Ontology-Based Classification of Molecules: a Logic Programming Approach

Despoina Magka

Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, OX1 3QD, UK
despoina.magka@cs.ox.ac.uk

Abstract. We describe a prototype that performs structure-based classification of molecular structures. The software we present implements a sound and complete reasoning procedure of a formalism that extends logic programming and builds upon the DLV deductive databases system. We capture a wide range of chemical classes that are not expressible with OWL-based formalisms such as cyclic molecules, saturated molecules and alkanes. In terms of performance, a noticeable improvement is observed in comparison with previous approaches. Our evaluation has discovered subsumptions that are missing from the the manually curated ChEBI ontology as well as discrepancies with respect to existing subclass relations. We illustrate thus the potential of an ontology language which is suitable for the Life Sciences domain and exhibits an encouraging balance between expressive power and practical feasibility.

Keywords: Knowledge representation and reasoning, Logic programming and answer set programming, Cheminformatics.

1 Introduction

The volume of bioinformatics data produced by research laboratories worldwide is increasing at an astonishing rate turning the need to adequately catalogue, represent and index the vast amounts of Life Sciences data sources into a pressing challenge. Semantic technologies have achieved significant progress towards the federation of biochemical information [33, 3, 4] via the definition and use of domain vocabularies with formal semantics, also known as *ontologies*. OWL [15], a family of logic-based knowledge representation (KR) formalisms standardised by W3C, has played a pivotal role in the advent of Semantic technologies due to its significant ability to reason over ontologies by means of logical inference. In particular, OWL bio- and chemo-ontologies with their intuitive hierarchical structure and their formal semantics are widely used for the modelling of Life Sciences terminologies.

Classification is a core activity in biochemical investigations, as hierarchies come with a number of compelling benefits. For instance, hierarchically organised knowledge is more accessible to humans as it is indicated, e.g. by the widespread use of the periodic table in chemistry. Additionally, organising a large number of different objects into meaningful groups facilitates the discovery of significant properties pertaining to the group; these discoveries can then be used to predict features of later detected members

of the group, such as the high volatility of esters with low molecular weight. As a consequence, classifying objects on the basis of shared characteristics is a central task in areas such as biology and chemistry with a long tradition of taxonomy use. Due to the availability of performant OWL reasoners, life scientists can employ OWL to encode expert human knowledge and thus drive fast, automatic and repeatable classification processes that produce high quality hierarchies [32, 5]. Nevertheless, a prerequisite is that OWL is expressive enough to model the entities that need to be classified as well as the properties of the superclasses that lie higher up in the hierarchy.

As discussed in our previous work [21], we identify two main restrictions in the expressive power of OWL which have also been highlighted in the past as hindering factors for the representation of biological knowledge with OWL [24]. First, due to the tree-model property of OWL [28] (which accounts for the robust computational properties of the language) one is not able to describe cyclic structures in OWL with adequate precision. Second, because of the open-world assumption adopted in OWL (according to which missing information is treated as *not known* rather than *false*) it is difficult to define classes based on the absence of certain characteristics. These limitations manifest themselves—among others—via the inability to define a broad range of classes in the chemical domain. For instance, one cannot encode effectively in OWL the class of compounds that contain a benzene ring or the category of molecules that do not contain carbon atoms, i.e. inorganic molecules.

These inadequacies obstruct the full automation of the classification process for chemical ontologies, such as the ChEBI ontology [22]. ChEBI is an open-access dictionary of molecular entities that provides high quality annotation and taxonomical information for chemical compounds. The ChEBI ontology fosters interoperability between researchers by acting as a common reference and supports tasks such as the study of metabolic networks, identification of disease pathways and pharmaceutical design [14, 10]. However, ChEBI is manually curated by human experts who annotate and determine the chemical classes of new molecular entries. Currently, ChEBI describes 29,295 fully annotated entities (release 95[1]) and grows at a rate of approximately 3,500 entities per year (estimate based on previous releases[1]). Given the size of other publicly available chemical databases, such as PubChem [30] that contains records for 1.6 million molecules and ChemSpider [25] that encompasses more than 26 million unique molecules, there is clearly a strong potential for ChEBI to expand by speeding up the curating tasks through automation of chemical classification.

The construction of chemical hierarchies has been the topic of various investigations capitalising on both logic-based KR [29, 17, 12, 9] and statistical machine learning (ML) [16, 8] techniques. In KR approaches, molecule and class descriptions are represented with logical axioms crafted by experts and subsumptions are identified with the help of automated reasoning algorithms; in ML approaches a set of annotated data is used to train an algorithm and the algorithm is next employed to classify new entries. So, KR approaches are based on the explicit axiomatisation of knowledge, whereas ML algorithms act as a 'black box' that assigns to new entries superclasses that are highly probable to be correct. As a consequence, the taxonomies produced using logic-based techniques are provably correct (as long as the modelling of the domain knowledge is

[1] `http://www.ebi.ac.uk/chebi/newsForward.do`

faithful), but the statistically produced hierarchies (although much faster) need to be evaluated against a curated gold standard. Thus, KR-based and ML-based classification techniques are not directly comparable. For a more detailed overview, Hastings et al. [13] provide a thorough analysis of the two approaches.

Here, we focus on logic-based chemical classification. In our previous work [21], we laid the theoretical foundation of a new sound and complete expressive ontology language under the name DGLP (Description Graph Logic Programs) that is suitable for the representation of graph-shaped objects; additionally, we demonstrated how DGLP, which draws upon logic programming, can be applied to the classification of molecules. DGLP addressed the expressivity limitations outlined above; however, the performance of the implementation—although faster than previous approaches—was not satisfactory (more than 7 minutes were needed to classify 70 molecules under 5 chemical classes on a standard desktop computer) failing thus to confirm practicability of the formalism.

In the current work, we describe an improved practical framework that relies on the same formalism but with ameliorated performance. In particular, our contributions can be summarised as follows:

1. We present a prototype that performs logic-based chemical classification based on a sound, complete and terminating reasoning algorithm; we model more than 50 chemical classes and we show that the superclasses of 500 molecules are computed in 40 seconds.
2. We harness the expressive power of logic programming to axiomatise a variety of chemical classes such as classes based on the containment of functional groups (e.g. esters) and on the exact cardinality of parts (e.g. dicarboxylic acids), classes depending on the overall atomic constitution (e.g. saturated molecules) and cyclicity-related classes (e.g. compounds containing a cycle of arbitrary length or alkanes).
3. We exhibit a significant speedup in comparison with previous ontology-based chemical classification implementations.
4. We identify examples of missing and contradictory subsumptions from the manually curated ChEBI ontology that are present and absent, respectively, from the hierarchy computed by our prototype.

