A System for Semantic Analysis of Chemical Compound Names

Henriette Engelken
EML Research gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany;
Institute for Natural Language Processing
University of Stuttgart
Azenbergstr. 12
70174 Stuttgart, Germany
engelken@eml-research.de

Abstract

Mapping and classification of chemical compound names are important aspects of the tasks of BioNLP. This paper introduces the architecture of a system for the syntactic and semantic analysis of such names. Our system aims at yielding both the denoted chemical structure and a classification of a given name. We employ a novel approach to the task which promises an elegant and efficient way of solving the problem. The proposed system differs significantly from existing systems, in that it is also able to deal with underspecifying names and class names.

1 Introduction

BioNLP is the branch of computational linguistics developing tools and algorithms tailored to the life sciences domain. Scientific and patent literature in this domain are growing at an enormous pace. This results in a valuable resource for researchers, but at the same time it poses the problem that it can hardly be processed manually by humans. Thus, a major goal of BioNLP is to automatically support humans by means of research in the area of information retrieval, data mining and information extraction. Term identification is of great importance in these tasks. Krauthammer and Nenadic (2004) divide the identification task into the subtasks of term recognition (marking the interesting words in a text), term classification (classifying them according to a taxonomy or an ontology) and term mapping[1] (identifying a term with respect to a referent data source).

Chemical compound names, i. e. names of molecules, are terms which prominently occur in scientific publications, patents and in biochemical databases. Any chemical compound can be unambiguously denoted by its molecular structure, either graphically or by certain representation standards. Established representation formats are SMILES strings (Simplified Molecular Input Line Entry System (Weininger, 1988)) and InChIs [2]. For example, a SMILES string such as *CC(OH)CCC* unambiguously describes a chain of five carbon (C) atoms connected by single bonds having an oxygen (O) and a hydrogen (H) atom connected to the second carbon atom by another single bond (Figure 1).

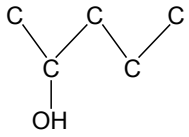

Figure 1: SMILES = *CC(OH)CCC*,
Name = *pentan-2-ol*

However, for communication purposes, e. g. in scientific publications and even in databases, it is common to use names for chemical compounds instead of a structural representation. Contrary to the structural representations, these names are neither always unique nor unambiguous. Biochemical terminology is a subset of natural language which appears to be highly regulated and systematic. The International Union of Pure and Applied Chemistry (IUPAC) (1979; 1993) has developed a nomenclature for chemical compounds. It specifies how to name a molecule systematically, as

[1] Term mapping is also called *term grounding*, amongst others by Kim and Park (2004).

[2] Cf. http://www.iupac.org/inchi/ (accessed May 17, 2009).

Proceedings of the ACL-IJCNLP 2009 Student Research Workshop, pages 36–44,
Suntec, Singapore, 4 August 2009.

well as by use of certain trivial names.

The morphemes constituting a name determine the chemical structure it denotes by specifying the type and number of the present atoms and bonds. Morphemes also interact with each other on this structural level. Typically, morphemes describe the atoms and bonds by introducing actions concerning so-called functional groups. About 50 different functional groups can be identified to be the most common ones in organic chemistry.[3] Functional groups are certain groups of atoms which determine the characteristic properties of a molecule, especially its chemical reactions. Hence, the presence or absence of certain functional groups plays a crucial role in classification of chemical compounds. For example, *hydroxy*, used as a prefix of a name, specifies the presence of an OH-group (consisting of an oxygen atom and a hydrogen atom). A molecular structure containing an OH-group can be classified to be an *alcohol*. The morpheme *dehydroxy* in contrast causes deletion of such an OH-group. Thus, it presupposes the existence of some OH-group, which consequently needs to be introduced by another morpheme of the given name. In case there is no additional OH-group left in this molecule after deletion, it does not belong to the class *alcohol*. Apart from addition and deletion, another frequent operation on functional groups, specified by the name's morphemes, is substitution. In this case, a presupposed functional group is replaced by a different functional group. Again, this may change the classes this chemical compound belongs to.

