Exploring Chemistry Using SMT

Rolf Fagerberg[1], Christoph Flamm[2], Daniel Merkle[1], Philipp Peters[1]

[1] Department of Mathematics and Computer Science University of Southern Denmark {daniel,phpeters,rolf}@imada.sdu.dk
[2] Institute for Theoretical Chemistry, University of Vienna, Austria xtof@tbi.univie.ac.at

Abstract. How to synthesize molecules is a fundamental and well studied problem in chemistry. However, computer aided methods are still under-utilized in chemical synthesis planning. Given a specific chemistry (a set of chemical reactions), and a specified overall chemical mechanism, a number of exploratory questions are of interest to a chemist. Examples include: what products are obtainable, how to find a minimal number of reactions to synthesize a certain chemical compound, and how to map a specific chemistry to a mechanism. We present a Constraint Programming based approach to these problems and employ the expressive power of Satisfiability Modulo Theory (SMT) solvers. We show results for an analysis of the *Pentose Phosphate Pathway* and the *Biosynthesis of 3-Hydroxypropanoate*. The main novelty of the paper lies in the usage of SMT for expressing search problems in chemistry, and in the generality of its resulting computer aided method for synthesis planning.

1 Introduction

The rigorous study of the properties of naturally occurring molecules requires their chemical synthesis from simpler precursor compounds. Therefore total synthesis of natural products is one of the fundamental challenges of organic chemistry. Chemical synthesis involves multistep synthetic sequences of elementary reactions. An elementary reaction transforms a set of chemical compounds (*reactants*) in a single step into a new set of chemical compounds (*products*) which are structurally different from the reactants. The step by step sequence of elementary reactions accompanying overall chemical change is denoted as *reaction mechanism*.

Finding a suitable sequence of elementary reactions leading from simple building blocks to a target molecule is in organic chemistry commonly referred to as the *synthesis planning problem*. Synthesis planning is a combinatorial complex problem and several heuristic approaches have been suggested [19] to attack this problem. Among synthetic chemists the *retrosynthetic analysis* [5] is one of the most popular approaches. This strategy systematically simplifies the target molecule by repeated bond disconnections in retrosynthetic direction, leading to progressively smaller precursors until recognized starting material emerges. Heuristic criteria are used to rank competing routes. Several computer programs are available implementing this approach (for a recent review see [4]).

With the advent of Synthetic Biology and Systems Chemistry the need for rational design of molecular systems with pre-defined structural and dynamical properties has been shifted into the focus of research. Over the last century mathematical prototype models for a great variety of chemical and biological systems with interesting nonlinear dynamic behaviour such as oscillation have been collected and mathematically analysed. The translation of the mathematical formalism back into real world chemical or biological entities, i.e., finding an instantiation of such abstract mechanisms in real chemical molecules (required for the rational design of de-novo molecular systems), is still an unsolved problem. The problem can be rephrased for chemical reaction systems in the following way: finding a set of compatible molecules that react according to a reaction mechanism which was translated from an abstract mathematical prototype model. Of course the solution of this inverse problem is usually not unique, and crucially depends on information that can be provided in a declarative manner via constraints. An example of this is the chosen chemistry, i.e. molecules and reactions that can be employed for solving the underlying problem.

Note that the declarative approach allows for many different levels of modeling, with varying degrees of realism. In this paper, we propose a post-processing step applied to the (possibly intentionally underspecified) declarative solution, to extract real-world chemical solutions.

This paper introduces to our knowledge for the first time a Satisfiability Modulo Theories (SMT) based approach to the problem of chemical synthesis. In addition to the standard product-oriented methods usually employed, our approach covers a significantly wider collection of chemical questions. Satisfiability solvers are used predominantly on computer science related problems, but they have also been used to tackle chemically or biologically relevant topics. In [11] "Synthesizing Biological Theories" were introduced in order to construct high-level theories and models of biological systems. In [**?**] constraint logic programming was used for chemical process synthesis in order to design so-called Heat Exchanger Networks (which is very different from compound synthesis as it will be discussed in this paper).

This paper is organized as follows. In Section 2, we describe how we model chemistry. In Section 3, we describe the chemical search space, and the constraints on it which the user may impose. In Section 4, we focus on the SMT-formulation, and in Section 5, we explain our post-processing. Finally, we in Section 6 present tests of our approach on several instances from real-world chemistry.

2 Modeling Chemistry

2.1 The Reaction Mechanism

When modeling a chemical synthesis, it is necessary to define the underlying mechanism, i.e., the molecules involved and the chemical reactions specifying how they are transformed. For our purposes, a sufficient modeling of an *elementary reaction* is a change of at most 2 reactant molecules into at most 2

product molecules. Almost all real-world elementary chemical reactions can be viewed this way, if necessary by splitting up reactions with a larger number of participating molecules. Such elementary reactions fall into four sub-categories with different numbers of participating molecules: isomerization (1-to-1), merging (2-to-1), splitting (1-to-2), and transfer (2-to-2).

A reaction mechanism is a combination of elementary reactions. It defines how many and which molecules react in each of these single reactions. Chemists usually denote the elementary reactions of the reaction mechanism as follows:

$$m(A + B \to C + D),$$

where A, B, C, D denote the molecules and m denote the multiplicity of the reaction in the mechanism, i.e. how many times the reaction happens.

