Towards a Logical Analysis of Biochemical Pathways

Patrick Doherty, Steve Kertes, Martin Magnusson, and Andrzej Szalas

Department of Computer and Information Science, SE-581 83 Linköping, Sweden,
email: {patdo,g-steke,marma,andsz}@ida.liu.se

Abstract. Biochemical pathways or networks are generic representations used to model many different types of complex functional and physical interactions in biological systems. Models based on experimental results are often incomplete, e.g., reactions may be missing and only some products are observed. In such cases, one would like to reason about incomplete network representations and propose candidate hypotheses, which when represented as additional reactions, substrates, products, would complete the network and provide causal explanations for the existing observations.
In this paper, we provide a logical model of biochemical pathways and show how abductive hypothesis generation may be used to provide additional information about incomplete pathways. Hypothesis generation is achieved using weakest and strongest necessary conditions which represent these incomplete biochemical pathways and explain observations about the functional and physical interactions being modeled. The techniques are demonstrated using metabolism and molecular synthesis examples.

Keywords: abduction, biochemical pathways, hypotheses generation, weakest sufficient and strongest necessary conditions

1 Introduction

Biochemical pathways or networks are generic representations used to model many different types of complex functional and physical interactions in biological systems. For example, metabolism can be viewed and modeled in terms of complex networks of chemical reactions catalyzed by enzymes and consisting of reactive chains of substrates and products (see, e.g., [1, 2]).

Often these models are incomplete. For example, reactions may be missing and only some products are observed. In such cases, one would like to reason about incomplete network representations and propose candidate hypotheses, which when represented as additional reactions, substrates, products, or constraints on such, would complete the network and provide causal explanations for the existing observations.

In this paper, we provide a logical model of biochemical pathways (Section 2) and show how abductive hypothesis generation may be used to provide additional information about incomplete pathways. Hypothesis generation is achieved using weakest and strongest necessary conditions for restricted fragments of 1st-order theories which represent these incomplete biochemical pathways and explain observations about the

functional and physical interactions being modeled (see Section 3). Quantifier elimination techniques (see [3–5]) are used to automatically generate these hypotheses, using the technique described in [6].

Part of the modeling process includes the use of approximate databases [7], where when queries are viewed as inferences, questions can be asked about the generated hypotheses (see Section 4).

These techniques are demonstrated in Section 5, using metabolism and molecular synthesis examples.

Comparing to other approaches, e.g., that presented in [8, 9], we focus on modeling reactions in the classical first-order logic and then on hypotheses generation, using approximations provided by strongest necessary and weakest sufficient conditions, and on evaluation of queries using approximate databases, where queries are tractable.

This paper is an extended version of extended abstract [10].

2 Data Models for the Analysis of Biochemical Pathways

2.1 Preliminaries

The analysis of biochemical pathways has been considered in numerous papers (see, e,g, [1, 2]). In this paper, a bipartite graph representation of chemical reactions will be used (see, e.g., [2]).

It is assumed that any reaction is specified by:

$$n : \mathrm{c}_1 + \ldots + \mathrm{c}_\mathrm{k} \xrightarrow{\alpha(\mathrm{n})} \mathrm{c}'_1 + \ldots + \mathrm{c}'_\mathrm{l},$$

where:

- n is a label (name) of the reaction
- $c_1, \ldots, c_k$ are reactants (inputs for n)
- $c'_1, \ldots, c'_l$ are products of n and $\alpha(n)$ is a formula that specifies additional conditions necessary for the reaction, such as temperature, pressure, presence of catalyzers, etc.

In a bipartite graph there are two types of nodes: *compound nodes* (depicted by circles) and *reaction nodes* (depicted by rectangles). An edge from a compound node to a reaction node denotes a substrate. An edge from a reaction node to a compound node denotes a product of the reaction. We additionally allow conditions placed in the boxes, if these are specified for particular reactions (see Figure 1).

2.2 Representing Reactions in Logic

The language In the paper, the classical first-order logic is used for specifying reactions.

