Modelling Highly Symmetrical Molecules: Linking Ontologies and Graphs

Oliver Kutz[1], Janna Hastings[2], and Till Mossakowski[1,3]

[1]Research Center on Spatial Cognition, University of Bremen, Germany;
[2]European Bioinformatics Institute, Cambridge, UK; [3]DFKI GmbH Bremen;
okutz@informatik.uni-bremen.de; hastings@ebi.ac.uk; Till.Mossakowski@dfki.de

Abstract. Methods for automated classification of chemical data depend on identifying interesting parts and properties. However, classes of chemical entities which are highly symmetrical and contain large numbers of homogeneous parts (such as carbon atoms) are not straightforwardly classified in this fashion. One such class of molecules is the fullerene family, which shows potential for many novel applications including in biomedicine. The Web Ontology Language OWL cannot be used to represent the structure of fullerenes, as their structure is not tree-shaped. While individual members of the fullerene class can be modelled in standard FOL, expressing the properties of the class as a whole (independent of the count of atoms of the members) requires second-order quantification. Given the size of chemical ontologies such as ChEBI, using second-order expressivity in the general case is prohibitively expensive to practical applications. To address these conflicting requirements, we introduce a novel framework in which we heterogeneously integrate standard ontological modelling with monadic second-order reasoning over chemical graphs, enabling various kinds of information flow between the distinct representational layers.

1 Introduction and motivation

Organic chemistry has seen a dramatic increase in available data in recent years, tracking progress in the search for novel therapeutics.[1] However, large-scale data that are not appropriately organised can be more of a burden than a benefit. Ontologies and knowledge-based methods for automated classification are increasingly harnessed to address this challenge. ChEBI – Chemical Entities of Biological Interest – is a chemical ontology that is widely used to organise and classify chemical data [4]. However, ChEBI is manually maintained, reducing its scalability. Methods for automated classification of chemical entities depend on algorithms which reduce complex molecular graphs to lists of interesting parts and properties, such as atomic constituents and groups, charges and overall molecular weight. Knowledge representation and reasoning for chemistry has also largely been dominated by this paradigm [13, 12, 14].

[1] The basic ideas formulated in this paper were previously presented at the Deep Knowledge Representation Workshop DKR-11, Banff, Canada, 2011 (2nd prize in the DKR competition).

In recent years there has been a progression in capacity for the synthesis of highly symmetrical, polycyclic chemical entities, which are made up of a very small number of part *sorts* (e.g. mainly carbon atoms) with a very large number of actual parts. Polycyclic carbon molecules show incredible topological versatility, not only forming spheres, tubes and sheets, but also molecular Möbius strips [8, 2, 11] and knots [7, 17], as illustrated in Figure 1. These molecules elicit increased interest following advances in synthesis methods towards a nanoscale molecular 'machinery' with carefully designed shapes that are able to rival the power and scale of biological machinery [19].

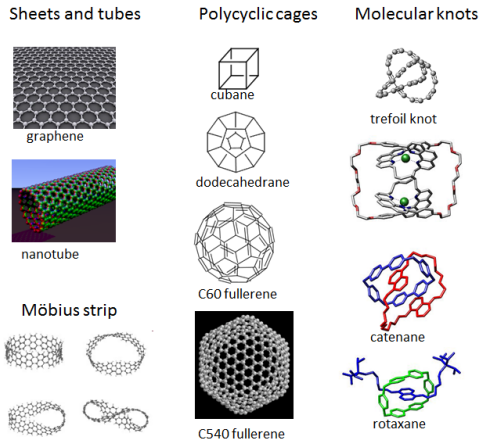


Fig. 1. Some examples of highly symmetric molecules, constituted almost entirely by carbon atoms. The overall arrangements of atoms in the molecules, rather than the nature of functional groups, characterise their types.

Since their parts are homogeneous, listing part types cannot distinguish distinct such classes. Rather, such molecules must be characterised by their *shape* or *topology*. For example, fullerene molecules form spherical or ellipsoidal cages, Möbius molecules display classical Möbius topologies, while molecular knots and interlinked chains display the topological and shape properties of macromolecular knots and chains. To adequately represent knowledge about these molecules requires the ability to describe and reason over features which apply to the entire molecular graph (i.e. the connection of atoms via bonds).