Despite the IUPAC nomenclature, name variations are still in use. On the one hand this is due to competing rules in different editions of the IUPAC nomenclature and on the other hand to the actual usage by chemists who can hardly know every single nomenclature rule. Thus, there can be a number of different names and name types for one chemical compound, namely several systematic, semi-systematic, trivial and trade names. For example, *pentan-2-ol* is the recommended name for the compound in Figure 1, but the same compound can be called *2-pentanol* or *2-hydroxypentane* as well.

Besides synonymy, names allow the omission of specific information about the structure of the compound they denote. This results in not only having a single compound as their reference but a whole set of compounds. Class names like *alcohol* or *alkene* are obvious cases. So-called underspecifying or underspecified[4] names (Reyle, 2006) like *pentanol*, *butene* or *3-chloropropenylidyne* also lack some structural information necessary to fully specify one compound, even though except for this, their names are built according to systematic naming rules. *Pentanol*, for instance, is missing the locant number and could hence stand for *pentan-1-ol*, *pentan-2-ol*, as well as *pentan-3-ol*. We distinguish underspecification from ambiguity, in that underspecifying names do not need to be resolved but denote a set of compounds, analogous to class names.

The particularities of chemical compound names mentioned above, namely synonymy, class names, underspecifying names and interaction between morpheme's meanings, complicate automatic classification and mapping of the names.

To achieve mapping of synonymous chemical compound names, name normalization is a possible approach. Rules can be set up to transform syntactic as well as morphological variations of names into a normalized name form. Basic transformations can be achieved via pattern matching (regular expressions) while for more complex transformations a linguistic parser, yielding a syntactic analysis, would be needed. For example, the names *glyceraldehyde-3-phosphate* and *3-phospho-Glyceraldehyde* could both be normalized to the form *3-phosphoglyceraldehyde* by such rules since the prefix *phospho* is synonymous with the suffix *phosphate*. This way, a synonym relation can be established between any two names which resulted in the same normalized name form. By using this method together with large reference databases[5] providing many synonymous names for their entries, the task of name mapping can be successfully solved in many cases.

However, there are limits to this string based approach. First, it relies on the quality of the referent data source and the quantity of synonyms provided by it. Currently available databases which could be used as a reference lack either quality or quantity. But whether a molecular structure for a term can be determined, or a term classi-

[3] Cf. (Ertl, 2003) and Wikipedia, *Functional group*, http://en.wikipedia.org/wiki/Functional_group (accessed May 17, 2009).

[4] Hereafter we will call these names *underspecifying names* because we consider them to underspecify a chemical structure rather than being underspecified.

[5] E. g. PubChem: http://pubchem.ncbi.nlm.nih.gov/ (accessed May 17, 2009).

fication can be achieved, depends only on this referent data source. Second, it is hardly possible to include every morphosyntactic name variation in the set of transformation rules. *2-hydroxy-3-oxopropyl dihydrogen phosphate*, for example, is the IUPAC name recommended for the chemical compound *glyceraldehyde-3-phosphate*, mentioned above. Obviously, a synonym relation can not be discovered by morphosyntactic name transformations in this case. Finally, this method is not able to deal with class names or underspecifying names.

These observations result in the need to take the meaning of a name's morphemes, i. e. the chemical structure, into account as well. A number of systems for name-to-structure conversion are being developed. The best known commercial systems are Name=Struct[6], ACD/Name[7] and Lexichem[8]. Being commercial, detailed documentation about their methods and evaluation results is not available. Academic approaches are OPSIN (Corbett and Murray-Rust, 2006) and ChemNomParse[9]. The greatest shortcoming of all these approaches is that they are not able to deal with underspecifying names. Instead, they either guess the missing information, in order to determine one specific structure for a given name, or simply fail. But for really underspecifying names and class names, to the best of our knowledge no chemical representation format, like a SMILES string, is provided. In addition, these approaches do not yield any classification of the processed names, regardless of whether these are underspecifying or not.

To overcome these limitations, CHEMorph (Kremer et al., 2006) has been developed. It contains a morphological parser, built according to the IUPAC nomenclature rules. The parser yields a syntactic analysis of a given name and also provides a semantic representation. This semantic representation can be used as a basis for further processing, namely for structure generation or classification. In the CHEMorph project, rules have been set up to achieve these two tasks, but there are limits in the number and correctness of structures and classes retrieved. These limits are partly due to the lack of a comprehensive valence and numbering model for the chemical structures. Also, classification should be based on the structural level rather than on the semantic representation, to ensure that not only the numbering but also default knowledge about chemical structures is included correctly.