Reaction nodes will be represented explicitly, while information about available compounds will be given via a suitable relation. Consequently, it is assumed that the following symbols are available:

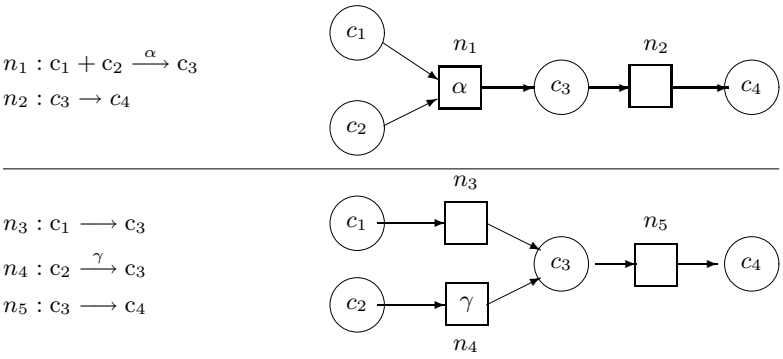

Fig. 1. Examples of bipartite graphs.

1. constants and variables:
   - constants representing (naming) reactions (denoted by n, n'), compounds (denoted by $c, c', h2o, co2$ etc.), and reaction nodes (denoted by r, r')
   - variables representing reactions, denoted by N, N', compounds (C, C'), and reaction nodes (R, R');

   for constants and variables we also use indices, when necessary
2. relation symbols reflecting static information:
   - $in(C, N)$ meaning that compound C is needed for reaction N
   - $out(N, C)$ meaning that compound C is a product of reaction N
3. relation symbols reflecting dynamic information:
   - $prec(R, R')$ meaning that reaction node R precedes reaction node R'
   - $notExcl(R, R')$ meaning that precedence $prec(R, R')$ is not excluded[1]
   - $chain(R, R')$ meaning that there is a chain of reactions $R, R_1, R_2, \ldots, R_k, R'$ such that $prec(R, R_1), prec(R_1, R_2), \ldots, prec(R_{k-1}, R_k), prec(R_k, R')$
   - $react(N, R)$ meaning that reaction N actually happened in reaction node R
   - $av(C, R)$ meaning that compound C is available for reaction represented by reaction node R.

Specifying Reactions in Logic Let e, t be any expressions and s any subexpression of e. By $e(s/t)$ we shall mean the expression obtained from e by substituting each occurrence of s by t.

It is assumed that any formula is implicitly universally quantified over all its variables that are not bound by a quantifier.

Any reaction of the form

$$n : \mathrm{c}_1 + \ldots + \mathrm{c}_\mathrm{k} \xrightarrow{\alpha(\mathrm{n})} \mathrm{c}'_1 + \ldots + \mathrm{c}'_\mathrm{l}, \tag{1}$$

is translated into the formula $react(n, R)$, where the following integrity constraints are assumed:

[1] $notExcl(R, R')$ is useful for constraining the hypothesis generation process.

– static information about reaction n:

$$in(c_1, n) \wedge \ldots \wedge in(c_k, n) \wedge$$
$$out(n, c'_1) \wedge \ldots \wedge out(n, c'_l)$$

– linking nodes in graphs with reactions

$$\begin{aligned} & react(n, R) \rightarrow \\ & \quad \alpha(n/R) \wedge \\ & \quad \forall C.\{in(C, n) \rightarrow av(C, R)\} \wedge \\ & \quad \forall R', C'.\{[prec(R, R') \wedge out(n, C')] \rightarrow av(C', R')\}. \end{aligned} \tag{2}$$

Observe that the above formula can be more efficiently expressed according to the schema:

$$\begin{aligned} & react(n, R) \rightarrow \\ & \quad \alpha(n/R) \wedge \\ & \quad av(c_1, R) \wedge \ldots \wedge av(c_k, R) \wedge \\ & \quad \forall R'.\{prec(R, R') \rightarrow av(c'_1, R')\} \wedge \\ & \quad \ldots \wedge \\ & \quad \forall R'.\{prec(R, R') \rightarrow av(c'_l, R')\}. \end{aligned} \tag{3}$$

However, in order to link to an existing biochemical database one would most probably have to use the previous version with suitable syntactic adjustments.