In what follows, we will argue that formalisms with limited expressivity such as OWL are not sufficient to represent deep knowledge about this class of molecules. We will therefore complement OWL with the more expressive formalism of monadic second-order logic (MSOL).

2 Background

OWL representation. Chemical entities can be represented and exchanged in the form of chemical graphs, in which atoms form the vertices and covalent chemical bonds the edges. However, complex graphs that contain cycles cannot be faithfully modelled at the class level in OWL due to the requirement that all axioms in OWL have models shaped like a tree [12].

Can we recognise members of the fullerene class of molecules based on the structure of their chemical graphs? A formulation of C60 fullerene and a graphene of 60 atoms might refer explicitly to their differing shapes in order to allow automated reasoning to distinguish them, using axioms such as *hasPart* only CarbonAtom and *hasShape* some Sphere or *hasShape* some Flat.

However, this approach clearly does not allow automated reasoning to deduce the class of the molecule based on the properties of the molecular graph, since a human has to specify the shape of the molecule. Following this pattern, a different shape has to be defined for every differently shaped molecule, with no means of automatically discerning relationships or similarity between the stated shapes. Furthermore, those properties of the molecules that depend on their shapes are not explained by the information contained in the ontology. Many of the properties of fullerenes stem from the fact that they can enclose other molecules inside their cage structure, a property not shared by graphene. The properties of molecular knots stem from the fact that they are mechanically interlocked. What is required is a framework that is able to *define* classes of molecules based on properties of the graphs of their members, and then *deduce* which molecular graphs belong to these classes.

Description graphs, rules and FOL. Cycles can be represented adequately in rules, which are combined with OWL for ontology engineering in the DL-safe rule extension [16]. The DL-safe rules extension, however, is applicable only to explicitly named objects in the ontology (individuals), to ensure decidability of the resulting knowledge base. This means that it is not possible to reason at the class level about highly symmetric molecules using this formalism.

This shortcoming motivated the introduction of description graphs [15], an OWL extension for expressing the structure of complex objects at the class level. However, the knowledge base is still constrained in that the OWL axioms and the edge properties in the description graphs must be kept separate. The reasoning capability of the framework is limited to what can be expressed in "graph-safe" rules: rules which do not mix graph edge properties and OWL object properties. Furthermore, there are inherent limitations in the use of rules for reasoning, since there is no $\forall$ quantification in the rules formalism, which means that properties of all atoms in a given graph cannot be used for reasoning [12]. In an effort to relax these limitations, a radically different semantics has recently been proposed, based on logic programming: description graph logic programs [14]. Since the semantics from logic programming ensures decidability in a different way to the OWL model-theoretic semantics, there is no need for property separation, thus the ontology designer may interchangeably use OWL and description graph properties in creating the knowledge base. This formalism allows representation and reasoning with cyclic chemical structures at the class level. It is

possible, for example, to define a particular member of the fullerenes class, such as dodecahedrane, and to use reasoning for detection of cycles of fixed lengths. However, it is not possible to express the properties of fullerenes as a whole. Using full FOL it is possible to get very close to a definition for the fullerenes, including axioms that every atom must have 3 or 4 bonds, every atom must belong to a cycle, and every cycle (face) must have 5 or 6 members. However, such constraints ("local perspective") cannot allow correct classification in all cases. For example, fullerenes of different sizes (for example, C540, C240 and C60) can be *nested* inside one another. The local perspective at each atom and at each face correctly matches the best definition that is possible to specify in FOL. Yet, this should be classified as a *complex* consisting of multiple fullerenes, rather than as itself a (single) fullerene molecule. To distinguish the complex from a single molecule, the second-order construct of *graph connectedness* is needed. However, it is well-known that connectedness is not first-order definable [5].

3 Properties of graphs for chemical classes

In order to distinguish between fullerenes, graphenes, strips and Möbius strips, we need to define some properties of graphs based on chemical graph theory [18]. For simplification we will assume that all graphs are *finite*, which is true of all graphs corresponding to real chemical entities.

