

How to model the shapes of molecules?

Combining topology and ontology using heterogeneous specifications

Janna Hastings^{1,2*}, Oliver Kutz³, and Till Mossakowski^{3,4}

¹ Chemoinformatics and Metabolism, European Bioinformatics Institute, Cambridge, UK

² Swiss Centre for Affective Sciences, University of Geneva, Switzerland

³ Research Center on Spatial Cognition, University of Bremen, Germany

⁴ DFKI GmbH Bremen and University of Bremen, Germany

Abstract. Classification of chemical entities is generally based on identifying the interesting parts and properties of the molecules. However, classes of chemical entities which are highly symmetrical and which contain large numbers of homogeneous parts (such as carbon atoms) are not straightforwardly classified in this fashion. One such class of molecules is the recently developed fullerene family, discovery of which led to the award of the Nobel prize for chemistry in 1996. Fullerene molecules show potential for many novel applications including in biomedicine. Whilst standard knowledge representation approaches in chemistry are inadequate to allow the automatic classification of chemical entities as members of the fullerene class based on their chemical structure, standard OWL representations are insufficient for this task as well.

We here sketch an alternative framework in which we heterogeneously integrate ontological modelling with monadic second-order reasoning over chemical graphs and topological features of the molecules, enabling a threefold information flow between these distinct representational layers, namely a debugging of the ontology, an abductive enrichment of the ontology, and an inductive learning of new graph classes.

1 Introduction and motivation

Overwhelmingly, bioactive chemical entities such as molecules and ions are characterised by their parts, that is, atomic constituents and groups of atoms which have characteristic reactivity. Knowledge representation and reasoning for chemistry has largely been dominated by this paradigm, with approaches to defining chemicals and reasoning over them making reference to constituent parts and their interconnections (e.g. [12, 11]). However, 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. only or mainly carbon atoms) with a very large number of actual parts. Buckminsterfullerene molecules are an example of such chemicals. Other examples are nanotubes, nanostrips, molecular knots and molecules which form molecular Möbius strips [16].

Such molecules are best characterised by their *shape* or *topology* rather than by a reduction to their parts. For example, fullerene molecules form spherical or ellipsoidal cages, and Möbius molecules display classical Möbius topologies. Traditional chemoinformatic algorithmic approaches do not provide mechanisms to derive the overall shape for the molecules based on the descriptions of the parts. What is required in order to adequately represent knowledge about these molecules is the ability to describe and reason over features which apply to the molecule as a whole composite unit. The OWL standard is commonly used for formalising ontologies. However, to adequately represent the topological features of these highly symmetrical molecules, and reason with these for purposes of classification, a formalism is needed which goes beyond OWL. Several hybrid formalisms have been proposed in the literature for combining ontological reasoning with spatial and topological reasoning, and we just mention logics for concepts and similarity [17], \mathcal{E} -connections [13], and complex concrete domains [15]. Whilst they are more expressive

* Corresponding author: Janna Hastings, Chemoinformatics and Metabolism, European Bioinformatics Institute, Cambridge, UK. Email: hastings@ebi.ac.uk

than OWL and can capture some of the relevant aspects of the structure of chemical molecules, they have not been designed with this application in mind and their applicability is therefore rather limited. Moreover, none of these formalisms has been implemented.

We here present an alternative approach for defining classes of molecules based on overall graph and topological properties within a framework that allows for heterogeneous and modular ontology design. In particular, we will make use of the theory of hyperontologies as introduced in [14], a framework for complex knowledge representation based on a heterogeneous set of (interlinked) logical languages and a set of structuring mechanisms between and within these languages. We will briefly sketch the main features of hyperontologies and then show how this framework can be used to characterise and classify highly symmetrical molecules using (monadic) second-order logic in combination with standard ontological representation of the molecules based on their parts as specified in OWL.

2 Highly symmetrical molecules

2.1 Background

Recent years have seen an explosion in novel methods for the synthesis of nanomaterials such as carbon fullerenes. Fullerenes were discovered in 1985, and their properties rendered them of immediate interest for multiple applications including in medicinal chemistry [2]. Their cage structure, hydrophobicity, electrochemical and physical properties coupled with the enormous scope for forming derivatives, render them attractive as carriers for gene and drug delivery systems and for many other novel applications [2]. An important property of such molecules is their high level symmetry. Figure 1 illustrates some examples of fullerenes as well as other highly symmetrical molecules.