3 Strongest Necessary and Weakest Sufficient Conditions

3.1 Preliminaries

The strongest necessary and weakest sufficient conditions, as understood in this paper and defined below, have been introduced in [11] and further developed in [6]. Observe that the weakest sufficient condition corresponds to the weakest abduction expressed in terms of a chosen vocabulary.

Definition 3.1. *By* a necessary condition of a formula α on the set of relation symbols P under theory T *we shall understand any formula ϕ containing only symbols in P such that $T \models \alpha \rightarrow \phi$. It is the* strongest necessary condition, *denoted by* $\textsc{Snc}(\alpha; T; P)$ *if, additionally, for any necessary condition ψ of α on P under T, $T \models \phi \rightarrow \psi$ holds.* □

Definition 3.2. *By* a sufficient condition of a formula α on the set of relation symbols P under theory T *we shall understand any formula ϕ containing only symbols in P such that $T \models \phi \rightarrow \alpha$. It is the* weakest sufficient condition, *denoted by* $\textsc{Wsc}(\alpha; T; P)$ *if, additionally, for any sufficient condition ψ of α on P under T, $T \models \psi \rightarrow \phi$ holds.* □

The set P in the definitions for wsc's and snc's is referred to as the *target language*. The following lemma has been proven in [6].

Lemma 3.3. *For any formula α, any set of relation symbols P and theory T such that the set of free variables of T is disjoint with the set of free variables of α:*

$$\text{SNC}(\alpha; T; P) \equiv \exists \bar{\Phi}.\, [T \wedge \alpha] \tag{4}$$

$$\text{WSC}(\alpha; T; P) \equiv \forall \bar{\Phi}.\, [T \rightarrow \alpha], \tag{5}$$

where $\bar{\Phi}$ consists of all relation symbols appearing in T and α but not in P. □

The above characterizations are second-order. However, for a large class of formulas, one can obtain logically equivalent first-order formulas [2] (see, e.g., [3, 4]) or fixpoint formulas[3] (see, e.g., [5]) by applying techniques for eliminating second-order quantifiers. The algorithms given in [3, 4] are implemented and are available online. The algorithm based on [5] is implemented, as described in Section 4.

3.2 Hypotheses Generation Using Snc's and Wsc's

Snc's and Wsc's provide a powerful means of generating hypotheses using abduction. Suppose one is given a (incomplete) specification of a set of interacting reactions of the form shown in equation (1). We would use this set of formulas as the background theory T. Suppose additionally, that a number of observations are made referring to reactions known to have occurred, or compounds known to be available for participation in a reaction, etc. Let α denote the formula representing these observations. Generally, it will not be the case that $T \models \alpha$ because T only provides an incomplete specification of the reactions.

We would like to generate a formula (candidate hypotheses) ϕ in a restricted subset of the language of reactions P such that ϕ together with the background theory T does entail the observations α. It is important that we do not over commit otherwise we could just as easily choose α itself as the hypothesis which wouldn't do much good. In fact, the $\text{WSC}(\alpha; T; P)$ does just the right thing since we know that $T \wedge \text{WSC}(\alpha; T; P) \models \alpha$ and it is the weakest such formula by definition.

$\text{WSC}(\alpha; T; P)$ actually represents alternative hypotheses for explaining α. If it is put in disjunctive normal form, each of the disjuncts makes $\text{WSC}(\alpha; T; P)$ true and represents a weakest hypothesis. To reason about what each candidate hypothesis might imply in terms of completing the reaction representation, one would simply add both the background theory T and the candidate hypothesis α' to the approximate database described in Section 4 and query the database as desired.

Technically, if the wsc/sns in question can be expressed using the classical first-order logic,[4] one would then perform the following steps:

1. replace any quantifier $\forall X.\, \alpha(X)$ by the conjunction $\alpha(X/a_1) \wedge \ldots \wedge \alpha(X/a_k)$, where $a_1, \ldots, a_k$ are all constants occurring in the database of the type compatible with the type of X

[2] The class of formulas includes, e.g., all non-recursive semi-Horn formulas - see, e.g., [12].

[3] Fixpoint formulas are obtained for semi-Horn formulas, a formalism substantially more expressive than the Horn fragment of first-order logic.

