_n$, written $L(T)$, is defined as the minimal confidence level of the reaction in T. In symbols:

$$L(T) = \min\{L(r) \mid r \in A_i, 1 \leq i \leq n\}$$

An axiom is a proposition of form (13) or (14) possibly preceded by $\neg$. A query is a propositional combination of propositions of form (13) or (14).

Confidence propositions induce an order on trajectories via the confidence, $L(T)$, associated with each trajectory, T, in a given transition system. Given two trajectories T and T', we say T' is *more confident* than T, written $T \leq T'$, if

$$L(T) \leq L(T')$$

A query Q is a *confident consequence* of a set Φ of axioms, written $\Phi \models^c_{\mathcal{T}} Q$, if every $\leq$-maximal trajectory of a transition system $\mathcal{T}$ satisfying all axioms in Φ also satisfies Q.

4 Toolbox

We have implemented our approach as a modular toolbox, comprising the off-the-shelf ASP grounder *gringo*[2] and ASP solver *clasp*[2] as reasoning engines.

The initial input of the system is a set of metabolic pathway given in SBML (or directly in our language $\mathcal{T}$). SBML files are usually downloaded from internet databases, like Biomodels [14] or Reactome [11]. To begin with, a pathway expressed in SBML is translated into our reaction language $\mathcal{T}$ (using the libsbml library [4]). A pathway in $\mathcal{T}$ can then be queried via query language $\mathcal{Q}_{\mathcal{T}}$. To this end, both the query and the pathway are translated into logic programs and given to the ASP grounder and solver, respectively. The resulting answer sets represent maximal trajectories satisfying the axioms and the query.

For displaying trajectories, we developed a parser to translate the answer sets into the language *dot* [7], providing an easy and human readable way to specify graphs and being readable by various software packages. Among them, we have chose *rtgraph3d* [3] to display trajectories in a 3-dimensional view. When clicking on a state, the set of species present in the state is displayed (as shown in Figure 2).

For the sake of readability, the states of the system are displayed as spheres and colored depending upon whether they are initial (in red), final (in blue), both (in turquoise) or none (in black). The transitions are displayed as links between these spheres.

Finally, as our approach produces many trajectories which can be equivalent from a biologist's point of view, we moreover provide the following additional features:

- minimize the set of species present in the initial state (in order to remove species being irrelevant to the query);
- fix the maximum number of reactions in each transition (to avoid displaying many equivalent paths);
- fix whether or not the reactant may disappear (instead of the non-deterministic approach in (8));
- fix a set of relevant species to be shown in the answer set; the trajectories are then projected onto this set of relevant species, which provide a more abstract view.[3]

[2] `http://potassco.sourceforge.net`

[3] For the reader interested in ASP, we mention that this is accomplished with *clasp*'s projective enumeration, invoked with `--project`.

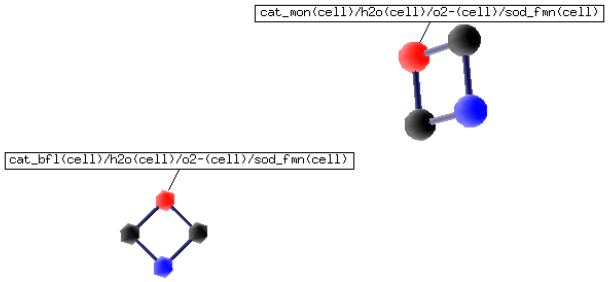

Figure 2: Detoxification pathway of *Sinorhizobium Meliloti 1021*.: Two initial situations (in red) can lead to the detoxification of O_2^-.