The objectives of our own name-to-structure system are the following: Naturally, it should yield a chemical compound structure, in some representation format, as well as a classification for a given name. In case the name does not fully specify one compound, but refers to a set of structures, the system should still allow for structure comparison (mapping) and classification. Several default rules about the names and the chemical structures have to be taken into account. By including default knowledge, a structure can be specified further even if the name itself has left it underspecified. Similarly, a comprehensive way of dealing with valences of atoms has to be included, since the valences restrict the way a chemical structure can be composed.

Our approach to achieve these goals is to use constraint logic programming (CLP). CLP over graph domains is ideal for modeling each name-to-structure task as a so-called constraint satisfaction problem (CSP) and thereby accomplish mapping and classification. We will describe our system, CLP(name2structure), in more detail in the following section.

In this introduction we described the particularities of biochemical terminology. Related work in the area of processing these terms was overviewed and we gave the motivation for our own approach. After presenting our system in Section 2 we will conclude this paper with Section 3, indicating directions for future research.

2 Our Approach

Following Reyle (2006), we observed that any chemical compound name can be seen as a description of a chemical structure – in other words it contains constraints on how the structure is composed. Even if a partial name or a class name does not specify the structure completely but leaves a certain part underspecified, there will at least be some constraints about the structure. On account of this, our proposed system – CLP(name2structure) – employs constraint logic

[6]Cf. http://www.cambridgesoft.com/databases/details/?db=16 (accessed May 17, 2009).

[7]Cf. http://www.acdlabs.com/products/name_lab/rename/batch.html (accessed May 17, 2009).

[8]Cf. http://demo.eyesopen.com/products/toolkits/lexichem-tk_ogham-tk.html (accessed May 17, 2009).

[9]Cf. http://chemnomparse.sourceforge.net/ (accessed May 17, 2009).

programming (CLP) to automatically model so-called constraint satisfaction problems (CSPs) according to given names. Such a CSP captures a name's meaning in that it represents the problem of finding the chemical structure(s) denoted by the name. The solutions to a CSP are determined by a constraint solver. It will find all the structures which satisfy every constraint given by the name. In the case of a fully specified chemical structure, the solution is exactly one structure. This structure is then mapped and classified. For underspecified structures or class names, we distinguish two methods: Either all the structures can be enumerated or the CSP itself can be used for mapping and classification.

Figure 2 shows an overview of the system's architecture. Its component details will be described in the following subsections.

2.1 Parsing and Semantic Representation

We decided to use the CHEMorph parser which is implemented in Prolog. It provides a morpho-semantic grammar which was built according to IUPAC nomenclature rules. The lexicon of this grammar contains the morphemes which can constitute systematic chemical compound names. Also, the lexicon contains a number of trivial and class names. In addition to a syntactic analysis, the CHEMorph parser also yields a semantic representation of the input name. This representation is a term which describes the meaning of the given chemical name in a kind of functor-arguments logic.[10] Example (1), (2) and (3) each show a compound name and its semantic representation generated by CHEMorph:

(1) compound name: *pentan-2,3-diol*
semantic representation: *compd(ane(5*'C'), pref([]), suff([2*[2, 3]-ol]))*

(2) compound name: *2,3-dihydroxy-pentane*
semantic representation: *compd(ane(5*'C'), pref([2*[2, 3]-hydroxy]), suff([]))*

(3) compound name: *propyn-1-imine*
semantic representation: *compd(yne(?? *[??], ane(3*'C')), pref([]), suff([?? *[1]-imine]))*

The general *compd* functor of each semantic representation has three arguments, namely the parent, prefix and suffix representation. The parent argument represents the basic molecular structure, denoted by the parent term of the name. In Example (1) and (2), the parent structure consists of five carbon (C) atoms. This semantic information is encoded with the morpheme *pent* in CHEMorph's lexicon. The parent structure is modified by the functor *ane*, which denotes single bond connections. Prefix and suffix operators, if present, specify further modifications of the basic parent structure. In the case of underspecifying names, as in example (3), the missing pieces of information are represented as *??*.

This way, the semantic representation provides all the information about the chemical structure that is given by the name. Thus, it is an ideal basis for further processing. The next section explains how our system models constraint satisfaction problems on the basis of CHEMorph's semantic representations.