In more formal terms, a *reaction mechanism* is a directed multi-hypergraph $G(V, E)$. Each vertex $v \in V$ represents a molecule. The directed hyperedges E represents the elementary reactions in the mechanism: each hyperedge $e \in E$ is a pair (e^-, e^+) of multisets $e^-, e^+ \subseteq V$ of molecules, denoting the reactants and products of the chemical reaction [2], and coefficient m_e represents the multiplicity of the hyperedge. Thus, the reaction $m(A + B \to C + D)$ is represented by the hyperedge $(\{A, B\}, \{C, D\})$ with multiplicity m.

The *balance* $\mathrm{bal}(v)$ of molecule $v \in V$ in a reaction mechanism is defined as an integer number indicating its net production or consumption over the entire synthesis:

$$\mathrm{bal}(v) = \sum_{e \in E} m_e(\mathbf{1}_{e^+}(v) - \mathbf{1}_{e^-}(v)), \tag{1}$$

where $\mathbf{1}_\alpha$ is the multiplicity function on the multiset α. If $\mathrm{bal}(v) < 0$, v is a reactant of the overall synthesis. If $\mathrm{bal}(v) > 0$, $v \in V$ is an end product. If $\mathrm{bal}(v) = 0$, either molecule v does not take part in the synthesis, or is produced and consumed in equal amount during the synthesis.

A related concept is the *overall reaction* of a reaction mechanism. It is defined by summing up the two sides of all reactions (including multiplicities) in the mechanism, cancelling out equal amounts of identical molecules appearing on both sides. Thus, the left hand side of the overall reaction is given by the molecules with negative balance, and the right hand side by the molecules with positive balance.

2.2 The Molecules

In the large field of organic chemistry, the properties of carbon based molecules are studied. Most properties of such molecules are determined by *functional groups* attached to a backbone of carbon atoms. Functional groups are reactive subparts of molecules and define the characteristic physical and chemical properties of families of organic compounds [13]. In Fig. 2 and Fig. 6 some examples are given. Fig. 1 illustrates how a chemical reaction changes the occurrences of functional groups by transferring atoms from the first to the second molecule, and opening the ring of the second molecule. The removed groups are marked

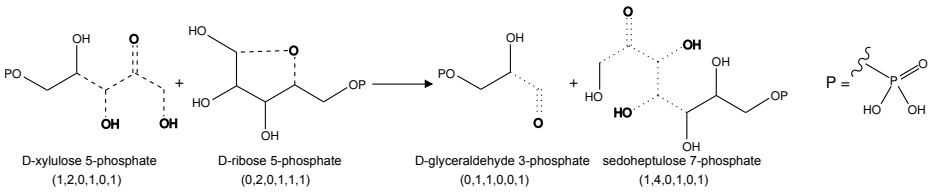

Fig. 1: The chemical reaction of D-xylulose 5-phosphate and D-ribose 5-phosphate to D-glyceraldehyde 3-phosphate and sedoheptulose 7-phosphate. The dashed-marked functional groups on the left side are removed during the reaction, the dotted groups on the right side are created. Non-participating groups are left black. The vectors of functional groups correlate with Fig. 2. For clarity, the phosphate group is substituted by "P".

dashed, the appearing groups are drawn dotted. Note that the number of untouched functional groups (black) in the products does not change. Line segment ends and junctions without annotations signify carbon atoms. In the example each of the reactants has one phosphate group, hence each product has a phosphate group.

In this paper, we model each molecule by a *vector of functional groups*. Position i in molecule A's vector provides the number of occurrences of the functional group x_i in A. The same vector is used for all molecules, i.e., there is one global set of functional groups. This set of functional groups is determined by the user, based on the chemistry, i.e., on what functional groups are deemed relevant to model for the molecules and reactions considered. Once this choice has been made, the functional groups are in our modeling simply positions in a vector. The length of that vector is the number of functional groups modeled, and the mapping between positions in vector and functional groups is arbitrary, but fixed.

This vector representation neglects the spatial structure of a molecule, i.e., only the number of occurrences of a functional group is noted, not its position(s) in the molecule. A chemical reaction is deemed feasible if its participating functional groups are present, irrespectively of whether these appear in the positions necessary for the reaction to take place. This implies that there may not be a real-world chemically valid equivalent to our vector-based reaction mechanism. Note that precise modeling of the chemical implications of the spatial structure of molecules is a hard problem in any formalism, with SMT being no exception. However, in Section 5 we provide a post-processing method which will allow us to filter our set of vector-based reaction mechanism, and retain only chemically viable solutions.

2.3 The Elementary Reactions

An elementary reaction can be defined formally in many ways in artificial chemistry [7], e.g. in a topological way or as a graph rewrite rule [3]. We here model

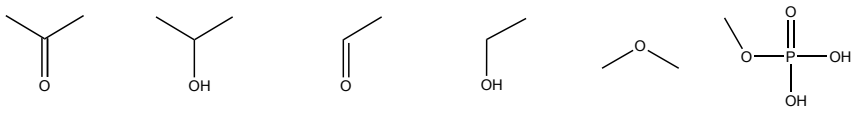

Fig. 2: The utilized functional groups in the order corresponding to their position in the molecule's vectors for modeling the reaction in Fig. 1 and the mechanism in Sec. 6.1.

an elementary reaction by its change of the number of occurrences of the functional groups of the reactants, i.e., its change of their vector representations. We call such a specified change of vectors a *rule*. In addition, a rule specifies preconditions which must hold for the reactants vector representations. Trivially, a specific functional group has to appear in a reactant when a chemical reaction reduces the number of this functional group. However, in chemistry, it may also be necessary that a specific other functional group appears in a reactant for the reaction to take place, and we allow this to be specified, too. This is for each reactant in a reaction expressed by a vector, whose entries state the minimum number of each functional group which is required to be present for the reaction to take place. A concrete example of a rule appears at the start of Section 4.3.