[4] E.g., second-order quantifier elimination results in a first-order formula.

2. replace any quantifier $\exists X.\, \alpha(X)$ by the disjunction $\alpha(X/a_1) \vee \ldots \vee \alpha(X/a_k)$, where $a_1, \ldots, a_k$ are as above
3. transform the resulting formula into prenex normal for with the quantifier-free part transformed into the disjunctive normal form.

Observe that two first steps are of linear complexity wrt size of the database. However, the third step might be of exponential complexity wrt the size of the formula obtained in previous steps. One can, however, generate suitable disjuncts one at a time, not all of them at once.

Each disjunct is a conjunction of literals and represents a possible set of facts making a given snc or wsc true. It could then be used to update the database containing the background theory. One could then query database, using the language outlined in Section 4 in order to verify certain properties of the generated hypotheses (e.g., whether the obtained chain of reactions is cycle-free or is acceptable by experts).

In the case when the wsc and snc are fixpoint formulas, one can use the standard characterization of fixpoints as disjunctions/conjunctions of iterations of fixpoint operators. However, the complexity of the resulting formula might in this case be unacceptable.

3.3 Applications of Snc's and Wsc's in the Analysis of Biochemical Pathways

Observe that:

– wsc corresponds to a weakest abduction expressed in a given target language. For example, consider

$$\textsc{Wsc}(av(fin, so3); Th; L),$$

where Th is the theory expressing properties of given reactions, as constructed in Section 2.2. Then
- if the target language L consists of av only, the resulting wsc expresses what compounds availability makes the required output of reaction node fin feasible
- if the target language consists of $react$, then the resulting wsc expresses reactions necessary to make the required output of fin feasible

– snc allows to infer facts from a negative information. In fact, snc expresses what would be possible under a given set of hypotheses. For example, if a certain side product has not been observed, a reaction can be excluded from the set of hypotheses.

4 Approximate Databases

4.1 Introduction

An approximate database is based on the standard concept of a deductive database but with a number of modifications that enable reasoning about incomplete and approximate knowledge.

The basis for the representation of data are approximate relations. An approximate relation, say R, is given by:

– a *lower approximation*, denoted by R_+, containing objects known to satisfy the relation
– an *upper approximation*, denoted by $R_\oplus$, containing objects that may satisfy the relation.

We also allow in the language R_-, consisting of tuples which are known not to satisfy R, $R_\ominus$ consisting of tuples that may not satisfy R and $R_\pm$, called the *boundary of* R, consisting of tuples for which it is unknown whether they satisfy R. Of course, $R_\oplus = R_+ \cup R_\pm$ and $R_\ominus = R_- \cup R_\pm$.

Queries to the database are approximate formulas, where all relations are approximate. In the case of a query containing free variables, the result of the query is a list of substitutions for those variables that satisfy the query. In the case of a query without free variables, the result is true, false or unknown.

A prototype implementation of the approximate database management system that functions as a front-end to the POSTGRESQL database has been developed. The architecture of implemented layers is depicted in Figure 2.

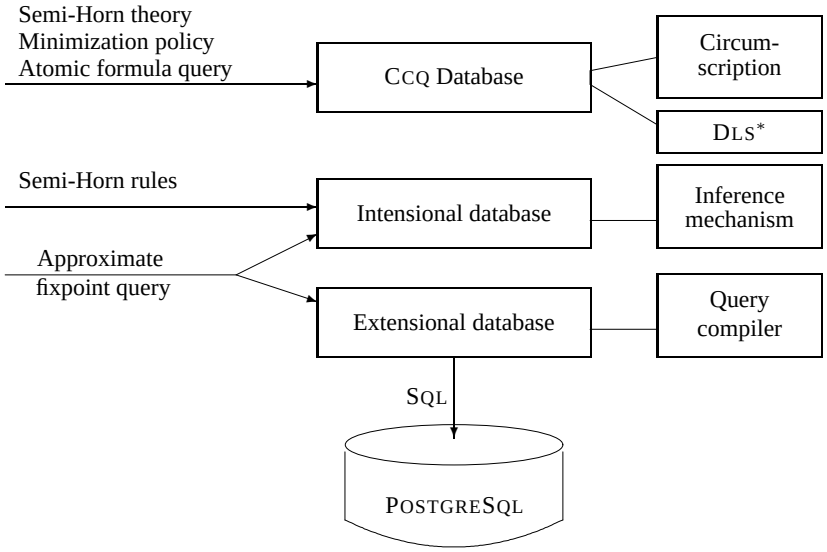

Fig. 2. The architecture of the knowledge database.