2.2 CSP Modeling

A chemical compound structure can be described as a labeled graph, where the vertices are labeled as atoms and the edges are labeled as bonds. Hence, a chemical compound name can be seen as describing such a graph in that it gives constraints which the graph has to satisfy. In other words, it picks out some specific graph(s) out of the unlimited number of possible graphs in the universe by constraining the possibilities. This observation serves us as a basis for modeling the name-to-structure task as a constraint satisfaction problem (CSP).

A CSP represents a problem as a collection of constraints over a collection of variables. Each of the variables has a domain, which is the set of possible values the variable can take. For the reasons named above, we are working with graph variables and graph domains. The number of chemical compounds, i. e. graphs, could possibly be infinite but we decided it was reasonable and safe to use finite domains. We hence limit the number of possible atoms and bonds for each compound in some way, e. g. on 500 vertices and the corresponding edges or another number estimated according to the semantic representation of the name being processed.

We implement the CSP in ECLiPSe[11], an open-source constraint logic programming (CLP) sys-

[10] Kremer et al. (2006) define the language of the semantic representation in Extended Backus-Naur Form.

[11] Cf. http://eclipse-clp.org/ (accessed May 17, 2009).

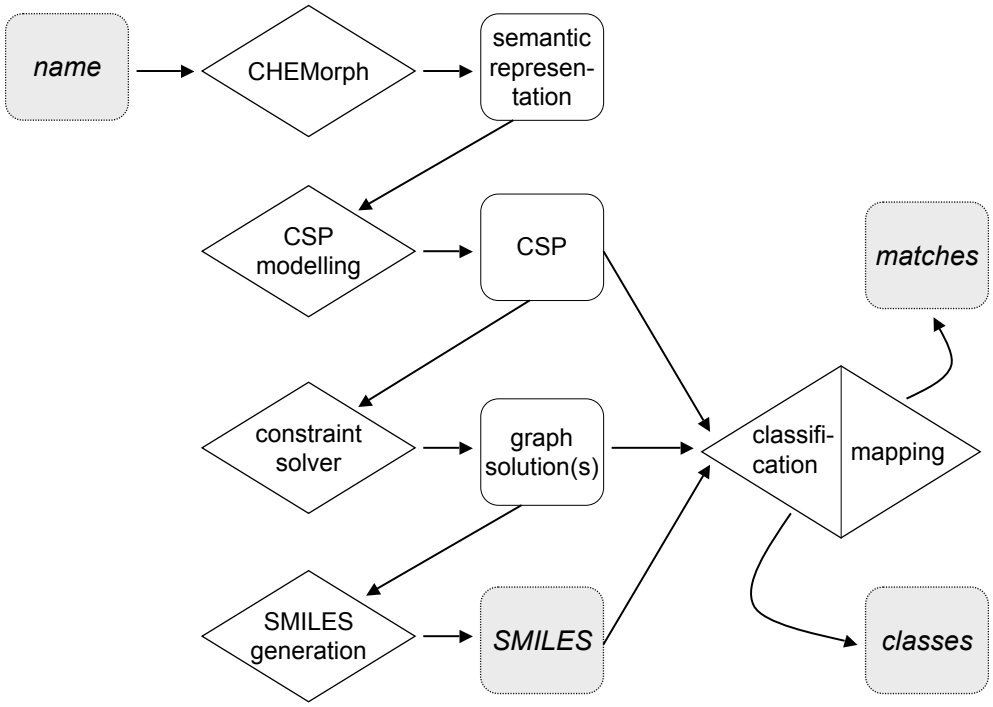

Figure 2: system architecture of CLP(name2structure)

tem, which contains a high-level modeling language, as well as several constraint solver libraries and interfaces for third-party solvers.

To model a CSP for a given input name, several steps have to be taken. First, the semantic representation term provided by CHEMorph has to be parsed. According to its functors and their arguments, the respective constraints have to be called. For this, we are developing a comprehensive set of functions which call the constraints with the correct parameters for the given input name. In these functions, it is determined which constraints over the graph variables a specific functor and argument of the semantic representation is imposing. Thus, in the form of constraints, the functions contain the actions concerning specific functional groups of the denoted molecule, which were described by the name's morphemes. As mentioned in Section 1, these actions include addition, deletion and substitution of certain groups of atoms.