Example 3 (Detoxification of O_2^-) *We now complete the detoxification pathway given in Example 1 by taking the whole detoxification pathway of a real bacteria,* Sinorhizobium Meliloti 1021 *[16]. Instead of facing hydrogen peroxide, the cell faces superoxide (viz. O_2^-). Superoxide is so toxic that intracellular levels above 1nM are lethal. This compound is one of the main causes of oxidative stress.*

To survive, the cell must have a superoxide dismutases (SOD) to transform superoxide into hydrogen peroxide and then some catalase (CAT) to transform hydrogen peroxide into water and oxygen. This is the case with almost all aerobic organisms, and thus applies to Sinorhizobium Meliloti 1021 *which has multiple isoforms of these enzymes.*

In Figure 2, we give a screenshot of our toolbox for this detoxification problem. We have taken the full detoxification network given in OxyGen [16] and the confidence levels given by OxyGen. The axioms specify that initially the superoxide is present in the cell and no oxidant is present in the final state (and the query is empty). Here, we show all maximal trajectories (in term of confidence levels) and then minimize the number of enzymes present initially.[4] *Furthermore, we restrict the trajectories to contain only one reaction per transition.*

Sinorhizobium Meliloti 1021 *contains one SOD and two isoforms of CAT that have been experimentally proved. In fact, it contains further isoforms of these enzyme (but not experimentally proved) and other enzyme of less interest. Since the proved enzyme are sufficient, we refrained from using the other ones and thus obtain only two possibilities (as shown in the figure).*

In this example, the trajectories are of length 2. Also, we have minimized the number of initial species. Figure 2 shows two initial states (in red), one state per catalase. For each initial state, there are two trajectories, one where the superoxide disappears after the first transition and the other where it disappears after the second step.

Example 4 (Glycolysis) *Next, we further elaborate on the glycolysis pathway, already described in Example 2. Recall that glycolysis takes glucose as input and after a first series of reactions, glycolysis has the choice to produce glycerol or not. Both cases result in the production of ethanol.*

[4] This is done with `#maximize` and `#minimize` statements provided by ASP systems.

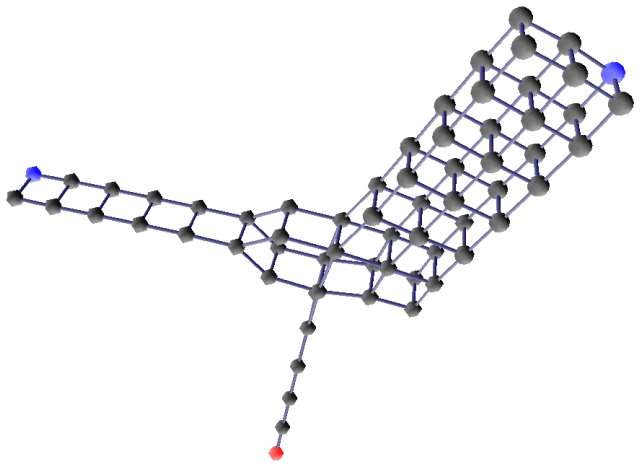

Figure 3: Glycolysis: The initial condition (where the glucose is outside, in red) can lead to two distinct cases (where glycerol is produced (left) or not (right)).

Figure 3 shows the output of our toolbox with this pathway. The axioms specify that initially glucose is present outside the cell along with each useful enzyme. The axioms also specify that ethanol is present outside the cell in the final state. Again, the query is empty.

For the sake of clarity, we have restricted trajectories to contain only one reaction per transition and every reactant of a reaction disappears after the corresponding reaction occurred. After a first series of reactions (in the bottom of Figure 3), the cell chooses to produce either only ethanol (in the left part) or to produce both ethanol and glycerol.

5 Application

We have used our method for identifying biological experiments in view of gathering new biological knowledge. As the networks are often incomplete or contain automatically generated reactions (which have not been proved), we aim at finding easy experiments that can prove parts of the network.

For this, we have used the whole detoxification networks of 655 bacteria given in OxyGen [16]. In OxyGen, each reaction (of each bacteria) is given a confidence level that expresses the accuracy of the annotation of the enzyme. There are three confidence levels in OxyGen:

- Enzymes that have been experimentally demonstrated, that is, found by comparison with a database of experimentally validated proteins (in this case, the confidence level is 3);
- Enzymes without biological evidence but for which the signature of the corresponding gene was found in the genome (in this case, the confidence level is 2);
- Enzymes from disrupted regions, like frameshifts[5] or pseudogenes[6] (in this case, the confidence level is 1).