3 Exploring Chemistry

Our overall goal is a system allowing chemists to explore questions of the following general format: can a subset of a given base set of reactions fit together to form a reaction mechanism fulfilling some given constraints? In Section 2.3, we defined rules (as preconditions and change vectors on a predefined set of functional groups), which is how the user will specify the base set of reactions.

In this section, we describe how reaction mechanisms are modeled in our system, how the system attempts to map rules to this, and the multiple ways the user can specify constraints on the reaction mechanism.

We note that our basic modeling of a reaction mechanism is very general. Our philosophy is to not limit beforehand what types of problems the chemist can tackle, while at the same time supplying a large collection of possibilities for expressing constraints on the reaction mechanism. These possibilities the user can utilize to adjust and narrow the search space in each concrete chemical setting, based on preferences, available knowledge, goal of the investigation, and new information learned during the exploration. Two concrete investigative approaches are described in Section 3.3.

3.1 Search Space

Rule Mapping As defined above, a reaction mechanism is a multi-hypergraph with a vertex set V of molecules and an edge set E of reactions. In the search

phase, each molecule in V is considered a vector of integer variables (namely, a counter for each functional group in the modeling), and the task of the system is to find values for these variables compatible with a subset of the rules supplied by the user. This means that for each edge, there must be a rule assigned to it for which the values of the variables in the nodes of the edges fulfil the constraints (change in vectors, preconditions of reactants) of that rule. We call such an assignment of rules to edges a *rule mapping*.

Thus, finding a solution to the specified chemical search problem means finding a rule mapping, and a set of values for the variables in the nodes, compatible with each other, as well as with any further constraints on the reaction mechanism specified via the methods described in Section 3.2. The task of the SMT-solver in our system is to find such a solution.

In the remainder of the paper, we will when needed distinguish between a reaction mechanism with nodes considered as variables (as described above) and a reaction mechanism constrained by requiring nodes to be specific molecules (either in the vector representation or real molecules) by using the terms *abstract reaction mechanism* and *concrete reaction mechanism*, respectively.

In our most general setup, we consider a mechanism of n 2-to-2 reactions (hence with $4n$ vector-valued variables representing the molecules), and m rules. No constraints to the structure of the mechanism are made, and each rule can be mapped onto every reaction (including e.g. a rule with one reactant and one product being mapped to a 2-to-2 reaction, in which case the two unused variables in the reaction mechanism are implicitly defined as null vectors).

Note that a rule mapping can be done in different ways: Consider a specific 2-to-2 rule, which is mapped to a reaction $A + B \rightarrow C + D$. The rules will be defined by two precondition (p_1 and p_2) and two change vectors (δ_1 and δ_2) for the two reactants. When mapping the rule to the reaction, four possibilities exist depending on whether p_1 and δ_1 is taken as the $A \rightarrow C$, $A \rightarrow D$, $B \rightarrow C$, or $B \rightarrow D$ change (with p_2 and δ_2 in all cases giving the change for the remaining part of the reaction).

Equivalence Relation In our most general setup, the system is free to identify different node variables when looking for a solution. Then part of the output will be an equivalence relation $id : V \times V$ which is used in order to define the identity of two variables in a reaction mechanism. An equivalence relation id on molecules implies:

$$\forall v, w \in V : (v, w) \in id \Rightarrow \forall x_p \in \text{functional groups} : v(x_p) = w(x_p), \qquad (2)$$

where $v(\cdot)$ denotes the number of occurrences of x_p in v. I.e., defining two variables to refer to the same molecule implies that occurrences of subgroups are identical.

Multiplicities *Multiplicities of single reactions* in the mechanism denote how often a reaction takes place. The overall consumption and production of molecules

$$\begin{array}{ll} m_1\ (\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C} + \mathrm{D}) & \\ & \Rightarrow \\ m_2\ (\mathrm{E} + \mathrm{F} \rightarrow \mathrm{G} + \mathrm{H}) & \end{array} \qquad \begin{array}{l} m_1\ (\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C} + \mathrm{D}) \\ \\ m_2\ (\mathrm{C} + \mathrm{D} \rightarrow \mathrm{H}) \end{array}$$

Fig. 3: A general reaction mechanism (left side) without constrained equivalence classes. A predefined equivalence relation $\{(C, E), (D, F)\}$ leads to the reaction mechanism on the right side; G is supposed to be an empty molecule.

is defined by the sum of all produced occurrences minus the sum of all consumed occurrences, including the multiplicities of reactions. To provide the largest generality, in our model the multiplicities do not have to be specified. They are also part of the solution to be found.

3.2 Constraining the Search Space

To answer different chemical questions and to incorporate previous chemical knowledge, our system allows for the specification by the user of a number of additional constraints, which we now present.