An extensional database layer (EDB) defines object constants and relations, including information about argument types, and explicitly stores positive (R_+) and negative (R_-) information for each relation R. The Boundaries of any approximate relation can be generated by combining its explicit information with all possible tuples of constants constrained by the argument types. A query compilation mechanism translates any approximate fixpoint formula into standard SQL and returns bindings of free variables using results from the POSTGRESQL database.

An intensional database layer (IDB) contains deductive rules in the form of approximate formula implications and infers new positive or negative information by applying the inference mechanism on data in the EDB.

Finally, a contextually closed query layer (the CCQ layer) provides functionality for local closed world assumptions by circumscribing a definition of the query in the context of a minimization/variation policy (see, [7]). The circumscription generates a second order formula, which can often be reduced into a logically equivalent fixpoint formula that can be evaluated by the EDB. This is achieved through application of the DLS* algorithm (described, e.g., in [13]).

4.2 Using Approximate Databases for Representing Biochemical Pathways

Bipartite graphs describing chemical reactions can be represented and reasoned about using queries to an approximate database with the language presented in Section 4.1. As is often the case, the reaction graph may be incomplete, in which case querying may be insufficient to draw any conclusions at all. In this case we propose the application of Wsc's and Snc's in order to hypothesize additional reaction pathways before querying.

According to the definitions of Wsc's and Snc's they are equivalent to second order formulas. Using the DLS* algorithm, which is part of the CCQ layer, these can often be reduced into equivalent first order of fixpoint formulas.

When a hypothesis has been generated, as described in Section 3.2, it can be added to the database together with the background theory and initial conditions regarding the availability of compounds. Any consequences of the hypothesis can then be investigated by querying the database.

For example one might want to find compounds that are available as a result of the reactions in the hypothesis by performing the query:

$$\exists N, R\,[react_+(N, R) \wedge out_+(N, C)].$$

The compound variable C is unbound and the query will return any compounds that resulted from any reaction that actually took place. If one of these compounds could not be found after performing the experiment, this might be a reason to exclude the hypothesis.

Consider now relations described in Section 2.2.

We assume that relations in and out are in the knowledge database or, alternatively, that the database contains definitions of reactions prepared according to schema (2) or (3) (see Section 2.2). In what follows we assume for simplicity that schema (3) is used whenever possible.

Relations $prec$, $chain$, $react$ and av are often known only partially on the basis of observations. In such a case, in order to find the missing information, we will project out those relation symbols using the methodology of approximations based on weakest sufficient and strongest necessary conditions (see Section 3). We will assume the following set of formulas, reflecting the partial knowledge:

– if it is known that $react(n, r)$ holds, where n and r are constants, and n is specified by (1), then one adds to the database formula

$$[N = n \wedge R = r] \rightarrow react_+(N, R)$$

– if $prec(r, r')$ is known, then one adds formula

$$[R = r \wedge R' = r'] \rightarrow prec_+(R, R'),$$

and similarly for $chain$

– if $av(c, r)$ is known, then one adds formula

$$[C = c \wedge R = r] \rightarrow av_+(C, R).$$

If some reaction nodes are missing then one has to make sure that database domain has some extra constants for such nodes. These constants can eventually be used in hypothesis generation (see Section 3.2).