Planar polyhedral graphs. A chemical graph is **planar** if it can be drawn on a flat plane without any edges crossing. Overwhelmingly, most chemical entities can be described by planar graphs. The only exception found in a recent analysis of public compound databases were Möbius-like molecules [8]. A graph is **cubic** if all vertices have degree three, i.e. are connected to three other vertices. It is **connected**, if any two vertices are connected by a path, and it is **3-connected** if it is connected and remains so after removal of any two vertices. A graph is **polyhedral** iff it is the graph of some convex polyhedron. By Steinitz' theorem (1922), this is equivalent to being 3-connected and planar (see [20]). Indeed, polyhedral graphs, while being planar (2D), are typically represented as convex polyhedra (3D). A **polycyclic cage** is any polyhedral graph. Chemical examples include cubane, tetrahedrane, and of course all fullerenes. A **fullerene** is a cubic polyhedral graph consisting of hexagons and pentagons only. By the Euler formula for polyhedra, one can show that the number of pentagons must always be 12. A **closed nanotube** is a fullerene which is extended into a tube shape with a circular extension consisting only of hexagons between the two ends, the latter consisting of two hemispheres of the buckyball structure. An **open nanotube** is a cubic polyhedral graph consisting of hexagons and two non-hexagons (the two non-hexagons are the outer boundaries).

Planar non-polyhedral graphs. A **graphene** is a planar graph consisting of hexagons and one face (the outer boundary) not necessarily being a hexagon, where all vertices involved in the outer boundary have degree two or three, while the remaining vertices have degree three.

Non-planar graphs. A **Möbius strip** is non-planar graph, consisting of hexagons and one non-hexagon (the outer boundary).

4 Describing molecule graph classes in MSOL

We want to formalise the definitions of graph classes such that membership in a graph class can be machine-checked. It has been noted (see [3]) that the role finite automata play for the specification of word languages is played by *monadic second-order logic* (MSOL) for expressing graph properties and defining graph classes. Although the general problem is NP-complete, monadic second-order logic for graphs can be model-checked quite efficiently; indeed, for graphs with bounded tree-width, model checking can be done in linear time. MSOL for graphs consists of untyped first-order logic, extended with quantification over sets (and membership in such sets). We assume binary predicates $edge$, $edge2$ and $edge3$ for all bonds, double bonds and triple bonds, respectively. We also assume unary predicates like $Carbon$ for the atoms (and suitable atom classes) in the periodic table. When writing MSOL formulas, we use syntactic sugar like unique-existential quantifiers and number quantifiers, which can easily be coded out even in first-order logic. We also will freely use standard set-theoretic notation where it can easily be coded out into MSOL. V denotes the set of all vertices. While the expressive power of MSOL suffices to axiomatise

$$Cubic \Leftrightarrow \forall x.\exists!3y.edge(x,y)$$

$$degree_n(x) \Leftrightarrow \exists!ny.edge(x,y)$$

$$Planar \Leftrightarrow \neg(\exists$$  $$) \wedge \neg(\exists$$  $$)$$

$$Connected\_Subgraph(C) \Leftrightarrow \forall D \subseteq C, E \subseteq C.C = D \cup E \Rightarrow \exists u \in D, v \in E.edge(u,v)$$

$$Connected \Leftrightarrow \exists C.\forall x.x \in C \wedge Connected\_Subgraph(C)$$

$$Cycle(C) \Leftrightarrow Connected\_Subgraph(C) \wedge \forall x \in C\ \exists y \in C.edge(x,y)$$

$$Three\_Connected \Leftrightarrow \forall x,y.Connected\_Subgraph(V \setminus \{x,y\})$$

$$Polyhedron \Leftrightarrow Planar \wedge Three\_Connected$$

$$Polycyclic\_Cage \Leftrightarrow Polyhedron$$

$$Face(C) \Leftrightarrow Cycle(C) \wedge Connected\_Subgraph(V \setminus C) \wedge \forall u,x,y,z \in C.edge(u,x) \wedge edge(u,y) \wedge edge(u,z) \rightarrow (x = y \vee x = z \vee y = z)$$