In any case, default rules have to be included while modeling the CSP. Default rules provide constraints about the chemical structures which are not mentioned by any morpheme of the name. For our system they are collected from IUPAC rules as well as from expert knowledge. For example, H-saturation is a default which applies to every chemical compound. This means that every atom of a structure, whose valences are not all occupied by other atoms, has as many H-atoms attached to it as there were free valences. This is one of the reasons why the valences of all the different types of atoms need to be taken into account. We decided to include them as axioms for our models. Knowledge about valences also proves useful for the resolution of underspecification in the case of partial names. Consider a name like *propyn-1-imine* (cf. example (3) in Section 2.1) where it is not specified where the triple bond (denoted by *yn*) is located. However, there are only three C-atoms (introduced by *prop*) to consider, the first of which is connected to an N-atom with a double bond (introduced by *1-imine*). The valence axioms included in our CSPs determine that C-atoms always have a valence of 4, so the first C-atom has only two free valences left until now, since the =N occupies two of them. Consequently, there cannot be a triple bond connected to the same C-atom, as this would use three valences. Hence, the only possibility left is that the triple bond must be located between the second and third C-atom. With the given constraints and axioms, the sys-

tem is thus able to infer the fully specified compound structure of what would correctly have to be named *prop-2-yn-1-imine* (Figure 3).

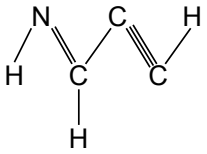

Figure 3: prop-2-yn-1-imine

After modeling a CSP according to the semantic represenation of the input name, the next step in processing is to run a constraint solver. This will be described in the following section.

2.3 Constraint Solver

A constraint solver is a library of tests and operations on constraints. Its purpose is to decide for every conjunction of constraints whether there is a model, i. e. a variable assignment, that satisfies these constraints. This is achieved by consistency checking as well as search techniques, taking the respective variable domains, i. e. the possible values, into account. Besides just deciding whether there is a model for a given CSP, a constraint solver is also able to yield the successful variable assignment(s).

In CLP(name2structure) we use GRASPER[12] (Viegas and Azevedo, 2007), a graph constraint solver based on set constraints. GRASPER enables us to model CSPs using graph variables. In GRASPER, a graph is defined by its set of vertices and its set of edges. Therefore, the domain of a graph consists of a set of possible vertices, in our case for the atoms, and possible edges, in our case for the bonds. The constraints can then narrow these two sets in several ways. For example, certain vertices can be defined to be included as well as the cardinality of a set can be constrained. Also, subgraphs can be defined independently which are then constrained to be part of the final graph solution.

The constraint solver finds one graph solution for graphs which are fully specified by the constraints our system models according to a name. For underspecified graphs, for which the constraints are gathered from underspecifying or class names, the constraint solver could find and enumerate all possible graph solutions if this is desired. This outcome would be the set of all chemical graphs which satisfy the constraints known so far. For example, *chlorohexane* would lead to the set of graphs representing *1-chlorohexane*, *2-chlorohexane* and *3-chlorohexane*.

In general, a chemical name-to-structure system aims at providing the chemical structures in a standard representation format, rather than in a graph notation. In our system, the SMILES generation component carries out this step.

2.4 Generation of a Structural Representation Format

Once a graph is derived from the input name as a solution to its CSP, it specifies the chemical structure completely. It contains the existent vertices and the edges between them, together with labels indicating their respective types and other information like the numbering of atoms. Thus, no additional information has to be considered to generate a chemical representation format from the graph. We focus on generating SMILES strings, rather than some other format, because SMILES themselves use the concept of a graph for representing the molecular structures (Weininger, 1988). For example, the graph solution determined for *pentan-2,3-diol* as well as for *2,3-dihydroxy-pentane* (cf. example (1) and (2) in Section 2.1) can be translated into the SMILES string *CC(OH)C(OH)CC*. In case more than one graph is determined as solution to the CSP (for underspecifying and class names), all the respective SMILES strings could be generated.

Once a SMILES string has successfully been generated, the name-to-structure task is fulfilled and the SMILES string can then be used for tasks such as mapping, classification, picture generation and the like. The next section will describe how classification – one of our main objectives – is accomplished in our approach.