[5]Two separate motifs of the same signature are found in two different frames of the same strand.

[6]One or two stops in frame.

Oxidant	3	2	1	0
O_2^-	4.7 %	86.5 %	0.3 % (2)	8.3 %
H_2O_2	12.8 %	84.8 %	0.1 % (1)	2.1 %
$R-OOH$	7.1 %	87.4 %	0 %	5.3 %
$ONOOH$	4.8 %	73.1 %	0.1 % (1)	21.8 %
NO	2.7 %	58.1 %	0 %	39.1 %

Table 1: For each oxidant, the bacteria are grouped by the confidence level needed to detoxify the oxidant.

We have used our method to find reactions of confidence level 1 which are mandatory to detoxify an arbitrary oxidant. Indeed, there is not so much knowledge gained in proving reactions of confidence level 2 because they are likely to exist. More interesting knowledge is obtained by asking whether reactions of level 1 exist. However, for those reaction, there often exist reactions of a higher confidence level that accomplish the same function. Proving them experimentally implies to knock-out a gene for all those reaction, or inhibit their effects and make the experiment more expensive. We thus search for reactions of level 1 which are mandatory, that is, there is no other reaction (or set of reactions) that accomplishes the same function.

To find them, we computed for each (network of each) bacteria and each oxidant, the maximal confidence level of the maximal trajectories that detoxifies the oxidant. This maximal level tells us the confidence level one can put on the detoxification of the oxidant for the given bacteria. For each oxidant, we have grouped the bacteria depending on this confidence level, and give this result in Table 1. For instance, for detoxifying H_2O_2, we noticed 12.8 % bacterias with confidence level 3, 84.8 % bacterias with confidence level 2, a single bacterium with confidence level 1, and 2.1 % bacterias that were unable to detoxify the oxidant at hand (relative to the knowledge comprised in OxyGen).

For a lot of bacteria, an enzyme of confidence level 2 is needed to detoxify the oxidant. For those bacteria, doing an experiment which shows that the detoxification occurs will only show that the annotation was accurate.

But we actually discovered four cases (two with O_2^-, one with H_2O_2, and one with $ONOOH$, indicated in parentheses in Table 1) where a reaction having a confidence level 1 is mandatory to detoxify an oxidant. For these bacteria, doing a simple experiment showing that the detoxification occurs will point out an unlikely yet existing reaction.

6 Related Work

Modeling methods for biological system fall into two categories, quantitative ones, focusing on measurable information, like concentration of molecules, and qualitative ones, focusing on the mere presence or absence of molecules.

The two major quantitative methods are Ordinary Differential Equations (ODEs) [6] and Flux Balance Analysis (FBA) [13]. In the former, the concentration of a metabolite is given by an ODE summarizing all reactions in which the metabolite is consumed or produced. FBA relies on the steady state assumption and models reactions by two matrices, one for stoichiometry and another for fluxes. In large scale net-

works, however, numerous parameters (in ODEs) and fluxes (in FBA) are unknown and thus estimated, so that despite the partly fine-grained input data the final results are prone to inaccuracy.

Among the qualitative approaches, we find petri nets. For modeling metabolic networks [15], compounds are modeled by places and quantities by tokens; reactions are transitions from reactants' places to products' places. Hybrid petri nets have been developed to use petri nets with differential equations. Along with petri nets, one can compute the "elementary modes", i.e. vectors of transition which forms the base of the petri net. Those vectors provide information on the topology of the network.

Biocham is a framework dedicated to biochemical reasoning [8]. Reactions are modeled by transformation rules which form a Kripke structure. Queries can be made in CTL and a symbolic model checker is used to solved them.

Action languages have already been used to model biological networks. Action language $\mathcal{A}$ [9] has been extended with triggers and inhibitions to model signaling networks [2]. This language is referred as $\mathcal{A}_T^0$ and has been further extended in [10] to form Language $\mathcal{C}_{TAID}$ featuring allowance statements and defaults.