One also adds the following formulas:

– sufficient condition for $prec$:

$$\exists C, N.\, [react_+(N, R) \wedge out_+(N, C) \wedge \\ av_+(C, R') \wedge notExcl_{\oplus}(R, R')] \rightarrow prec_+(R, R') \tag{6}$$

– necessary condition for $prec$

$$prec_+(R, R') \rightarrow \\ \exists C, N.\, [react_+(N, R) \wedge out_+(N, C) \wedge av_+(C, R')]. \tag{7}$$

Remark 4.1. All the formulas considered so far allow one to compute weakest sufficient and strongest necessary conditions as classical first order formulas, provided that additional conditions expressed by formula α in all reactions (see Equation (1)) are non-recursive. □

The following formulas are sometimes[5] required, but should be avoided whenever possible, since computing weakest sufficient and strongest necessary conditions in their presence results in fixpoint formulas which are much more complex to handle during the hypothesis generation stage.[6]

– definition of $chain$:

$$chain_+(R, R') \stackrel{\text{def}}{\equiv} \\ prec_+(R, R') \vee \\ \exists R_1.\, [prec_+(R, R_1) \wedge chain_+(R_1, R')].$$

– no cycles:
$\forall R.\, [\neg chain_+(R, R)]$ (or, alternatively, $\forall R.\, [chain_{\ominus}(R, R)]$).

One can also describe the availability of certain compounds and some other conditions. For example, $\forall R.\, [av_+(h2so4, R)]$ specifies that $H2SO4$ is known to be available for any reaction node.

[5] When the reasoning concerns chains of reactions of not bounded a priori length .

[6] In fact, from the point of view of complexity, it is better to use these formulas as integrity constraints to rule out certain hypotheses (see discussion in Section 3.2).

5 Examples

5.1 A Metabolism Example

Consider a fragment of the aromatic amino acid pathway of yeast, shown in Figure 3 (this is a fragment of a larger structure used in [1]). Rectangles are labelled with enzyme names, meaning that a respective enzyme is to be available for reaction, i.e., that av holds. For example, $av(ydr127w, r)$ is necessary, when the label of r is "YDR127W".

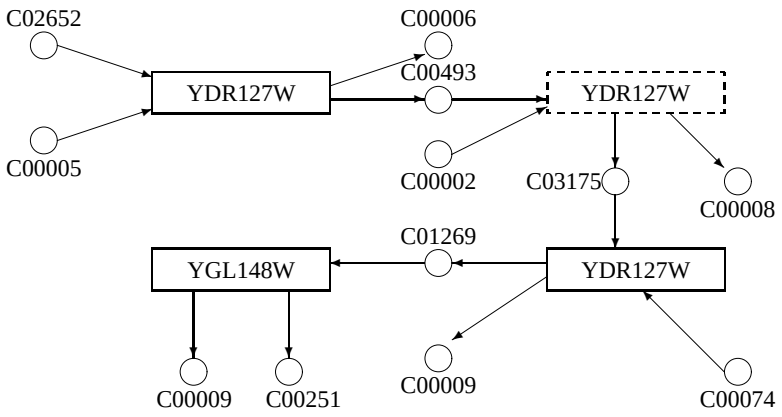

Fig. 3. A fragment of the aromatic amino acid pathway of yeast.

Figure 3 depicts the following reactions:

$$\begin{aligned}
&\mathrm{n}_1 : \mathrm{C02652} + \mathrm{C00005} \overset{\mathrm{YDR127W}}{\longrightarrow} \mathrm{C00006} + \mathrm{C00493}\\
&\mathrm{n}_3 : \mathrm{C03175} + \mathrm{C00074} \overset{\mathrm{YDR127W}}{\longrightarrow} \mathrm{C01269} + \mathrm{C00009}\\
&\mathrm{n}_4 : \mathrm{C01269} \overset{\mathrm{YGL148W}}{\longrightarrow} \mathrm{C00009} + \mathrm{C00251}.
\end{aligned}$$

It is assumed that reaction

$$\mathrm{n}_2 : \mathrm{C00493} + \mathrm{C00002} \overset{\mathrm{YDR127W}}{\longrightarrow} \mathrm{C03175} + \mathrm{C00008}$$

depicted by the dashed box is, in fact, missing.