Mechanism Specification By having a predefined equivalence relation id, a generic reaction mechanism can be constrained by specifying predefined equivalence classes of identical molecules. Figure 3 gives an example: On the left side a short generic reaction mechanism of two reactions is shown. Using the predefined equivalence relation $\{(C, E), (D, F)\}$ leads to the reaction mechanism on the right side. Molecules which do not take part (in the example molecule G) in a reaction are left out.

The pre-definition of equivalence classes implies that rules must now map exactly onto the reactions in terms of number of participating compounds. E.g., in contrast to the case where the equivalence relation is not predefined, a rule mapping of a 2-to-1 rule to a 2-to-2 reaction is not allowed. From a chemical perspective, such predefined identities imply an already known reaction mechanism and is an instance of the *Inverse Reaction Mechanism Problem* (see below).

The number n of reactions is always specified in our setup. If the user wants to e.g. search for a minimal n for which solutions exist, several runs with differing value of n can be done.

Balance and Overall Reaction Another constraint for the reaction mechanism can be set via the balance of molecules. As mentioned in Section 2, for each molecule a specified balance $\mathrm{bal}(v), v \in V$ for the whole mechanism holds. Especially in the product-oriented approach it makes sense to specify a desired amount of the product by a positive balance, and an amount of reactants by a negative balance. If it is assumed that there are no side products in a synthesis produced, $\mathrm{bal}(v) = 0$ can be set for all other molecules. (However, any so-called food and waste molecules, which provide energy, like Adenosine triphosphate

(ATP), or can be consumed and produced in infinite amounts, like H_2O, should have their balance unconstrained.)

Multiplicities In the most generic approach, the reaction mechanism and the multiplicities m_e of reactions are not specified in advance. But if it makes chemical sense to restrict the multiplicities, this of course can be specified. This could be done e.g. to prevent the solver to add chemically implausible high frequencies of single reactions to the solution. Note that balances and multiplicities are linked by Eq. (1). Only solutions fulfilling this will be produced.

Molecule Size Also the number of functional groups, or the number of a specific functional group in a molecule can be restricted. This can be employed in order to find a solution using a minimum necessary (or at least a small) number of functional groups for a desired reaction mechanism. Like a constraint on the multiplicities, a restriction on the molecule size may be used to prevent solutions with (chemically) unrealistic numbers.

3.3 Product-Oriented Exploration vs. IRMP

This generic approach introduced above is used in order to pose and answer different chemical questions. We distinguish two major lines of questions:

In a *product-oriented exploration* of chemistry, the properties of a desired product are known (in terms of functional groups or even as specific molecules). This knowledge serves as constraints to the molecule's vectors defining the number of occurrences of functional groups. This corresponds to a classical question of how to synthesize a specific compound based on a given set of chemical reactions. Based on existing chemical knowledge, a suggestion for the abstract reaction mechanism (including the equivalence relation for its molecules) for the synthesis may actually be known, or, more likely, it may be unknown.

A new approach for synthesis planning, and even more importantly, for understanding chemical reaction patterns, is what we define in this paper as the *Inverse Reaction Mechanism Problem (IRMP)*: In the *IRMP* it is assumed that an underlying reaction mechanism of a synthesis is known. Then, it is investigated if for the same abstract mechanism a *different* set of elementary chemical reactions (rules) can be mapped to it (potentially generating different molecules). In our model this corresponds to finding rule mappings and multiplicities, (but not equivalence classes of identical molecules, as this is assumed to be known), based on set of elementary chemical reactions different from the ones originally participating in the reaction mechanism.

4 The SMT-Implementation

We have implemented the approach delineated above using SMT. In this section, we present central parts of this implementation. The language used is

SMT-LIB [17] and the SMT-Solver is Microsoft's Z3 [6]. Our implementation creates an SMT program, based on input specification files that define preconditions/changes for the rules, the predefined balances, and the equivalence constraints. The auto-generated program is then handed over to the SMT solver.

4.1 Declarations

A subset of the most important data types and functions will be defined here. For concreteness, we as example use the second reaction from Fig. 4, representing the chemical reaction from Fig. 1. The capital letters `A,C,D,E` constitute the type `MOL` representing molecules, the lowercase letters `a,b,c,d,e,f` constitute the type `SUB` of functional groups, and the mechanism's reactions `REACT` are numbered from 1 to n.

```
(declare-datatypes () ((MOL  A C D E)))
(declare-datatypes () ((SUB  a b c d e f)))
(declare-datatypes () ((REACT  react1 react2 ... reactn)))
```

The function `NrOfGroups` provides a non-negative number of occurrences of each functional group for each molecule. Due to simplicity of summing up balances later, *stoichiometric coefficients* `STOI` for the molecules in each reaction are defined. In the notation of chemistry this means that `STOI` for a reactant is the negatively signed value of the multiplicity of the reaction, whereas it is the same but positively signed value for each product. The equivalence relation is implemented as a Boolean matrix `ID` and provides the identity of molecules. `PRODUCT` and `REACTANT` functions provide Boolean values and define if a molecule should be a product or reactant in the whole reaction mechanism.

```
(declare-fun NrOfGroups (MOL SUB) Int)
(declare-fun STOI (REACT MOL) Int)
(declare-fun ID (MOL MOL) Bool)
(declare-fun REACTANT (MOL) Bool)
(declare-fun PRODUCT (MOL) Bool)
```