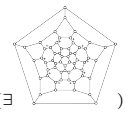
$$Pent(C) \Leftrightarrow Cycle(C) \wedge \exists!5x.x \in C$$

$$Hex(C) \Leftrightarrow Cycle(C) \wedge \exists!6x.x \in C$$

$$Carbon\_Allotrope \Leftrightarrow \forall x.Carbon(x)$$

$$Fullerene \Leftrightarrow Carbon\_Allotrope \wedge Polycyclic\_Cage \wedge Cubic \wedge \forall C.Face(C) \rightarrow Pent(C) \vee Hex(C)$$

$$Closed\_Nanotube \Leftrightarrow Fullerene \wedge$$

$$\wedge\ (\exists$$  $$)$$

$$Open\_Nanotube \Leftrightarrow Carbon\_Allotrope \wedge Polyhedron \wedge Cubic \wedge \exists B,C.Face(B) \wedge Face(C) \wedge B \neq C \wedge \forall D.Face(D) \rightarrow B = D \vee C = D \vee Hex(D)$$

$$Graphene \Leftrightarrow Carbon\_Allotrope \wedge Planar \wedge \exists B.Face(B) \wedge (\forall x \in B.degree_2(x) \vee degree_3(x)) \wedge \forall C.Face(C) \rightarrow B = C \vee (Hex(C) \wedge \forall x \in C.degree_3(x))$$

$$Moebius\_Strip \Leftrightarrow \neg Planar \wedge \exists B.Face(B) \wedge \forall C.Face(C) \rightarrow B = C \vee Hex(C)$$

Fig. 2. MSOL formalisation of molecule classes.

most graph classes that we are interested in, even with the above syntactic sugar, often the axioms can become cumbersome and large. We therefore additionally use the *nested conditions* of [9, 10]. The simplest and most prominent formulas here are of form $(\exists G)$, where G is a graph with some edges annotated with $+$. The semantics is that G can be injectively embedded into the given graph, where each edge labelled with $+$ may be mapped to a finite path (this may be used to express that a certain G is a minor of the given graph). The MSOL formalisation of the above notions is shown in Fig. 2. Note that graph classes are represented as MSOL model classes; this means that e.g. a graph is cubic if and only if it (when seen as a MSOL model) satisfies the formula $\forall x.\exists!3y.edge(x,y)$. The correctness of the definition of polyhedral graph follows from Steinitz' theorem discussed above, and the correctness of the definition of planarity follows from Kuratowski's characterisation in terms of forbidden minors.

5 Connecting Ontology and Graph Layers

We have seen that monadic second-order logic combined with nested conditions provides a convenient formalism for adequate formalisation of graph-conditions relevant for the modelling of chemicals. However, how can such specifications of graph classes be related to existing ontologies of molecules such as ChEBI [4] that are formulated in a light-weight ontology language like OWL-EL? Clearly, one cannot expect to be able to formalise deeper graph-theoretical properties in OWL. However, using the MSOL formalisation, we can build what we call a **grounded ontology**: class names such as fullerene are equipped with a (or several) formal MSOL specification(s), and specific instances, i.e. object names are equipped with concrete graphs. Such an association, if done systematically, will give rise to a number of automated reasoning problems such as model and subclass checking, deduction in MSOL, and abduction, here put into a new context. To motivate the following definition, note the following considerations. Two (different) ontology classes may be equipped with the same MSOL theories. This reflects an *intensionality* in the definition of the ontological classes which, although having different ontological definitions, denote the same structural class of molecules. Conversely, one and the same ontology class may be equipped with different MSOL theories. This corresponds to an intensionality in the realm of graph classes, where different descriptions of a graph class may be found in the literature (e.g. if the molecule has different structural variants). Therefore, we express soundness of the relation between ontology classes and monadic second-order theories as functionality modulo logical equivalence.