Concerning future benefits, our prototype could form the basis of a Semantic Web application to assist biocurators of the ChEBI ontology towards the sanitisation and the enrichment of the existing chemical taxonomy. Similarly, such a tool could contribute to a more rapid development of the ChEBI ontology and to the efforts of the ChEBI team to make annotated chemical datasets available to the public. From a modelling point of view, our approach could stimulate the adoption of a different and expressive reasoning paradigm based on logic programming for which state-of-the-art and highly optimised reasoners are available; it could thus pave the way for the representation of a broader spectrum of Life Sciences and biomedical knowledge.

2 Modelling Chemicals with Logic Programming

The reasoning task carried out by our methodology is the identification of chemical classes for molecules, e.g. the class of inorganic molecules for water or cyclic molecules

for benzene. In this section we provide a high-level description of the knowledge base (KB) we built for the purposes of the said structure-based chemical classification. By abuse of terminology, we use the word 'classification' to refer to the detection of subsumptions between molecules and chemical classes rather than to the computation of the partial order of the set of chemical classes and molecules w.r.t. the subclass relation. The KB consists of logic programming (LP) rules that formally describe molecular structures and chemical classes; this representation can subsequently be used to determine the subsumers of each molecule in terms of chemical classes. Since our main focus is to illustrate the transformation of chemical graphs and chemical class definitions into LP rules, we omit the technical details and describe our setting by means of a running example. For a formal definition of syntax and semantics as well as decidability proofs, we refer the interested reader to the relevant articles [21, 6].

2.1 Molecular Structures

Next, we describe how a chemical graph can be converted into an LP rule that encodes its structure. We use as an example the molecule of ascorbic acid, a naturally occurring organic compound commonly known as vitamin C. The chemical graph of ascorbic acid is depicted in the top right corner of Figure 1.

Conceptually, the structure of ascorbic acid can be abstracted with the help of a directed labeled graph such as the one that appears in the bottom right of Figure 1 and which in our framework is called *description graph* (DG) [21]. In order to simplify the depiction of the ascorbic acid DG in Figure 1 a legend is used for the edge labels; all arrowless edges are assumed to be bidirectional. The description graph of a molecule is a labeled graph whose nodes correspond to the atoms of the molecule (nodes 1–13 for ascorbic acid) plus an extra node for the molecule itself (node 0) and whose edges correspond to the bonds of the molecule (e.g. (1,7)) plus some additional edges that connect the molecule node with each one of the atom nodes (e.g. (0,1)); additionally, the atom nodes are labeled with the respective chemical elements (e.g. o for node 1) and the bond edges with the corresponding bond order (e.g. single for (1,7)); finally, the molecule node is labeled with molecule and the edges that connect the molecule node with each of the atom nodes are labeled with hasAtom. In our setting, we follow the implicit hydrogen assumption according to which hydrogen atoms are usually suppressed (excluding cases where stereochemical information is provided for the formed bond as in node 13). Finally, we point out that both the nodes and the edges can have multiple labels allowing us to also encode molecular properties such as charge values for atoms.

The description graph of ascorbic acid can next be converted into an LP rule that faithful represents its molecular structure (in fact we need a separate rule for each conjunct in the head but we use just one rule here to simplify the presentation). The LP rule that the DG of ascorbic acid is translated into is as follows (for the sake of brevity only one direction of the bonds appears in the rest of the text and we shorten an expression

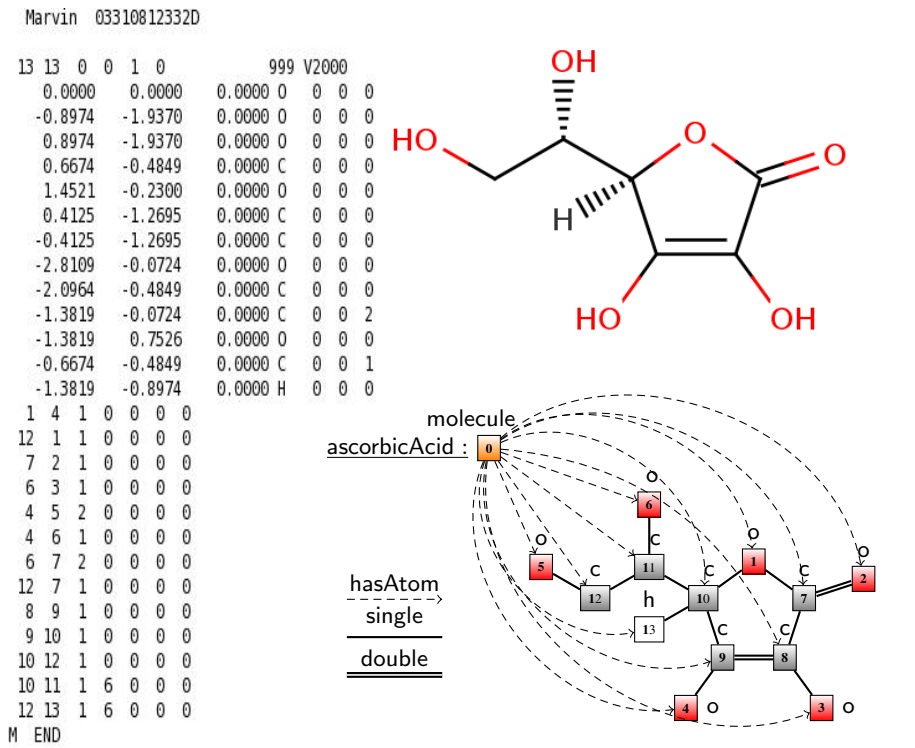

Fig. 1. Molfile (left), chemical graph (top right) and description graph (bottom right) encoding the molecular structure of ascorbic acid.