4.2 Equivalence Relation

Additional to the properties of the equivalence relation (reflexivity, symmetry, transitivity) for the equivalence relation *id*, an implication has been implemented. Two molecules being in the same class implies the same number of occurrences for all functional groups (cmp. Eqn. 2).

```
(assert (forall ((mol MOL)(mol2 MOL))
     (=>  (=(ID mol mol2) true)
     (forall ((sub SUB))
          (= (NrOrGroups mol sub) (NrOrGroups mol2 sub)) ))))
```

4.3 Mapping

In the following, an example of a rule mapping for a 2-to-2 reaction $A + C \rightarrow D + E$ will be presented. Picking the second reaction from Fig. 4, we will restrict the rule mapping to the case where molecule A is changed into molecule D (implying molecule C will change to E). Assume that the rule mapping is defined by the change vector (-1,-1,1,-1,0,0) and the precondition vector (1,1,0,1,0,0) for A, and change vector (1,2,0,0,-1,0) and precondition vector (0,0,0,0,1,0) for C. This leads to the following implementation:

```
(assert (and
     ; stoichiometry constraints:
     ; (the stoi. coeff. of A needs to be the negative of D, etc.)
     (< (STOI react1 A) 0)
     (= (STOI react1 A) (STOI react1 C))
     (= (STOI react1 A) (- (STOI react1 D)))
     (= (STOI react1 D) (STOI react1 E))
     (or  (and
          ;preconditions
           (>= (NrOrGroups A a) 1)  (>= (NrOrGroups A b) 1)
           (>= (NrOrGroups A d) 1)  (>= (NrOrGroups C e) 1)
          ;changes made to A, which results in D
           (= (NrOrGroups D a) (- (NrOrGroups A a) 1))
           (= (NrOrGroups D b) (- (NrOrGroups A b) 1))
           (= (NrOrGroups D c) (+ (NrOrGroups A c) 1))
           (= (NrOrGroups D d) (- (NrOrGroups A d) 1))
           (= (NrOrGroups D e) (NrOrGroups A e))
           (= (NrOrGroups D f) (NrOrGroups A f))
          ;changes made to C, which results in E
           (= (NrOrGroups E a) (+ (NrOrGroups C a) 1))
           (= (NrOrGroups E b) (+ (NrOrGroups C b) 2))
           (= (NrOrGroups E c) (NrOrGroups C c))
           (= (NrOrGroups E d) (NrOrGroups C d))
           (= (NrOrGroups E e) (- (NrOrGroups C e) 1))
           (= (NrOrGroups E f) (NrOrGroups C f))
     ))))
```

4.4 Balance

For all molecules, a balance for the whole mechanism can be defined. In the following example the balance of a product molecule (i.e. `PRODUCT(mol)` is `true`) shall be greater than zero or equal to a specific positive amount.

$$\forall \texttt{mol} \in \texttt{MOL} : \texttt{PRODUCT(mol)} \Rightarrow \sum_{\texttt{r}=1}^{n} \texttt{STOI(r,mol)} > 0$$

The SMT code implementing this constraint is:

```
; balance for product should be > 0, symmetric for reactant
(assert(forall(mol MOL)
     (=> (= (PRODUCT mol) true)
     (> (+(STOI react1 mol)(STOI react2 mol)
     . . .
     (STOI reactn mol)) 0))))
```

The negative amount for the reactant can be constrained similarly, as well as the balance of value 0 for all other non-product or non-reactant molecules.

5 Post-processing

The solution output by the SMT-solver contains a rule mapping and a set of vector values, and is thus expressed in the vector representation of molecules. As noted earlier, this representation neglects the spatial structure of molecules, implying that false positives can occur in the sense that some found solutions may not have corresponding real-world chemical reactions.

In this section, we describe an automated post-processing method which allows us to filter our set of SMT-solutions, and retain only chemically viable solutions consisting of existing real-world chemical reactions.

The method is based on the existence of large chemical databases of reactions, such as KEGG [16]. The KEGG database is a biochemical database containing biochemical pathways and most of the known metabolic pathways. After converting an entry for a reaction in the KEGG database to a form searchable by the Graph Grammar Library [**?**,**?**], we in an automated way search for the appearances of the functional groups. From this, we generate the vector representations of its participating molecules, and deduce the rule version of that reaction. The IDs of the participating real-world molecules are stored with the rule. We then apply a straightforward search algorithm for finding a conversion of the SMT-generated reaction mechanism from vector representation to a form where nodes contain the IDs of real-world molecules and where all edges represent a real-world reaction from the database. This is done by first for each hyperedge e of the SMT-generated reaction mechanism finding the set K_e of KEGG reactions whose vector representation is compatible with that of e. Then for some fixed order $e_1, e_2, \ldots$ of the edges doing a backtracking DFS-type search for an assignment of KEGG reactions to edges for which the implied molecule IDs agree for all nodes. In details, the search starts by assigning the first reaction in K_{e_1} to e_1, recording the implied molecule IDs for the nodes of e_1, and then advancing to the next edge. If for an edge e_i no reaction in K_{e_i} can be found which is compatible with the molecule IDs implied by the currently assigned reactions for e_1 to e_{i-1}, the search backtracks, and tries the next edge of $K_{e_{i-1}}$.