2.5 Classification

Our system offers three different procedures for compound classification. Selection of the appropriate procedure depends on the starting point which could either be a SMILES string, a graph (or a set of graphs) or a CSP.

First, a given SMILES string can be classified based on the functional groups it is comprised of. We use the SMILES classification tool described by Wittig et al. (2004).

[12] GRASPER is distributed with recent builds of the ECLiPSe CLP system.

Second, a graph which is found as solution to a CSP representing an input name can be classified according to a given set of class names. This could for example be some taxonomy which is freely available (like ChEBI (Degtyarenko et al., 2008)). Those class names first have to be transformed into CSPs by use of the parsing and modeling modules of the CLP(name2structure) system. Subsequently, the constraint solver checks whether the graph, or even a set of graphs in the case of an underspecified compound, is a solution to a CSP representing one of the given class names. If the graph or the set of graphs are solutions to one of these CSPs, the compound belongs to the class which provided that CSP. The constraints for the class name *alcohol* for instance, include (amonst others) the presence of an OH-group. Consequently, *pentanol* can be determined to be an alcohol, since its three graph solutions, representing *pentan-1-ol*, *pentan-2-ol* and *pentan-3-ol*, each satisfy the constraints given by *alcohol*.

Third, for some underspecifying names and for class names, it would not be reasonable to generate and classify all the graph solutions or all the SMILES strings – it could simply be too many or even infinitely many. That would slow down performance significantly. Therefore, the system also aims at classifying CSPs themselves, by comparing them directly. If the constraints of CSP-1 are a subset of the constraints of CSP-2, the name which provided CSP-2 is classified to be a hyponym of the more general name which provided CSP-1.

Besides classification, our system aims at mapping chemical compounds. The last module of our system therefore provides algorithms to fulfill this task.

2.6 Mapping

Mapping is needed to fulfill the identification task and to resolve coreference of synonyms. Given a referent data source of chemical compounds, an identity relation should be established if the currently processed compound can successfully be mapped to one of the entries. Again, the procedure depends on whether there is a SMILES string, a set of graph solutions or a CSP to be mapped.

First, matching a SMILES string can be done by simple string comparison. An identity relation between any two compounds holds if their unique SMILES strings (Weininger et al., 1989) match exactly. For example, this is the case for *pentan-2,3-diol* and *2,3-dihydroxy-pentane* since they both yield the same SMILES string (cf. Sections 2.1 and 2.4).

Second, if an underspecifying input name leads to an enumerable number of graph solutions, the set of all the corresponding SMILES strings can be generated. Subsequently, it can be compared to the sets of SMILES strings having been determined for the underspecifying names of the referent data source. If it equals one of the reference SMILES sets, the input name and the respective reference name are successfully identified and thus detected to be synonyms.

Third, mapping of CSPs becomes necessary for class names and underspecifying names with too many graph solutions to enumerate. This works analogously to CSP classification described in Section 2.5 above. The only difference is that a synonym relation between two names, leading to CSP-1 and CSP-2 respectively, is established if the constraints of CSP-1 equal the constraints of CSP-2.

3 Conclusions and Future Work

In this paper we presented the architecture of CLP(name2structure), a system for semantic and syntactic processing of chemical compound names. In the introductory section, we described the characteristic phenomena of biochemical terminology which challenge any such system. Our approach is composed of several modules, carrying out the defined tasks of structure generation, classification and mapping. By employing a morphological parser and constraint logic programming over graph variables, our approach is able to handle the particularities of the chemical compound names.

However, the proposed system CLP(name2structure) still requires work on several of its components. The central task to be completed is to enrich the repository of functions which call the appropriate constraints corresponding to CHEMorph's semantic representation output. This is not a trivial task since it requires to formalize the IUPAC rules of syntax and semantics of the relevant morphemes. This formalization needs to result in an abstract description of the respective constraints over graph variables. Thereby, phenomena like interaction of morphemes' meanings play an important role.

Before we can accomplish the implementation

of the complete system according to the proposed architecture, we need to answer a couple of remaining open questions. For example, the exact method on how to compare two CSPs has to be elaborated. Gennari (2002) describes algorithms for normalizing CSPs to enable subsequent equivalence checking. However, these methods can not be applied to our case as they stand but will have to be substantially adapted. Another problem we need to deal with is that labeled graphs, which are required by our system, are not directly supported by the constraint solver GRASPER. Therefore we are currently working on a way to handle the labels indirectly.