of the form $\wedge C_1 \ldots \wedge C_n$ with $\wedge_{i=1}^{n} C_i$):

$$\begin{aligned} \mathsf{ascorbicAcid}(x) \rightarrow\ & \wedge_{i=1}^{13}\, \mathsf{hasAtom}(x, f_i(x)) \wedge \mathsf{molecule}(x) \wedge_{i=1}^{6}\, o(f_i(x)) \wedge_{i=7}^{12}\, c(f_i(x)) \wedge \\ & h(f_{13}(x)) \wedge \mathsf{single}(f_8(x), f_3(x)) \wedge \mathsf{single}(f_9(x), f_4(x)) \\ & \wedge_{i=1,9,11,13}\, \mathsf{single}(f_{10}(x), f_i(x)) \wedge_{i=5,11}\, \mathsf{single}(f_{12}(x), f_i(x)) \\ & \wedge_{i=1,8}\, \mathsf{single}(f_7(x), f_i(x)) \wedge \mathsf{single}(f_{11}(x), f_6(x)) \wedge \\ & \mathsf{double}(f_2(x), f_7(x)) \wedge \mathsf{double}(f_8(x), f_9(x)) \end{aligned}$$

The rule above is a typical first-order implication with a single atomic formula in the body and a conjunction of atomic formulae in the head. Informally, the rule ensures that every time that an ascorbic acid molecule is encountered in the KB, its structure is unfolded according to its specified DG. Thus, triggering of the rule implies that (i) new terms that correspond to the DG's nodes are generated (excluding node 0), e.g. $f_1(x)$ represents atom node 1 (ii) each new term is typed according to the label of the relevant node with the help of a unary atomic formula (e.g. $o(f_1(x))$) and (iii) each pair of terms

with corresponding nodes connected in the DG is assigned the respective label with the help of a binary atomic formula (e.g. $\mathsf{single}(\mathsf{f}_1(\mathsf{x}), \mathsf{f}_7(\mathsf{x}))$). In order to ensure disjointness of the several molecular structures on the interpretation level, distinct function symbols are used in the LP rule of each molecule.

2.2 Background Knowledge and Chemical Classes

Before presenting the modelling of various chemical classes, we demonstrate how LP rules can encode background chemical knowledge, e.g. the fact that single and double bonds are kinds of bonds or that atoms with positive or negative charge are charged; LP rules may also denote a particular class of atoms based on their elements, e.g. atoms that are hydrogens or carbons:

$$\mathsf{single}(\mathsf{x},\mathsf{y}) \rightarrow \mathsf{bond}(\mathsf{x},\mathsf{y}) \qquad \mathsf{double}(\mathsf{x},\mathsf{y}) \rightarrow \mathsf{bond}(\mathsf{x},\mathsf{y})$$
$$\mathsf{negative}(\mathsf{x}) \rightarrow \mathsf{charged}(\mathsf{x}) \qquad \mathsf{positive}(\mathsf{x}) \rightarrow \mathsf{charged}(\mathsf{x})$$
$$\mathsf{h}(\mathsf{x}) \rightarrow \mathsf{horc}(\mathsf{x}) \qquad \mathsf{c}(\mathsf{x}) \rightarrow \mathsf{horc}(\mathsf{x})$$

For our experiments, we represented 51 chemical classes using LP rules; we based our chemical modelling on the textual definitions found in the ChEBI ontology [22]. We covered a diverse variety of classes that can roughly be categorised into four groups. Please note that there are cases in which more than one LP rule is needed to encode a class definition. Due to space restrictions, we show in full detail only a sample of the rules; the complete logic program is available online.[2]

Existence of subcomponents The great majority of the modelled chemical classes is defined via containment of atoms, functional groups or other atom arrangements. Examples of this type include carbon molecular entities, halogens, molecules that contain a benzene ring, carboxylic acids, carboxylic esters, polyatomic entities, amines, aldehydes and ketones.

$$\mathsf{molecule}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x},\mathsf{y}) \wedge \mathsf{c}(\mathsf{y}) \rightarrow \mathsf{carbonMolEntity}(\mathsf{x})$$
$$\mathsf{molecule}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x},\mathsf{y}_1) \wedge \mathsf{hasAtom}(\mathsf{x},\mathsf{y}_2) \wedge \mathsf{y}_1 \neq \mathsf{y}_2 \rightarrow \mathsf{polyatomicEntity}(\mathsf{x})$$
$$\wedge_{\mathsf{i}=1}^{3}\mathsf{hasAtom}(\mathsf{x},\mathsf{y}_\mathsf{i}) \wedge \mathsf{o}(\mathsf{y}_1) \wedge_{\mathsf{i}=2}^{3} \mathsf{bond}(\mathsf{y}_1,\mathsf{y}_\mathsf{i}) \wedge \mathsf{y}_2 \neq \mathsf{y}_3 \rightarrow \mathsf{middleOxygen}(\mathsf{y}_1)$$
$$\mathsf{molecule}(\mathsf{x}) \wedge_{\mathsf{i}=1}^{4} \mathsf{hasAtom}(\mathsf{x},\mathsf{y}_\mathsf{i}) \wedge \mathsf{c}(\mathsf{y}_1) \wedge \mathsf{o}(\mathsf{y}_2) \wedge \mathsf{o}(\mathsf{y}_3) \wedge$$
$$\mathsf{horc}(\mathsf{y}_4) \wedge \mathsf{double}(\mathsf{y}_1,\mathsf{y}_2) \wedge \mathsf{single}(\mathsf{y}_1,\mathsf{y}_3) \wedge \mathsf{single}(\mathsf{y}_1,\mathsf{y}_4) \wedge$$
$$\mathbf{not}\ \mathsf{middleOxygen}(\mathsf{y}_3) \wedge\ \mathbf{not}\ \mathsf{charged}(\mathsf{y}_3) \rightarrow \mathsf{carboxylicAcid}(\mathsf{x})$$
$$\mathsf{molecule}(\mathsf{x}) \wedge_{\mathsf{i}=1}^{5} \mathsf{hasAtom}(\mathsf{x},\mathsf{y}_\mathsf{i}) \wedge_{\mathsf{i}=1,4} \mathsf{c}(\mathsf{y}_\mathsf{i}) \wedge_{\mathsf{i}=2,3} \mathsf{o}(\mathsf{y}_\mathsf{i}) \wedge$$
$$\mathsf{horc}(\mathsf{y}_5) \wedge \mathsf{double}(\mathsf{y}_1,\mathsf{y}_2) \wedge_{\mathsf{i}=3,5} \mathsf{single}(\mathsf{y}_1,\mathsf{y}_\mathsf{i}) \wedge \mathsf{single}(\mathsf{y}_3,\mathsf{y}_4) \rightarrow \mathsf{carboxylicEster}(\mathsf{x})$$