We consider the ontology as the primary, and the graph-based formalisation as a secondary source of information. This is reflected in the following formal definition for ontologies expressed in $\mathcal{ALC}$:[2]

Definition 1. *Fix an $\mathcal{ALC}$ ontology $O = \langle T, A\rangle$, where T is a TBox, and A is an ABox. Let $\mathbf{C}$ be the set of $\mathcal{ALC}$ (sub)concept descriptions (atomic or complex) and $\mathbf{I}$ the set of object names appearing in O, $\mathfrak{T}$ a set of finite* MSOL *theories*[3]*, and $\mathfrak{G}$ a set of* MSOL *finite undirected graphs. An **ontology-graph association** (**oga** for short) is a pair of relations $\rightsquigarrow = \langle \rightsquigarrow_{\mathsf{T}}, \rightsquigarrow_{\mathsf{A}} \rangle$, where*

$$\rightsquigarrow_{\mathsf{T}} \subseteq \mathbf{C} \times \mathfrak{T} \quad \textit{and} \quad \rightsquigarrow_{\mathsf{A}} \subseteq \mathbf{I} \times \mathfrak{G}$$

*$\rightsquigarrow$ is **total** if for any concept $C \in \mathbf{C}$ and object $a \in \mathbf{I}$ there exist $T \in \mathfrak{T}$, $G \in \mathfrak{G}$ such that $C \rightsquigarrow_{\mathsf{T}} T$ and $a \rightsquigarrow_{\mathsf{A}} G$.*
*$\rightsquigarrow$ is **sound** if for all $O \models C \sqsubseteq D$ and $a : C$ and $b : D$ we have: $C \rightsquigarrow_{\mathsf{T}} S$, $D \rightsquigarrow_{\mathsf{T}} T$ implies $S \vDash_{\mathsf{MSOL}} T$ and $a \rightsquigarrow_{\mathsf{A}} G$ implies $G \models_{\mathsf{MSOL}} S$ and $b \rightsquigarrow_{\mathsf{A}} H$ implies $H \models_{\mathsf{MSOL}} T$.*
*$\rightsquigarrow$ is **complete** if for all $C \rightsquigarrow_{\mathsf{T}} S$, $D \rightsquigarrow_{\mathsf{T}} T$ we have: $S \vDash_{\mathsf{MSOL}} T$ implies $O \models C \sqsubseteq D$ and for all $a \rightsquigarrow_{\mathsf{A}} G$ and $b \rightsquigarrow_{\mathsf{A}} H$ with $G \models_{\mathsf{MSOL}} S$ and $H \models_{\mathsf{MSOL}} T$ we have $a : C$ and $b : D$.*
*$\rightsquigarrow$ is **graph-extensional** if $C \rightsquigarrow_{\mathsf{T}} S$, $C \rightsquigarrow_{\mathsf{T}} T \implies \vDash_{\mathsf{MSOL}} S \leftrightarrow T$.*
*$\rightsquigarrow$ is **class-extensional** if $C \rightsquigarrow_{\mathsf{T}} S$, $D \rightsquigarrow_{\mathsf{T}} S \implies O \models C \equiv D$.*

Proposition 1. *Completeness implies class-extensionality (not vice versa), and soundness implies graph-extensionality (not vice versa).*

The logical structure of ontology-graph associations is illustrated in Figure 3. Note that the new aspect here is that there is a *shift of levels*, namely graphs are models in MSOL, while they are individuals in OWL. The correspondences (including reasoning) are as follows: Rather than *integrating* or *combining* different logics and running the risk of losing any of the desirable properties of the special purpose formalisms, our approach realises an interlinked formalisation of different aspects of the domain of chemical molecules that relies on a mapping between the different layers. Whilst (lightweight) ontology languages are used to cope with the rather large chemical ontologies, MSOL is used to adequately capture some of the ontologically relevant spatial structure of molecule classes. Obviously, to make this approach worthwhile, we need to establish systematic ways of exchanging information between these two layers of different abstraction and expressiveness.

[2] We focus here on OWL ontologies with at most $\mathcal{ALC}$ expressivity. However, all definitions carry over to FOL ontologies mutatis mutandis.

[3] We can therefore meaningfully use Boolean combinations of such theories.

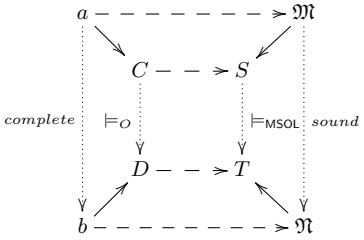


Fig. 3. Ontology-graph association: OWL and MSOL.