The results of this post-processing may for each SMT-solution provide potentially many reaction mechanisms with real-world chemical reactions and real molecules, or it may find that none can be given based on the database in question. The post-processing may then be repeated with any further solutions from the SMT-solver.

6 Results

In this section, we will present results of our SMT-based exploring approach on chemistry, namely the well studied and well understood *Pentose Phosphate Pathway (PPP)* [10] and the industrial important *biosynthesis of 3-Hydroxypropanoate (3HP)*. As SMT-Solver, *Microsoft's Z3 SMT-Solver* was used on an Intel Core2 Duo CPU T7500 @ 2.20GHz with a memory size of 2 GB.

6.1 The Pentose Phosphate Pathway

The *PPP* can be found in most organisms, including mammals, plants and bacteria such as *E. coli*. It generates the co-enzyme *NADPH*, which takes part in many anabolic reactions as reducing agent, and a sugar with six carbon atoms (here: fructose 6-phosphate) [14]. Its products are used for the synthesis of nucleotides and amino acids. The PPP is also an alternative to the glycolysis which converts glucose into pyruvate and releases highly energetic molecules ATP (adenosine triphosphate). Our model of the PPP takes as input 6 sugar molecules with 5 carbon atoms (*pentoses*, here ribulose 5-phosphate) and releases 5 molecules of fructose 6-phosphate which has 6 carbon atoms.

Product-Oriented Exploration Starting with a set of reactions, the goal is to identify an abstract reaction mechanism to create a certain amount of fructose 6-phosphate.

Our instance consists of 7 2-to-2 reactions, where not all molecules have to appear in the latter abstract reaction mechanism. The multiplicities of reactions and the equivalence relation *id* are not restricted, and will be part of the solution. The properties and amounts of the reactant and the product are known. These two variables are predefined in the instance, and in addition their balances, too. The latter is done by the constraints: $\mathrm{bal}(\text{reactant}) = -6$ and $\mathrm{bal}(\text{product}) = 5$. Additionally, we specify a water and a phosphate molecule, whose balances are not restricted. All other molecules appearing in the mechanism should have balance zero.

For solving this instance, a set of molecules and a set of rules over these molecules are assumed to be given. These sets can be derived for example from a database request (as from KEGG). In our case we chose molecules and rules from the natural appearing PPP and added three additional sugar-molecule rules (giving 10 rules and 11 molecules in total). For this instance, we chose to let functional groups correspond to complete molecules. This implies that all rules remove exactly one "functional group" and add exactly one "functional group", namely the complete molecules.

A valid mapping of these rules to the reaction set instantiates a possible abstract reaction mechanism to synthesize fructose 6-phosphate in 7 reactions.

The auto-generated SMT-Program for this example was solved by Z3 in 128 minutes. Fig. 4 shows a solution of this instance in which the Pentose Phosphate Pathway occurs as it can be found in nature. If the equivalence classes of molecules are specified in advance, the solution is found in less than 10 seconds.

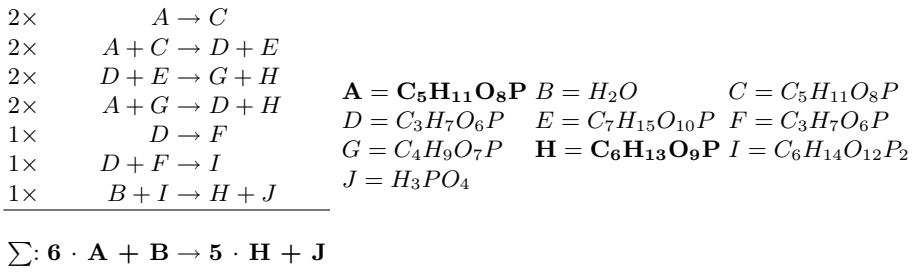

Fig. 4: The abstract PPP reaction mechanism as it occurs in nature, found by our SMT approach. The letters in the mechanism represents the molecules given to the right. Depicted in bold are ribulose 5-phosphate and fructose 6-phosphate.

Inverse Reaction Mechanism Problem The solution from the product-oriented approach provides an abstract reaction mechanism, i.e., the equivalence classes of identities of the molecules and the multiplicities are now fixed. By giving this, the *IRMP* is instantiated and a concrete reaction mechanism can be sought after. This means that molecules from the abstract mechanism are now seen as vectors of functional groups and one abstract mechanism can serve as template for several concrete mechanisms. The result will be highly dependent on the given set of rules; in this example we focus on finding the PPP. For testing reasons, the chemistry is chosen in a simple way, it consists of 8 rules from "sugar chemistry", based on 6 functional groups. These are shown in Fig. 2 where they are ordered as in the molecule's vectors. Note that the groups are not overlapping, they only share carbon atoms. As an example we illustrated a rule at the start of Section 4.3, where a *transketolase* (cf. Fig. 1 and second line of Fig. 4) was modeled, including change and precondition vectors. This 2-to-2 reaction transfers a fragment (a keto group) from one molecule to another.