We show above the rules for the classes of carbon molecular entities, polyatomic entities, carboxylic acids and esters. We define as carbon molecular entities the molecules

[2] `http://www.cs.ox.ac.uk/isg/people/despoina.magka/tools/chemRules`

that contain carbon; polyatomic entities are the entities that contain at least two different atoms. Carboxylic acids are defined as molecules containing at least one carboxy group (a functional group with formula C(=O)OH) attached to a carbon or hydrogen; due to the implicit hydrogens assumption we are not able to distinguish between an oxygen and a hydroxy group and, so, we need to specify that the oxygen of the hydroxy group is not charged ($\mathbf{not}\ \mathsf{charged}(\mathsf{y}_3)$) and participates to only one bond ($\mathbf{not}\ \mathsf{middleOxygen}(\mathsf{y}_3)$). Similarly, carboxylic esters contain a carbonyl group connected to an oxygen ((C=O)O) which is further attached to two atoms that are carbon or hydrogen.

Exact cardinality of parts These chemical classes comprise molecules with an exact number of atoms or of functional groups. Examples include molecules that contain exactly two carbons, molecules that contain only one atom and dicarboxylic acids, that is molecules with exactly two carboxy groups. We show the definition of molecules with exactly two carbons.

$$\mathsf{molecule}(\mathsf{x}) \wedge \wedge_{\mathsf{i}=1}^{2} \mathsf{hasAtom}(\mathsf{x}, \mathsf{y}_\mathsf{i}) \wedge \mathsf{c}(\mathsf{y}_\mathsf{i}) \wedge \mathsf{y}_1 \neq \mathsf{y}_2 \rightarrow \mathsf{atLeast2Carbs}(\mathsf{x})$$
$$\mathsf{molecule}(\mathsf{x}) \wedge \wedge_{\mathsf{i}=1}^{3} \mathsf{hasAtom}(\mathsf{x}, \mathsf{y}_\mathsf{i}) \wedge \mathsf{c}(\mathsf{y}_\mathsf{i}) \wedge \wedge_{\mathsf{i}=2}^{3} \mathsf{y}_1 \neq \mathsf{y}_\mathsf{i} \wedge \mathsf{y}_2 \neq \mathsf{y}_3 \rightarrow \mathsf{atLeast3Carbs}(\mathsf{x})$$
$$\mathsf{atLeast2Carbs}(\mathsf{x}) \wedge\ \mathbf{not}\ \mathsf{atLeast3Carbs}(\mathsf{x}) \rightarrow \mathsf{exactly2Carbs}(\mathsf{x})$$

Exclusive composition These are classes of molecules such that each atom (or bond) they contain satisfies a particular property. E.g. inorganic molecules consist exclusively of non-carbon atoms,[3] hydrocarbons only contain hydrogens and carbons and saturated compounds are defined as the compounds whose carbon to carbon bonds are all single. Hydrocarbons are defined as follows:

$$\mathsf{molecule}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x}, \mathsf{y}) \wedge\ \mathbf{not}\ \mathsf{c}(\mathsf{y}) \wedge\ \mathbf{not}\ \mathsf{h}(\mathsf{y}) \rightarrow \mathsf{notHydroCarbon}(\mathsf{x})$$
$$\mathsf{carbonMolEntity}(\mathsf{x}) \wedge\ \mathbf{not}\ \mathsf{notHydroCarbon}(\mathsf{x}) \rightarrow \mathsf{hydroCarbon}(\mathsf{x})$$

Cyclicity-related classes These chemical classes include the definition of molecules containing a ring of any length as well as other definitions that depend on the cyclicity of molecules (for instance alkanes). Assuming the (somewhat technical) definition of cyclic and of saturated molecules, we provide the definition of alkanes.

$$\mathsf{molecule}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x}, \mathsf{y}) \wedge \mathsf{loopAtLeast3Atom}(\mathsf{y}) \rightarrow \mathsf{cyclic}(\mathsf{x})$$
$$\mathsf{saturated}(\mathsf{x}) \wedge \mathsf{hydroCarbon}(\mathsf{x}) \wedge\ \mathbf{not}\ \mathsf{cyclic}(\mathsf{x}) \rightarrow \mathsf{alkane}(\mathsf{x})$$

2.3 Determining Subclass Relations

Finally, we demonstrate how meaningful subsumptions can be derived using a KB containing the rules outlined in the previous two sections. In order to determine the superclasses of a certain molecule, we extend the KB with a suitable fact (i.e., a variable-free

[3] Inspite of the fact that there are many compounds with carbons considered inorganic, we aligned our encoding to the ChEBI definition (CHEBI:24835) according to which inorganic molecular entities contain no carbons.

atomic formula) and we examine the model that satisfies the KB under the *stable model semantics*. An exact definition of the stable model semantics is provided by Gelfond and Lifschitz [11]. Intuitively, the stable model of a KB is the minimal set of facts that are derived by exhaustively applying the existing rules under a particular rule order; a rule is applied if its positive body can be matched to the so far derived facts and no atom of the negative body is in the already produced set of facts for the said matching.

Table 1. Stable model of the KB with the input fact $\mathsf{ascorbicAcid}(\mathsf{a})$ and the rules of Section 2.1 and 2.2; $\mathsf{f}_\mathsf{i}(\mathsf{a})$ is abbreviated with $\mathsf{a}_\mathsf{i}^\mathsf{f}$ for $1 \leq \mathsf{i} \leq 13$.