MSOL term	chemical notion	OWL term
MSOL theory	molecule class	OWL class
graph	molecule	OWL individual
model checking	instance checking	OWL ABox checking
logical entailment	subclass relation	OWL subclass (TBox)
consistent theory	nonempty class	satisfiable OWL class

Fig. 4. Grounded ontology: correspondences between MSOL, OWL, and chemical notions.

Deduction. Proven entailment between MSOL theories may be used to assert subsumption between the corresponding classes in the chemical ontology (e.g. the ontology in Fig. 5 has been obtained in this way). This corresponds to ensuring completeness as defined above.

Abduction. Abduction [6] can be used to hypothesise new correspondences. For example, given $A \sqcap B \rightsquigarrow_{\mathsf{T}} T_1$, $A \rightsquigarrow_{\mathsf{T}} T_2$ and $\models_{MSOL} T_1 \leftrightarrow T_1 \wedge T_2$, this may have the explanation $B \rightsquigarrow_{\mathsf{T}} T_3$.

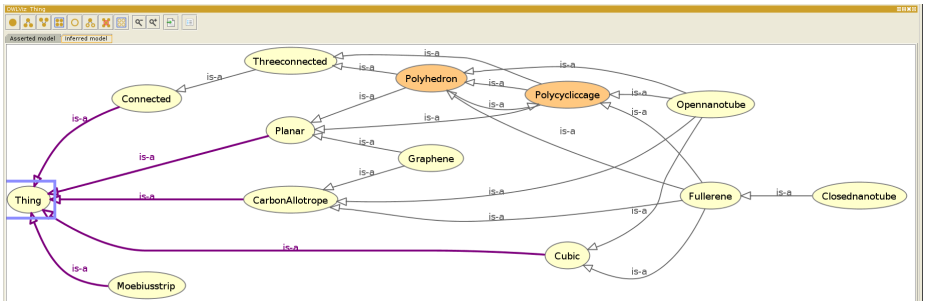


Fig. 5. Class hierarchy computed from MSOL implications

Other related reasoning problems are **induction**, i.e. given a set of example molecules (e.g. MOL datafiles), learn a corresponding graph class specified in MSOL, and **model checking**, i.e. given a Möbius strip, check (using a tool such as [1]) that it is non-planar. Moreover, to show that it is additionally a loop is an interesting and non-trivial subsumption check. Although MSOL entailment is in general undecidable, logical entailment in second-order logic can be approximated with automated theorem provers like LEO-II (`http://www.ags.uni-sb.de/~leo/`). An initial OWL ontology computed from the logical implications among the MSOL axiomatisations given in Sec. 4 is shown in Fig. 5 (available at `ontohub.org`). Here, the implications between the graph classes are mostly

definitorial and therefore easy to check automatically, and trigger the creation of corresponding subsumptions in the ontology.

6 Conclusions

Representation and reasoning with structured objects such as molecules is still an area of active research and development for ontologists and chemoinformaticians. Chemical ontologies such as ChEBI [4] provide one solution to this problem through careful manual classification. Formal ontology aims to supplement such manual efforts with explicit computable knowledge representation and accompanying automated reasoning. We here focus on a particularly interesting and challenging class of molecules for such formalisation, and examine an approach which uses the expressive power of monadic second-order logic (MSOL) to formalise properties that cannot be defined in OWL, proposing to systematically link the two layers.

Compared to algorithmic approaches of molecule classification, we can offer a language for a *declarative* description of molecules and molecule classes, which offers a path to not only instance checking (as in the algorithmic case), but also to subclass checking, through MSOL theorem proving. We propose to combine this with OWL ontologies such as ChEBI, thus obtaining a "grounded ontology", where OWL subclass relations can be verified or inferred by looking at the corresponding graph properties in MSOL.

In the semi-automatic generation of MSOL theories chemical graphs datasets via inductive reasoning, a problem that has to be considered is that abstracting a graph class from a finite number of sample molecules can sometimes produce ambiguous results. Importantly, classes of molecules conforming to particular graph theories may have characteristic emergent properties in terms of chemical and biochemical reactivity and activity profile that none of the superclasses (with less restrictive accompanying graph theories) display. The activity and reactivity properties of molecules would need to be included in a separate ontological layer within the framework we describe. Note that the MSOL approach can only classify molecules based on properties of their graphs. However, from a graph-theoretic point of view, the molecular trefoil is equivalent to a simple loop. In order to distinguish it from the loop, one has to consider its embedding into Euclidean space, and use knot theory. Future work should consider invariants from knot theory (such as genus, polynomials and groups) in a similar role as that in which we presently propose to use MSOL. Also, the results of computational graph theory will be useful, e.g. for optimising parts of the model checking for graphs.