The above set of reactions is expressed by formulas as defined in Section 2.2. For example, the first reaction is expressed by:

$$\begin{aligned}
&react(n_1, R) \rightarrow\\
&\quad av(ydr127w, R) \wedge\\
&\quad av(c02652, R) \wedge av(c00005, R) \wedge\\
&\quad \forall R'.\,[prec(R, R') \rightarrow av(c00006, R')] \wedge\\
&\quad \forall R'.\,[prec(R, R') \rightarrow av(c00493, R')].
\end{aligned}$$

The missing reaction is also present, among many other reactions, in the database, and is expressed by:

$$\begin{array}{l} react(n_2, R) \rightarrow \\ \quad av(ydr127w, R) \wedge \\ \quad av(c00493, R) \wedge av(c00002, R) \wedge \\ \quad \forall R'.\,[prec(R, R') \rightarrow av(c03175, R')] \wedge \\ \quad \forall R'.\,[prec(R, R') \rightarrow av(c00008, R')]. \end{array}$$

We assume that the underlying database contains partial information about the observed chain of reactions:

$$react(n_1, r_1) \wedge react(n_3, r_3) \wedge react(n_4, r_4)$$

together with a description of reactions n_1, n_3, n_4 and many other reactions, including n_3. Let the considered knowledge base be denoted by KDB. We assume, for simplicity, that no precedence of reactions is excluded, $\forall R, R'.\, notExcl_{\oplus}(R, R')$.

We can now consider, e.g., WSC(α; KDB; av), where

$$\alpha \stackrel{\text{def}}{\equiv} \exists N.[react(N, r_2) \wedge prec(r_1, r_2) \wedge prec(r_2, r_3)],$$

providing one with the weakest requirement expressed in terms of av only, making α true, provided that the background theory given by KDB holds.

In our case, the generated hypotheses will contain the disjunct $av_+(c00002, r_2)$, reflecting, among others, sufficient conditions for $prec$, expressed by (6).

The SNC(α; KDB;$\{out\}$) will contain the disjunct

$$out_+(N, c03175) \wedge out_+(N, c00008),$$

reflecting, among others, necessary conditions for $prec$, expressed by (7). If one of the compounds $c03175$, $c00008$ has not been observed during the reaction chain, one can reject the hypothesis that reaction N in node r_2 was n_2.

5.2 Synthesis of 3-bromo-4propylphenol from Benzene

This example shows how to express additional conditions on reactions.

Consider the following reactions for synthesis of 3-bromo-4propylphenol from benzene (see [14]):[7]

n_1: $\mathrm{C6H6} + \mathrm{C3H5OCl} + \mathrm{AlCl3} \longrightarrow \mathrm{C9H10O}$
n_2: $\mathrm{C9H10O} + \mathrm{Zn(H5)} + \mathrm{HCl} \longrightarrow \mathrm{C8H12}$
n_3: $\mathrm{C9H12} + \mathrm{HNO3} + \mathrm{H2SO4} \longrightarrow \mathrm{C9H11NO2}$
n_4: $\mathrm{C9H11NO2} + \mathrm{Br2} + \mathrm{FeBr3} \longrightarrow \mathrm{C9H10NO2Br}$
n_5: $\mathrm{C9H10NO2Br} + \mathrm{Sn} + \mathrm{HCl} \longrightarrow \mathrm{C9H12NBr}$
n_6: $\mathrm{C9H12NBr} + \mathrm{NaNO2} + \mathrm{H2SO4} + \mathrm{H2O} \stackrel{0^{\circ}\mathrm{C+heat}}{\longrightarrow} \mathrm{C9H11OBr}$

In order to express the condition "$0^{\circ}\mathrm{C} + \mathrm{heat}$", one has to extend the language by adding the following relations:

[7] The condition in reaction n_6 is intended to mean that the reaction has to start at a temperature of $0^{o}C$ and then the reactants are to be heated.

- $temp(R, T)$ meaning that the temperature in node R is initially T
- $heat(R)$ meaning that compounds in node R are heated.