Input fact: $\mathsf{ascorbicAcid}(\mathsf{a})$
Stable model: $\mathsf{ascorbicAcid}(\mathsf{a})$, $\mathsf{hasAtom}(\mathsf{a}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $1 \leq \mathsf{i} \leq 13$, $\mathsf{o}(\mathsf{a}_\mathsf{i}^\mathsf{f})$ for $1 \leq \mathsf{i} \leq 6$, $\mathsf{c}(\mathsf{a}_\mathsf{i}^\mathsf{f})$ for $7 \leq \mathsf{i} \leq 12$, $\mathsf{h}(\mathsf{a}_{13}^\mathsf{f})$, $\mathsf{single}(\mathsf{a}_8^\mathsf{f}, \mathsf{a}_3^\mathsf{f})$, $\mathsf{single}(\mathsf{a}_9^\mathsf{f}, \mathsf{a}_4^\mathsf{f})$, $\mathsf{single}(\mathsf{a}_{12}^\mathsf{f}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $\mathsf{i} \in \{5, 11\}$, $\mathsf{single}(\mathsf{a}_{10}^\mathsf{f}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $\mathsf{i} \in \{1, 9, 11, 13\}$, $\mathsf{single}(\mathsf{a}_7^\mathsf{f}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $\mathsf{i} \in \{1, 8\}$, $\mathsf{single}(\mathsf{a}_{11}^\mathsf{f}, \mathsf{a}_6^\mathsf{f})$, $\mathsf{double}(\mathsf{a}_2^\mathsf{f}, \mathsf{a}_7^\mathsf{f})$, $\mathsf{double}(\mathsf{a}_8^\mathsf{f}, \mathsf{a}_9^\mathsf{f})$, $\mathsf{bond}(\mathsf{a}_8^\mathsf{f}, \mathsf{a}_3^\mathsf{f})$, $\mathsf{bond}(\mathsf{a}_9^\mathsf{f}, \mathsf{a}_4^\mathsf{f})$, $\mathsf{bond}(\mathsf{a}_{12}^\mathsf{f}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $\mathsf{i} \in \{5, 11\}$, $\mathsf{bond}(\mathsf{a}_{10}^\mathsf{f}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $\mathsf{i} \in \{1, 9, 11, 13\}$, $\mathsf{bond}(\mathsf{a}_7^\mathsf{f}, \mathsf{a}_\mathsf{i}^\mathsf{f})$ for $\mathsf{i} \in \{1, 8\}$, $\mathsf{bond}(\mathsf{a}_{11}^\mathsf{f}, \mathsf{a}_6^\mathsf{f})$, $\mathsf{bond}(\mathsf{a}_2^\mathsf{f}, \mathsf{a}_7^\mathsf{f})$, $\mathsf{bond}(\mathsf{a}_8^\mathsf{f}, \mathsf{a}_9^\mathsf{f})$, $\mathsf{horc}(\mathsf{a}_\mathsf{i}^\mathsf{f})$ for $7 \leq \mathsf{i} \leq 13$, $\mathsf{molecule}(\mathsf{a})$, $\mathsf{carbonMolEntity}(\mathsf{a})$, $\mathsf{polyatomicEntity}(\mathsf{a})$, $\mathsf{middleOxygen}(\mathsf{a}_1^\mathsf{f})$, $\mathsf{carboxylicEster}(\mathsf{a})$, $\mathsf{atLeast2Carbons}(\mathsf{a})$, $\mathsf{atLeast3Carbons}(\mathsf{a})$, $\mathsf{notHydroCarbon}(\mathsf{a})$, $\mathsf{cyclic}(\mathsf{a})$

The initially added fact is the molecule name predicate instantiated with a fresh constant so that the rule that encodes the DG of that molecule is triggered. For the case of ascorbic acid, if we append the fact $\mathsf{ascorbicAcid}(\mathsf{a})$ to the previously described KB, we obtain the stable model that appears in Table 1. From the highlighted atoms we can infer the superclasses of ascorbic acid, that is we deduce that ascorbic acid is—among others—a polyatomic, cyclic molecular entity that contains carbon and a carboxylic ester. If there is no relevant atom for a chemical class in the stable model, then we conclude that the said class is not a valid subsumer, e.g. since $\mathsf{carboxylicAcid}(\mathsf{a})$ is not found in the stable model, carboxylic acid is not a superclass of ascorbic acid.

2.4 Decidability check

The KB discussed above contains rules with function symbols in the head, such as the rule used to encode the molecular structure of ascorbic acid. These rules may incur non-termination during the computation of the stable model due to the creation of new terms that might be infinitely many. In order to ensure termination of our reasoning process and thus decidability of the employed formalism, we perform a *decidability* check on the constructed KB. Roughly, the decidability check (also known and as *model-summarising acyclicity* [6]) involves transforming the rules of the KB and inspecting the stable model of the transformed KB. If the KB passes the decidability check, then termination is guaranteed. Technical details of the aforementioned condition are out of the scope of this text and can be found in the relevant sources [6].

3 Prototype Implementation

The current section provides an overview of LoPStER,[4] the prototype we developed for structure-based chemical classification. The implementation is heavily based on DLV

[4] **Lo**gic **P**rogramming for **St**ructured **E**ntities **R**easoner

system, a powerful and efficient deductive databases and logic programming engine [19]. DLV constitutes the automated reasoning component used by LoPStER for stable model computation of a set of LP rules. Figure 2 depicts the basic processing steps as well as the different files that are processed and produced by LoPStER. LoPStER is implemented in Java and is available online.[5] Next, we describe in more detail the several stages of execution.