The SMT-solver provided a solution in less than 2 seconds. The solution could by the way be seen to be minimal in the total number of occurring functional groups (using a smaller overall number of functional groups lead to unsatisfiability for the given set of rules), so artificially large molecules seemed to be avoided. Fig, 5 shows the concrete reaction mechanism with the vectors defining the number of functional groups in the molecules. Note that the second to last reaction was modeled here just as a 1-to-1 reaction, because water (B) is always available, also phosphate (J) serves as waste molecule and can be produced in an infinite amount. Based on the SMT solution, the post-processing step generates a known real-world synthesis from the PPP.

6.2 Biosynthesis of 3-Hydroxypropanoate

3-Hydroxypropanoate (3HP) is a high-value organic molecule and is used in numerous reactions. Usually it is organically synthesized, but biosynthetic pathways to this product are in high demand [18]. To produce 3HP from pyruvate (an

$$
\begin{array}{lcl}
2\times & (1\ 2\ 0\ 1\ 0\ 1) & \rightarrow (0\ 2\ 0\ 1\ 1\ 1) \\
2\times & (1\ 2\ 0\ 1\ 0\ 1) + (0\ 2\ 0\ 1\ 1\ 1) & \rightarrow (0\ 1\ 1\ 0\ 0\ 1) + (1\ 4\ 0\ 1\ 0\ 1) \\
2\times & (0\ 1\ 1\ 0\ 0\ 1) + (1\ 4\ 0\ 1\ 0\ 1) & \rightarrow (0\ 2\ 1\ 0\ 0\ 1) + (0\ 3\ 0\ 1\ 1\ 1) \\
2\times & (1\ 2\ 0\ 1\ 0\ 1) + (0\ 2\ 1\ 0\ 0\ 1) & \rightarrow (0\ 1\ 1\ 0\ 0\ 1) + (0\ 3\ 0\ 1\ 1\ 1) \\
1\times & (0\ 1\ 1\ 0\ 0\ 1) & \rightarrow (1\ 0\ 0\ 1\ 0\ 1) \\
1\times & (0\ 1\ 1\ 0\ 0\ 1) + (1\ 0\ 0\ 1\ 0\ 1) & \rightarrow (0\ 3\ 0\ 0\ 1\ 2) \\
1\times & B + (0\ 3\ 0\ 0\ 1\ 2) & \rightarrow (0\ 3\ 0\ 1\ 1\ 1) + J
\end{array}
$$

Fig. 5: concrete PPP reaction mechanism with vectors of occurrences of functional groups (cmp. Fig 2) of molecules as found by the SMT-Solver. Post-processing with molecules occurring in this solution of PPP leads to the PPP as it occurs in nature.

end product of the glycolysis), biosynthetic pathways have been assembled [9, 12] and additionally one pathway has already been implemented in an industrial setup [20].

The biosynthesis of 3HP was investigated here as an instance of the *IRMP*, i.e. an abstract reaction mechanism and the chemistry (a set of rules) are given, and the solution of this instance identifies possible pathways from pyruvate to the desired product 3HP. The vectors of the reactant and the product were predefined. A reaction mechanism of length n, consisting of only 1-to-1 reactions was used, as shown in Fig. 7. The multiplicities of reactions were set to 1. The equivalence classes of molecules were specified, and by doing so, a cascading mechanism $A \rightarrow B \rightarrow C \rightarrow \ldots$ was created. In total, 19 chemical 1-to-1 rules with 10 functional groups were used in order to define the chemistry. These rules were defined by chemical expertise (details omitted in this paper) as well as derived by a recent database-supported approach [8].

The concrete reaction mechanisms provided by the SMT-Solver were post-processed using the KEGG database. Due to space limitation, Fig. 7 shows only 3 of the 27 found pathways from the post-processing. The functional groups marked dashed disappear in the subsequent reaction, the bold-marked functional groups are pre-conditional for the reaction to take place. All pathways generated by [8] could be found.

Additionally, by post-processing a concrete mechanism of length 2, a solution for the *IRMP* could be found that does not produce 3HP. I.e., a pathway was found that employs exactly the same reaction pattern as the synthesis of 3HP but is based on a different set of molecules. The alternative two-step-pathway syntheses 2-phospho-D-glycerate from 3-phosphohydroxypyruvate, using KEGG-notation can be stated as $C03232 \xrightarrow{R01513} C00197 \xrightarrow{R01518} C00631$.

7 Conclusions

We introduced the combination of two rather different fields of research, namely Satisfiability Modulo Theories (SMT) and theoretical and real-world chemistry.

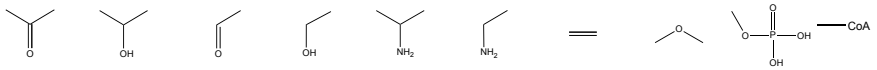

Fig. 6: The utilized functional groups in the order corresponding to their position in the molecule's vectors for the modeling of the biosynthesis of 3HP.