In [17], the authors used an abductive logic programming system to revise metabolic pathways. Their method allows removing or adding new reactions, enzymes, or inhibition rules from a network and given observational data but they do not define any abstract language to express network and queries.

Between all those method, the Systems Biology Markup Language (SBML) is an attempt to model biological networks in a machine-readable format. It's applicable to models of metabolism, cell-signaling, and many others. Reactions are modeled as relations between chemical compound in compartments. Those relations can contain information on stoichiometry and differential equations. BioModels [14] is a database referencing all SBML models from the literature.

7 Discussion

We have presented a new action language dedicated to metabolic networks. This language handles chemical compounds in compartments and chemical reactions along with essential features as priorities and confidence level. We have applied our method to the detoxification pathway of 655 bacteria and found 4 reactions of interest.

With confidence level, our method facilitates the completion of networks. By giving a high confidence level to published reactions and a low one to non-validated reactions, our toolbox finds out essential non-validated reaction to accomplish some function. It shows where biological experiments can bring new knowledge.

As a next step, we plan to use more genericity in queries and compartments. The static time step in expression of our query language could be improved by the use of modal logic and allow more expressiveness. We also want to allow for the definition of types of compartments. Those types will allow for more general observations of the system.

Acknowledgements The second author was supported by the Federal Ministry of Education and Research within the GoFORSYS project (http://www.goforsys.org/; grant 0313924).

References

[1] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University, 2003.

[2] C. Baral, K. Chancellor, N. Tran, N. Tran, A. Joy, and M. Berens. A knowledge based approach for representing and reasoning about signaling networks. *Bioinformatics*, 20(1):15–22, 2004.

[3] P. Biondi. `http://www.secdev.org/projects/rtgraph3d/`.

[4] B. Bornstein, S. Keating, A. Jouraku, and M. Hucka. Libsbml: An API library for SBML. *Bioinformatics*, pages btn051+, 2008.

[5] R. Caspi, H. Foerster, C. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Rhee, A. Shearer, C. Tissier, T. Walk, P. Zhang, and P. Karp. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Res*, 2007.

[6] N. Duarte, M. Herrgård, and B. Palsson. Reconstruction and validation of saccharomyces cerevisiae ind750, a fully compartmentalized genome-scale metabolic model. *Genome Res*, 14(7):1298–1309, 2004.

[7] S. North, E. Koutsofios. Drawing graphs with dot. Technical report, At&T.

[8] F. Fages, S. Sollman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine biocham. *Journal of Biological Physics and Chemistry*, 4:64–73, 2004.

[9] M. Gelfond and V. Lifschitz. Action languages. *Electronic Transactions on AI*, 3(6):193–210, 1998.

[10] S. Dworschak, S. Grell, V. Nikiforova, T. Schaub, and J. Selbig. Modeling biological networks by action languages via answer set programming. *Constraints*, 13(1-2):21–65, 2008.

[11] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res*, 33, 2005.

[12] M. Kanehisa and S. Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[13] K. Kauffman, P. Prakash, and J. Edwards. Advances in flux balance analysis. *Curr Opin Biotechnol*, 14(5):491–496, 2003.

[14] N. Le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. Snoep, and M. Hucka. Biomodels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res*, 34, 2006.

[15] J. Pinney, D. Westhead, and G. McConkey. Petri net representations in systems biology. *Biochem Soc Trans*, 31(Pt 6):1513–1515, 2003.

[16] D. Thybert, S. Avner, C. Lucchetti-Miganeh, A. Chéron, and F. Barloy-Hubler. Oxygene: an innovative platform for investigating oxidative-response genes in whole prokaryotic genomes. *BMC Genomics*, 9:637+, 2008.

[17] Ray, O., Whelan, K., King, R.: A nonmonotonic logical approach for modelling and revising metabolic networks. In: Proceedings CISIS'09, IEEE Press (2009) To appear.