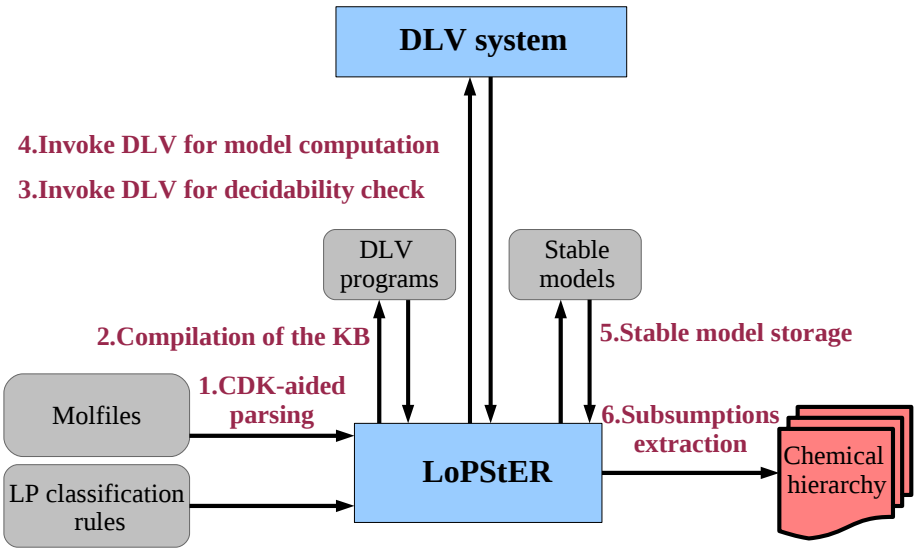

Fig. 2. Architecture of DLVStructuredEntities

1. **CDK-aided parsing.** LoPStER parses the molfiles [7] of the molecules to be classified using the Chemistry Development Kit Java library [27]. The molfile is a widely used chemical file format that describes molecular structures with a connection table; e.g., the molfile of ascorbic acid appears in the left of Fig. 1. For each molecule, a description graph (e.g. Fig. 1 bottom right) representation is generated from its molfile according to a transformation as the one described for ascorbic acid.
2. **Compilation of the KB.** For each molecule the description graph representation is used to produce a set of LP rules that encode the structure of the molecule, following the translation that was discussed in Section 2.1. These rules along with the LP classification rules (Section 2.2) and the facts necessary to determine subclass relations (as described in Section 2.3) are combined to produce DLV programs (i.e. sets of LP rules) that are stored as plain text files on disk. In particular two kinds of DLV programs are created for each molecule, the program needed to perform the

[5] `http://www.cs.ox.ac.uk/isg/people/despoina.magka/tools/LoPStER.zip`

decidability check as described in Section 2.4 and the program needed to compute subclass relations between the molecules and the chemical classes.
3. **Invoke DLV for decidability check.** During this step, the model of the program, which was produced in the previous step for acyclicity testing, is computed. If the check is successful, then execution proceeds to the next stage; otherwise, the program is exited with a suitable output message.
4. **Invoke DLV for model computation.** This is the stage where DLV is invoked to compute the stable model of the KB. Due to the check of the previous step, the computation is guaranteed to terminate.
5. **Stable model storage.** At this point, the stable model computed by DLV is stored in a file on disk to enable subsequent discovery of the subclass relations.
6. **Subsumptions extraction.** This is the final phase where the stable model file is parsed in order to detect the superclasses of each molecule. All the subsumee-subsumer pairs are stored in a separate spreadsheet file on disk.

4 Empirical Evaluation

In order to assess the applicability of our implementation, we measured the time required by LoPStER to perform classification of molecules. To obtain test data we extracted molfile descriptions of 500 molecules from the ChEBI ontology. The represented compounds were of diverse size, varying from a few atoms to less or equal than 59 atoms. Next, we investigated the scalability of our prototype by altering two different parameters of the knowledge base, namely the number of represented molecules and the type of modelled chemical classes. Initially, we constructed 10 DLV programs each of which contained rules encoding $50 \cdot i$ different compounds, where $1 \leq i \leq 10$, and rules defining the chemical classes previously described excluding the cyclicity-related classes (48 classes in total). Next, we repeated the same construction but this time including the rules for the cyclicity-related classes (51 classes). In the rest of the section, we refer to the first setting as 'no cyclic' and to the second as 'with cyclic'.

Table 2. Experimental results

No molecules	No of rules	Time no cyclic (sec)	Time with cyclic (sec)
50	3614	3.21	4.69
100	6832	3.11	5.75
150	18072	8.06	19.98
200	23746	10.27	25.08
250	28502	12.41	29.87
300	31892	14.34	31.63
350	35046	14.44	34.05
400	38095	15.97	35.99
450	41536	17.36	37.97
500	43629	19.14	39.66

Additionally and in order to optimise the performance, we explored how classification times fluctuate depending on the size of DLV programs. In particular, we par-

titioned the DLV programs into modules, we measured classification times for each module separately and we summed up the times. Each module contains the facts and the rules describing a subset of the molecules represented in the initial DLV program; the rules defining chemical classes are appended to each one of the modules. Thus, the size of each module depends on the number of encoded molecules. We tested modules of size 5, 10, 20, 25 and 50 as well as DLV programs without any partitioning for both the 'no cyclic' mode and the 'with cyclic' mode. Modifying the size of the module had a clear impact on the measured times and performing classification with the modularised knowledge base was always quicker than with the unpartitioned one; we observed the shortest execution times for module size 50 when testing in 'no cyclic' mode and for module size 25 when testing in 'with cyclic' mode.

Table 2 summarises the classification times for the previously described KBs. The experiments were performed on a desktop computer with 3.7 GB of RAM and Intel CoreTM 2 Quad Processors running at 2.5 GHz and 64 bit Linux. The first column displays the number of molecules, the second column the number of LP rules contained in the corresponding DLV program and the third (fourth) column the time needed to perform classification in 'no cyclic' ('with cyclic') mode. We only display the number of LP rules for the 'no cyclic' mode because there are only six rules more in the DLV programs with cyclicity-related definitions. The times that appear in the third and fourth column of Table 2 were measured for module size 50 and 25 respectively. All the DLV programs that were tested passed the decidability check. The classification experiment for each knowledge base was repeated three times and the results were averaged over the three runs; also, the durations of Table 2 are comprehensive, that is they count the time elapsed before the molfiles parsing and after the subsumptions extraction. Figure 3 depicts the plots of the time intervals appearing in Table 2 both with regard to the number of molecules and the number of rules contained in the respective DLV program.