Another important task we plan to carry out in the future is the evaluation of CLP(name2structure). Since no gold standard for name-to-structure generation or classification is available yet, such a gold standard or dataset needs to be created first. We propose to use as such a dataset a subset of the entries of an existing curated database, such as ChEBI, which contains names, chemical structures and a classification for currently 17842 compounds. Unless the morphological parser and the repository of constraint functions is further enriched, we suppose our system will yield a high precision rather than a high coverage. To evaluate underspecification handling of our system, underspecifying names from general reaction descriptions[13] could be collected. For this kind of evaluation, determining the correctness of the analysis would require the help of domain experts.

Acknowledgments

The author is funded by the Klaus Tschira Foundation gGmbH, Heidelberg, Germany. Thanks to Uwe Reyle and Fritz Hamm from the University of Stuttgart, Germany, for contributing to the main ideas and for in-depth discussions. Thanks to the Scientific Databases and Visualization group of EML Research, Heidelberg, Germany, for their support. Thanks to Ruben Viegas for comments on graph constraint solving. Thanks to Berenike Litz and the anonymous reviewers for comments on this paper.

[13] As listet by the Enzyme Nomenclature Recommendations: http://www.chem.qmul.ac.uk/iubmb/enzyme/ (accessed May 17, 2009).

References

IUPAC. Commission on the Nomenclature of Organic Chemistry. 1993. *A Guide to IUPAC Nomenclature of Organic Compounds (Recommendations 1993)*. Blackwell Scientific Publications, Oxford.

Peter Corbett and Peter Murray-Rust. 2006. High-Throughput Identification of Chemistry in Life Science Texts. *CompLife*, pages 107–118.

Kirill Degtyarenko, Paula de Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj, and Michael Ashburner. 2008. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, 36(Database-Issue):344–350.

Peter Ertl. 2003. Cheminformatics Analysis of Organic Substituents: Identification of the Most Common Substituents, Calculation of Substituent Properties, and Automatic Identification of Drug-like Bioisosteric Groups. *Journal of Chemical Information and Computer Science*, 43:374–380.

Rosella Gennari. 2002. *Mapping Inferences. Constraint Propagation and Diamond Satisfaction*. Ph.D. thesis, Universiteit van Amsterdam.

Jung-jae Kim and Jong C. Park. 2004. BioAR: Anaphora Resolution for Relating Protein Names to Proteome Database Entries. In *Proceedings of the Reference Resolution and its Applications Workshop in Conjunction with ACL 2004*, pages 79–86.

Michael Krauthammer and Goran Nenadic. 2004. Term Identification in the Biomedical Literature. *Journal of Biomedical Informatics*, 37(6):512–526.

Gerhard Kremer, Stefanie Anstein, and Uwe Reyle. 2006. Analysing and Classifying Names of Chemical Compounds with CHEMorph. In Sophia Ananiadou and Juliane Fluck, editors, *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine,* Friedrich-Schiller-Universität Jena, Germany, 2006, pages 37–43.

IUPAC. Commission on the Nomenclature of Organic Chemistry. 1979. *Nomenclature of Organic Chemistry, Sections A, B, C, D, E, F and H*. Pergamon Press, Oxford.

Uwe Reyle. 2006. Understanding Chemical Terminology. *Terminology*, 12(1):111–136.

Ruben Viegas and Francisco Azevedo. 2007. GRASPER: A Framework for Graph CSPs. In Jimmy Lee and Peter Stuckey, editors, *Proceedings of the Sixth International Workshop on Constraint Modelling and Reformulation (ModRef'07)*, Providence, Rhode Island, USA.

David Weininger, Arthur Weininger, and Joseph L. Weininger. 1989. SMILES 2. Algorithm for

Generation of Unique SMILES Notation. *Journal of Chemical Information and Computer Science*, 29(2):97–101.

David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36.

Ulrike Wittig, Andreas Weidemann, Renate Kania, Christian Peiss, and Isabel Rojas. 2004. Classification of chemical compounds to support complex queries in a pathway database. *Comparative and Functional Genomics*, 5:156–162.