Now "$0^{\circ}\mathrm{C} + \mathrm{heat}$" is expressed by $temp(R, 0) \wedge heat(R)$ and the reaction n_6 is expressed by:

$$
\begin{aligned}
&react(n_6, R) \rightarrow \\
&\quad temp(R, 0) \wedge heat(R) \wedge \\
&\quad av(c9h12nbr, R) \wedge av(nano2, R) \wedge \\
&\quad av(h2so4, R) \wedge av(h2o, R) \wedge \\
&\quad \forall R'.\, [prec(R, R') \rightarrow av(c9h11obr, R')].
\end{aligned}
$$

The hypotheses generated here might also have to be expressed in terms of a richer language. For example, the weakest sufficient condition for reaction n_6 to react in node r would involve $temp(r, 0) \wedge heat(R)$. In fact, no condition expressed in terms of relations introduced in Section 2.2 would make n_6 feasible, since none of them implies $temp(r, 0)$ or $heat(R)$.

6 Conclusions

In the paper we have presented a logical model of biochemical pathways and have shown how abductive hypothesis generation may be used to provide additional information about incomplete pathways. Hypothesis generation is achieved using weakest and strongest necessary conditions which explain observations about the functional and physical interactions being modeled.

The language for expressing knowledge about biochemical pathways that permits second-order quantifier elimination is quite broad and includes semi-Horn formulas (see [12]), a formalism substantially more expressive than the Horn fragment of the first-order logic. However, if one uses non-recursive semi-Horn formulas, the resulting formalism is much more efficient.

One can extend the approach in various directions. In particular, one can model much more complicated conditions required for reactions as well as time dependencies.

The approximate database implementation as well as our earlier work on snc's and wsc's has been supported in part by the WITAS project grant under the Wallenberg Foundation, Sweden.

References

1. Bryant, C., Muggleton, S., Oliver, S., Kell, D., Reiser, P., King, R.: Combining inductive logic programming, active learning and robotics to discover the function of genes. Linkoping Electronic Articles in Computer and Information Science **6** (2001)
2. Deville, Y., Gilbert, D., van Helden, J., Wodak, S.: An overview of data models for the analysis of biochemical pathways. Briefings in Bioinformatics **4** (2003) 246–259
3. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited. Journal of Automated Reasoning **18** (1997) 297–336

4. Gabbay, D.M., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic. In Nebel, B., Rich, C., Swartout, W., eds.: Principles of Knowledge Representation and Reasoning, KR 92. (1992) 425–435
5. Nonnengart, A., Szałas, A.: A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In Orłowska, E., ed.: Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa, Springer Physica-Verlag (1998) 307–328
6. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. International Joint Conference on AI (IJCAI'2001) (2000) 145 – 151
7. Doherty, P., Kachniarz, J., Szałas, A.: Using contextually closed queries for local closed-world reasoning in rough knowledge databases. In Pal, S., Polkowski, L., Skowron, A., eds.: Rough-Neuro Computing: Techniques for Computing with Words. Cognitive Technologies, Springer–Verlag (2003) 219–250
8. Chabrier, N., Fages, F.: Symbolic model checking of biochemical networks. In: Proceedings of the First International Workshop on Computational Methods in Systems Biology CMSB'03. Volume 2602 of LNCS., Springer-Verlag (2004) 149–162
9. Chabrier-Rivier, N., Fages, F., Soliman, S.: The biochemical abstract machine BIOCHAM. In: Proceedings of the Second International Workshop on Computational Methods in Systems Biology CMSB'04. LNCS, Springer-Verlag (2004)
10. Doherty, P., Kertes, S., Magnusson, M., Szałas, A.: Towards a logical analysis of biochemical pathways (extended abstract). In: Proceedings of the European Conference on Artificial Intelligence ECAI'2004. (2004) to appear.
11. Lin, F.: On strongest necessary and weakest sufficient conditions. In Cohn, A., Giunchiglia, F., Selman, B., eds.: Principles of Knowledge Representation and Reasoning, KR'2000. (2000) 167–175
12. Doherty, P., Łukaszewicz, W., Szałas, A.: A reduction result for circumscribed semi-Horn formulas. Fundamenta Informaticae **28** (1996) 261–271
13. Doherty, P., Łukaszewicz, W., Szałas, A.: General domain circumscription and its effective reductions. Fundamenta Informaticae **36** (1998) 23–55
14. Pine, S., Hendrickson, J., Cram, D., Hammond, G.: Organic Chemistry. McGraw-Hill, Inc. (1980)