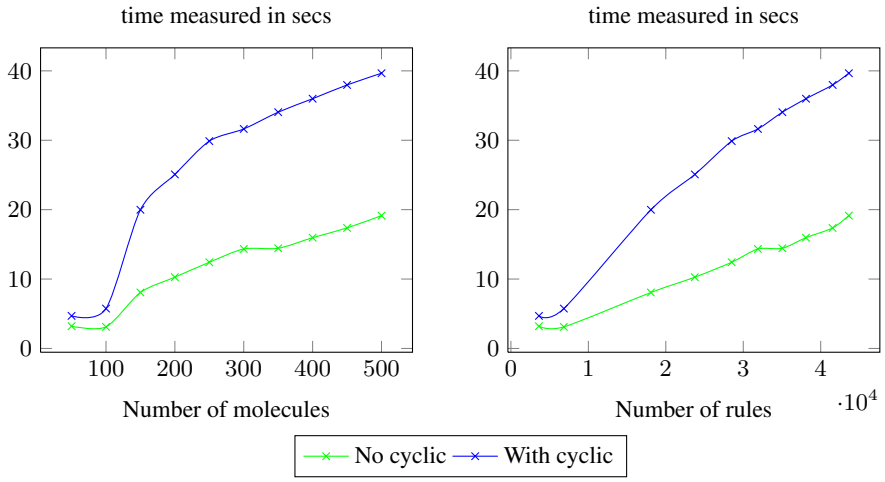

Fig. 3. Curves of classification times

5 Discussion and Related Work

The performance results of Table 2 are encouraging for the practical feasibility of our approach: the time measurements for 500 molecules suggest that the entire set of molecular entities that are currently represented by the ChEBI ontology (29,295 3-star entities as of release 95) could be classified in less than 40 minutes for the suite of 51 modelled chemical classes. One can observe that the rules encoding cyclicity-related classes introduce a significant overhead for the classification times. In fact, it is the class that recognises molecules with cycles of arbitrary length that incurs the performance penalty. The rules that encode the class of cyclic molecules need to identify patterns that are extremely frequent in molecular graphs; as a consequence, the amount of computational resources that is needed to detect ring-containing molecules is much higher. However, since our class definition for cyclic molecules detects compounds with cycles of variable length which is a significant property for the construction of chemical hierarchies, we consider this overhead acceptable.

Concerning expressive power, the current approach allows for the representation of strictly more chemical classes in comparison with other logic-based applications for chemical classification. Villanueva-Rosales and Dumontier [29] were the first to describe an OWL DL ontology of functional groups for the classification of chemical compounds; in their work, they point out the inherent inability of OWL to represent cycles and how this hinders the use of OWL in logic-based chemical classification. As a remedy, Hastings et al. [12] employ an extension of OWL [23] for the representation of non-tree-like structures and, thus, for the classification of molecular structures. However, the used formalism only allows for the identification of cycles of fixed length and with alternating single and double bonds. In the current approach we are able to recognise molecules containing cycles of both arbitrary and fixed length and without requiring a particular configuration of bonds.

Moreover, in both approaches outlined above the adopted open world assumption of OWL prevents one from defining structures based on the absence of certain characteristics. In our approach we operate under the closed world assumption which permits for the definition of a broad range of chemical classes that were not expressible before such as the class of inorganic, hydrocarbon or saturated compounds. Finally and in comparison with our previous work [21], we take full advantage of the suggested formalism by specifying a much wider range of chemical classes and we do not require from the modeller a precedence relation between the represented structures.

In terms of performance, the classification results appear more promising than previous and related work. Hastings et al. [12] report that a total of 4 hours was required to determine the superclasses of 140 molecules, whereas LoPStER identifies the chemical classes of 500 molecules in less than 40 seconds. LoPStER is quicker in comparison with our previous work too [21] where 450 seconds were needed to classify 70 molecules (80 times faster). Please note that both cases discussed above considered a subset of the chemical classes used here. We identify the following two main factors for the significant change in speed. First, DLV is a more suitable reasoner for our setting due to its bottom-up computation strategy as well as its active maintenance team and frequent releases. Second, we employ a more efficient condition (model-summarising

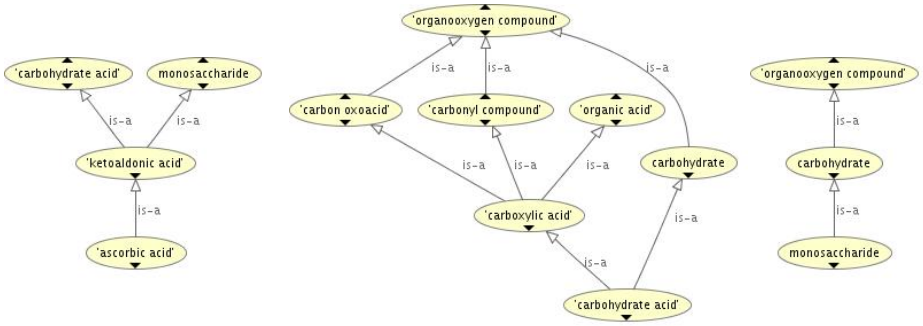

Fig. 4. Superclasses of ascorbic acid for the ChEBI OWL ontology release 95

acyclicity [6] instead of semantic acyclicity [21]) in order to obtain termination guarantees which allows for a more prompt decidability check.

Furthermore, while conducting the experiments we discovered a number of missing and inconsistent subsumptions from the manually curated ChEBI ontology; due to space restrictions we only mention a few of them. As one can infer from the chemical graph of ascorbic acid appearing in the top right of Figure 1, ascorbic acid is a carboxylic ester as well as a polyatomic cyclic entity. Inspite of the fact that these superclasses were exposed by our classification methodology, we were not able to identify them in the ChEBI hierarchy. Figure 4 shows the ancestry of ascorbic acid (CHEBI:29073) in the OWL version of the ChEBI ontology (due to space limitations we are not able to demonstrate subsumers above organooxygen compound); none of the concepts cyclic entity (CHEBI:33595), polyatomic entity (CHEBI:36357) or carboxylic ester (CHEBI:33308) is encountered among the superclasses of ascorbic acid (neither among the superclasses of organooxygen compound). Moreover, ascorbic acid is asserted as a carboxylic acid which is not the case as it can be deduced by the lack of a carboxy group in the chemical graph of ascorbic acid. We interpret the revealing of these modelling errors as an indication of the practical relevance of our contribution.

6 Conclusion and Future Research

We presented an implementation that performs logic-based classification of chemicals and builds upon a sound and complete reasoning procedure for an extension of logic programming; our prototype relies on the DLV system and is considerably quicker than previous approaches. For our evaluation, we represented a wide variety of chemical classes that are not expressible with OWL-based formalisms; additionally, our software revealed subclass relations that are missing from the manually curated ChEBI ontology as well as some erroneous ones. We demonstrated thus the capabilities of a logic programming ontology language which for the purposes of structure-based classification displays a favourable trade-off between expressive power and performance.

For the future, we plan to design a SMILES-based [31] surface syntax such that cheminformaticians are able to define chemical classes more intuitively and without the

need to script logic programming rules. We will also seek to extend our framework to accommodate subsumption between chemical classes so as to generate the complete chemical hierarchy as well as representation of numerical values [20] that would allow for more expressive modelling, such as classes depending on molecular weight. Moreover, we would be interested in exploring the integration of our prototype with ontology editors [2], Life Sciences platforms [26] and chemical structure visualisation tools [1, 18] as well as defining a mapping of the introduced formalism to RDF.