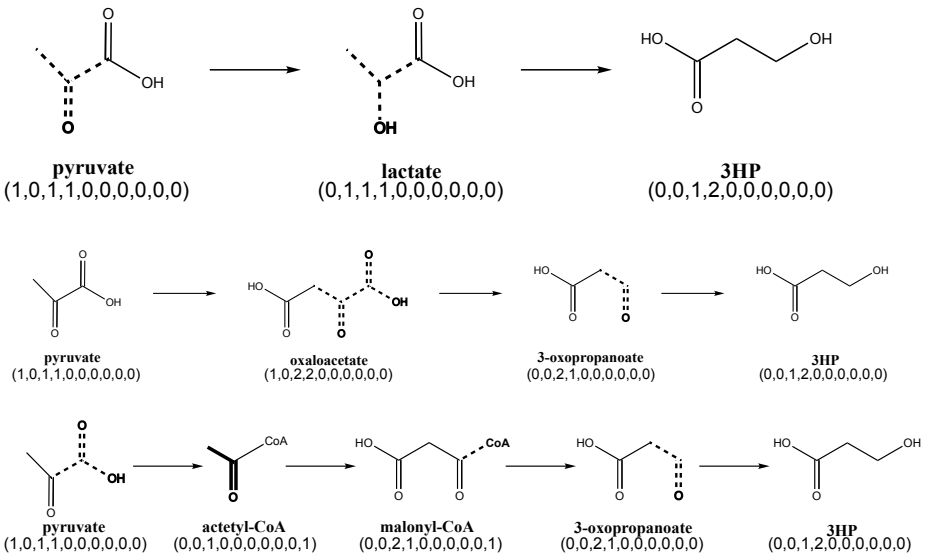

Fig. 7: Three example pathways for the biosynthesis of 3HP from pyruvate; the vectors underneath are using the functional groups and their order from Fig. 6; reacting functional groups are drawn dashed; functional groups which are necessary but remain unchanged are depicted bold.

.

Defining chemical questions like the synthesis of a specific compound or the search for pathway patterns formally as instances for SMT solvers allows to answer a large set of chemically highly relevant but so far unasked questions. To underline this we introduced and solved the Inverse Reaction Mechanism Problem (IRMP), which can be used to identify reaction mechanism patterns via SMT. Solutions to the IRMP might have significant impact on chemical compound fabrication and can help to understand patterns in chemical reaction mechanisms. We have shown the applicability of the new approaches on two real-world chemical setups, namely the analysis of the Pentose Phosphate Pathway and the biosynthesis of 3-Hydroxypropanoate.

References

1. H. Abbass, G. Wiggins, R. Lakshmanan, and B. Morton. Constraint logic programming for chemical process synthesis. 2007.
2. J. Andersen, C. Flamm, D. Merkle, and P. Stadler. Maximizing output and recognizing autocatalysis in chemical reaction networks is np-complete. *Journal of Systems Chemistry 2012*, 3(1), 2012.
3. G. Benkö, F. Centler, P. Dittrich, C. Flamm, B. Stadler, and P. Stadler. A topological approach to chemical organizations. *Artificial Life*, 15(1):71–88, 2009.
4. A. Cook, A. P. Johnson, J. Law, M. Mirzazadeh, O. Ravitz, and A. Simon. Computer-aided synthesis design: 40 years on. *WIREs Comput Mol Sci*, 2:79–107, 2012.
5. E. J. Corey. General methods for the construction of complex molecules. *Pure Appl Chem*, 14:19–38, 1967.
6. L. De Moura and N. Bjørner. Z3: An efficient smt solver. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, 2008.
7. P. Dittrich, J. Ziegler, and W. Banzhaf. Artificial chemistries-a review. *Artificial life*, 7(3):225–275, 2001.
8. C. Henry, L. Broadbelt, and V. Hatzimanikatis. Discovery and analysis of novel metabolic pathways for the biosynthesis of industrial chemicals: 3-hydroxypropanoate. *Biotechnology and bioengineering*, 106(3):462–473, 2010.
9. X. Jiang, X. Meng, and M. Xian. Biosynthetic pathways for 3-hydroxypropionic acid production. *Applied microbiology and biotechnology*, 82(6):995–1003, 2009.
10. N. Kruger and A. von Schaewen. The oxidative pentose phosphate pathway: structure and organisation. *Current Opinion in Plant Biology*, 6(3):236–246, 2003.
11. H. Kugler, C. Plock, and A. Roberts. Synthesizing biological theories. In *Computer Aided Verification*, pages 579–584. Springer, 2011.
12. A. Maris, W. Konings, J. Dijken, and J. Pronk. Microbial export of lactic and 3-hydroxypropanoic acid: implications for industrial fermentation processes. *Metabolic Engineering*, 6(4):245–255, 2004.
13. A. McNaught and A. Wilkinson. *IUPAC compendium of chemical terminology*, volume 2. Blackwell Scientific Publications, 1997.
14. D. Nelson, M. Cox, and M. Cox. *Lehninger Biochemie*. Springer, 2005.
15. R. Nieuwenhuis. Sat modulo theories: Getting the best of sat and global constraint filtering. In *Proceedings of the 16th International Conference on Principles and Practice of Constraint Programming (CP 2010)*, volume 6308 of *LNCS*, pages 1–2. Springer, 2010.
16. H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 27(1):29, 1999.
17. S. Ranise and C. Tinelli. The smt-lib standard: Version 1.2. *Department of Computer Science, The University of Iowa, Tech. Rep*, 2006.
18. P. Suthers, D. Cameron, et al. Production of 3-hydroxypropionic acid in recombinant organisms, Feb. 8 2005. US Patent 6,852,517.
19. M. H. Todd. Computer-aided organic synthesis. *Chem Soc Rev*, 34:247–266, 2005.
20. T. Willke and K. Vorlop. Industrial bioconversion of renewable resources as an alternative to conventional chemistry. *Applied microbiology and biotechnology*, 66(2):131–142, 2004.