References

1. S. Arnborg, 'A general purpose MSOL model checker and optimizer based on Boolean function representation', Technical report, KTH, Stockholm, Sweden, (1994).
2. E. W. S. Caetano, V. N. Freire, S. G. dos Santos, D. S. Galvao, and F. Sato, 'Möbius and twisted graphene nanoribbons: stability, geometry and electronic properties', *The Journal of Chemical Physics*, **v128**(164719), (2008).
3. B. Courcelle and J. Engelfriet, *Graph structure and monadic second-order logic—A language theoretic approach*, Cambridge University Press, 2011.
4. P de Matos, R Alcántara, A Dekker, M Ennis, J Hastings, K Haug, I Spiteri, S Turner, and C Steinbeck, 'Chemical Entities of Biological Interest: an update', *Nucl. Acids Res.*, **38**, D249–D254, (2010).
5. H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer, 2005.
6. C. Elsenbroich, O. Kutz, and U. Sattler, 'A Case for Abductive Reasoning over Ontologies', in *Proc. of OWLED-06*, (2006).
7. J. Canceill et. al, 'From classical chirality to topologically chiral catenands and knots', in *Supramolecular Chemistry I Directed Synthesis and Molecular Recognition*, volume 165 of *Topics in Current Chemistry*, 131–162, Springer Berlin / Heidelberg, (1993).
8. M. J. Wester et al., 'Scaffold Topologies. 2. Analysis of Chemical Databases', *Journal of Chemical Information and Modeling*, **48**(7), 1311–1324, (2008).
9. A. Habel and K.-H. Pennemann, 'Correctness of high-level transformation systems relative to nested conditions', *Mathematical Structures in Computer Science*, **19**(2), 245–296, (2009).
10. A. Habel and H. Radke, 'Expressiveness of graph conditions with variables', *ECEASST*, **30**, (2010).
11. D. Han, S. Pal, Y. Liu, and H. Yan, 'Folding and cutting dna into reconfigurable topological nanostructures', *Nature Nanotechnology*, **5**, 712–717, (2010).
12. Janna Hastings, Despoina Magka, Colin Batchelor, Lian Duan, Robert Stevens, Marcus Ennis, and Christoph Steinbeck, 'Structure-based classification and ontology in chemistry', *Journal of Cheminformatics*, **4**(1), 8, (2012).
13. M Konyk, A De Leon, and M Dumontier, 'Chemical knowledge for the semantic web', in *Proceedings of Data Integration in the Life Sciences (DILS2008)*, volume LNBI 5109, pp. 169–176, Evry, France, (2008). Lecture Notes in Computer Science.
14. D. Magka, B. Motik, and I. Horrocks, 'Modelling structured domains using description graphs and logic programming', Technical report, Dept. of Computer Science, U. of Oxford, (2011).
15. B. Motik, B. Cuenca Grau, I. Horrocks, and U. Sattler, 'Representing Ontologies Using Description Logics, Description Graphs, and Rules', *Artificial Intelligence*, **173**(14), 1275–1309, (2009).
16. B. Motik, U. Sattler, and R. Studer, 'Query Answering for OWL-DL with Rules', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, **3**(1), 41–60, (2005).
17. H. Rzepa. Molecular möbius strips and trefoil knots, 2003. `http://www.ch.ic.ac.uk/motm/trefoil/`, last accessed December 2011.
18. N. Trinajstic, *Chemical graph theory*, CRC Press, Florida, USA, 1992.
19. H.-R. Tseng, S. A. Vignon, and J. F. Stoddart, 'Toward chemically controlled nanoscale molecular machinery', *Angewandte Chemie International Edition*, **42**, 1491–1495, (2003).
20. E. W Weisstein. Polyhedral graph, 2011. `http://mathworld.wolfram.com/PolyhedralGraph.html`.