7 Acknowledgements

I would like to thank my supervisors Dr Markus Krötzsch and Prof. Ian Horrocks for insighful comments, the anonymous reviewers for useful references and Dr Chris Batchelor-McAuley for answering my chemistry questions. This work was supported by the EU FP7 SEALS and the EPSRC projects ConDOR, ExODA and LogMap.

References

1. Jmol: an open-source Java viewer for chemical structures in 3D., `www.jmol.org`
2. Protégé ontology editor, `http://protege.stanford.edu`
3. Chepelev, L., Dumontier, M.: Chemical Entity Semantic Specification: Knowledge representation for efficient semantic cheminformatics and facile data integration. Journal of Cheminformatics 3(20) (2011)
4. Chepelev, L., Dumontier, M.: Semantic Web integration of Cheminformatics resources with the SADI framework. Journal of Cheminformatics 3(16) (2011)
5. Chepelev, L.L., Riazanov, A., Kouznetsov, A., Low, H.S., Dumontier, M., Baker, C.J.O.: Prototype Semantic Infrastructure for Automated Small Molecule Classification and Annotation in Lipidomics. BMC Bioinformatics 12, 303 (2011)
6. Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., Wang, Z.: Acyclicity Conditions and their Application to Query Answering in Description Logics. In: KR 2012. AAAI Press (2012)
7. Dalby, A., Nourse, J.G., Hounshell, W.D., Gushurst, A.K.I., Grier, D.L., Leland, B.A., Laufer, J.: Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. Journal of Chemical Information and Computer Sciences 32(3), 244–255 (1992)
8. Deshpande, M., Kuramochi, M., Wale, N., Karypis, G.: Frequent Substructure-Based Approaches for Classifying Chemical Compounds. IEEE TKDE 17(8), 1036–1050 (2005)
9. Dumontier, M.: Molecular Symmetry and Specialization of Atomic Connectivity by Class-based Reasoning of Chemical Structure. In: OWLED (2012)
10. Ferreira, J.D., Couto, F.M.: Semantic Similarity for Automatic Classification of Chemical Compounds. PLoS Computational Biology 6(9) (2010)
11. Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming. In: ICLP/SLP. pp. 1070–1080 (1988)
12. Hastings, J., Dumontier, M., Hull, D., Horridge, M., Steinbeck, C., Stevens, R., Sattler, U., Hörne, T., Britz, K.: Representing Chemicals Using OWL, Description Graphs and Rules. In: OWLED. vol. 614 (2010)
13. Hastings, J., Magka, D., Batchelor, C., Duan, L., Stevens, R., Ennis, M., Steinbeck, C.: Structure-based classification and ontology in chemistry. Journal of Cheminformatics 4(8) (2012)

14. Hoehndorf, R., Dumontier, M., Gkoutos, G.V.: Identifying aberrant pathways through integrated analysis of knowledge in pharmacogenomics. Bioinformatics 28(16), 2169–2175 (2012)
15. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: the making of a Web Ontology Language. J. Web Sem. 1(1), 7–26 (2003)
16. King, R., Muggleton, S., Srinivasan, A., Sternberg, M.: Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. Proceedings of the National Academy of Sciences 93, 438–442 (1996)
17. Konyk, M., Battista, A.D.L., Dumontier, M.: Chemical Knowledge for the Semantic Web. In: DILS. pp. 169–176. Springer (2008)
18. Krause, S., Willighagen, E.L., Steinbeck, C.: JChemPaint - using the collaborative forces of the internet to develop a free editor for 2D chemical structures. Molecules pp. 93–98 (2000)
19. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM TOCL 7(3) (2006)
20. Magka, D., Kazakov, Y., Horrocks, I.: Tractable extensions of the description logic $\mathcal{EL}$ with numerical datatypes. J. Autom. Reasoning 47(4), 427–450 (2011)
21. Magka, D., Motik, B., Horrocks, I.: Modelling Structured Domains Using Description Graphs and Logic Programming. In: ESWC. pp. 330–344. Springer (2012)
22. de Matos, P., Alcántara, R., Dekker, A., Ennis, M., Hastings, J., Haug, K., Spiteri, I., Turner, S., Steinbeck, C.: Chemical Entities of Biological Interest: an update. Nucleic Acids Research 38(Database-Issue), 249–254 (2010)
23. Motik, B., Cuenca Grau, B., Horrocks, I., Sattler, U.: Representing ontologies using description logics, description graphs, and rules. Art. Int. 173(14) (2009)
24. Mungall, C.: Experiences Using Logic Programming in Bioinformatics. In: ICLP. pp. 1–21 (2009), Keynote talk.
25. Pence, H.E., Williams, A.: ChemSpider: An Online Chemical Information Resource. J. Chem. Educ. 87(11), 1123–1124 (2010)
26. Spjuth, O., Alvarsson, J., Berg, A., Eklund, M., Kuhn, S., Mäsak, C., Torrance, G.M., Wagener, J., Willighagen, E.L., Steinbeck, C., Wikberg, J.E.S.: Bioclipse 2: A scriptable integration platform for the life sciences. BMC Bioinf. 10, 397 (2009)
27. Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., Willighagen, E.L.: Recent developments of the chemistry development kit (CDK) - an open-source java library for chemo- and bioinformatics. Curr. Pharm. Des. 12(17), 2111–20 (2006)
28. Vardi, M.Y.: Why is Modal Logic So Robustly Decidable? In: Descriptive Complexity and Finite Models DIMACS Workshop. pp. 149–184 (1996)
29. Villanueva-Rosales, N., Dumontier, M.: Describing Chemical Functional Groups in OWL-DL for the Classification of Chemical Compounds. In: OWLED (2007)
30. Wang, Y., Xiao, J., Suzek, T.O., Zhang, J., Wang, J., Zhou, Z., Han, L., Karapetyan, K., Dracheva, S., Shoemaker, B.A., Bolton, E., Gindulyte, A., Bryant, S.H.: PubChem's BioAssay Database. Nucleic Acids Research 40(Database-Issue), 400–412 (2012)
31. Weininger, D.: 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 (1988)
32. Wolstencroft, K., Brass, A., Horrocks, I., Lord, P.W., Sattler, U., Turi, D., Stevens, R.: A Little Semantic Web Goes a Long Way in Biology. In: ISWC (2005)
33. Wolstencroft, K., Lord, P.W., Tabernero, L., Brass, A., Stevens, R.: Protein classification using ontology classification. In: ISMB (Supplement of Bioinformatics). pp. 530–538 (2006)