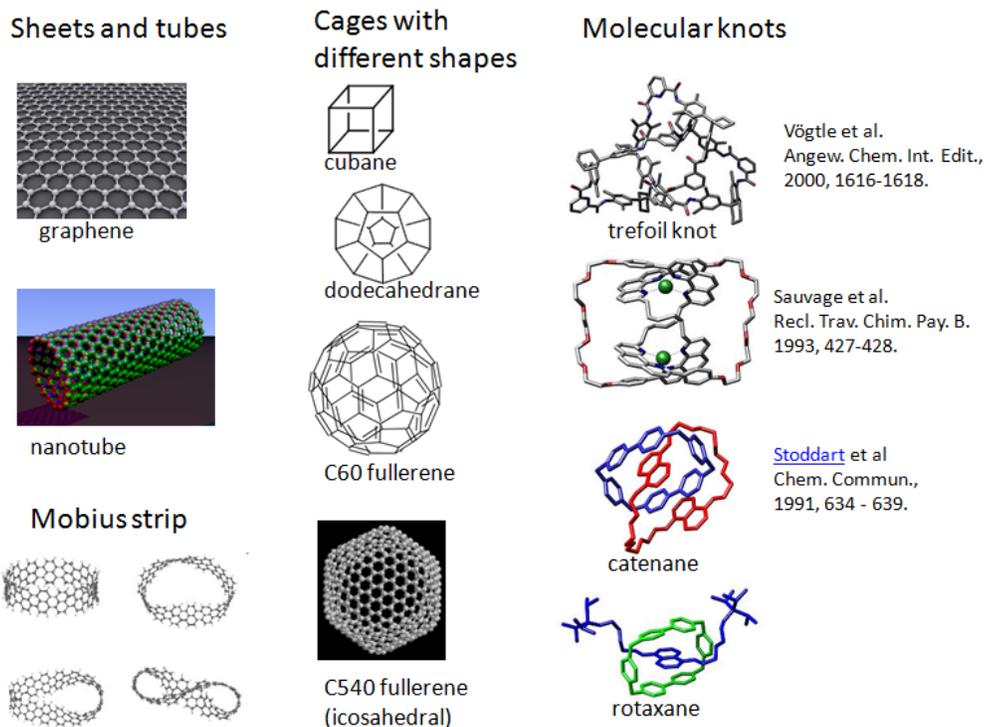


Fig. 1. Some examples of highly symmetric molecules

Polycyclic carbon molecules show incredible topological versatility, not only forming spheres, tubes and sheets, but also molecular Möbius strips [20, 4, 10] and knots [5, 16]. A classification of these molecules based on their shapes therefore needs to be able to distinguish between flat, cylindrical, spherical, ellipsoidal and more complex shape arrangements of the atoms within the molecules.

2.2 OWL representation

Chemical entities are commonly represented and exchanged in the form of chemical graphs, in which atoms form the vertices and covalent chemical bonds count as edges. However, such complex graphs cannot be modelled in OWL directly due to the tree model requirement [11]. Chemical graphs can be represented in description graphs, a recent extension to OWL for modelling structured objects, but the content of such graphs cannot be referred to in OWL axioms at the expense of decidability, and properties of the overall graph cannot be used for classification even when combined with rules, due to the absence of a \forall quantification in the rules formalism [11].

The questions which will guide our implementation in what follows are:

- Can we recognise members of the fullerene class of molecules based on chemical graphs?
- That is, can we distinguish fullerenes from, say, graphenes, and from molecular strips or tubes?
- Can we distinguish standard molecular strips or tubes from molecular Möbius strips?

As an initial attempt to formalise in OWL, using C60 fullerene as an example and comparing with a graphene of 60 atoms, we have:

```
C60Fullerene SubClassOf hasPart (exactly 60) CarbonAtom
                    and    hasPart only CarbonAtom
                    and    hasShape some Sphere
Graphene SubClassOf hasPart (min 60) CarbonAtom
                    and    hasPart only CarbonAtom
                    and    hasShape some Flat
```

We have explicitly characterised the shape of the molecule using named shapes. Following this approach, however, leaves very mysterious the nature of the shapes being described, and it may well lead to a situation where a different shape has to be defined in the ontology for every differently shaped molecule. And the properties of those molecules which 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 which is not shared by graphene. The properties of molecular knots stem from the fact that they are shaped as mechanically interlocked toruses.

While the characterisation of the shape in terms of predicates not further defined is interesting for chemists and humans, it does not allow for answering more sophisticated questions such as *Can the molecule serve as a carrier for another much smaller molecule?* 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.

3 Topological properties of graphs relevant for chemical classes

In order to distinguish fullerenes from graphenes and graphenes from 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 in what follows are *finite*, which is of course true of all graphs corresponding to real chemical entities. In the below, we describe the graph properties of chemical classes in the family of highly symmetric molecules.

3.1 Planar polyhedral graphs

A chemical graph is *planar* if it can be drawn on a flat plane without any edges crossing. Note that a planar graph may have non-planar representations depending on how it is laid out, but to be classified as planar it suffices that at least one planar representation exists.

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 [20]. However, since both fullerenes and graphenes are planar, we will need to define a few more properties in order to formalise our definition of these classes.

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 [19]. 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 end, 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).

3.2 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.

3.3 Non-planar graphs

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

4 Reasoning scenarios and formal representation

4.1 Describing graph classes

We want to formalise the definitions of graph classes of the previous section, such that membership in a graph class can be machine-checked. It has been noted that the role finite automata play for the specification of word languages is played by *monadic second-order logic* (MSOL) [6] 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 easily can be coded out even in first-order logic. We also will freely

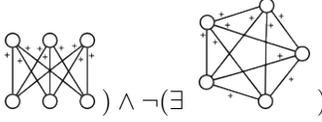
use standard set-theoretic notation where it can easily be coded out into MSOL. V denotes the set of all vertices.

While the expressive power of MSOL suffices to axiomatise 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 [8, 9]. 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 labeled 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).

For example, the above notions can be formalised as follows:

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

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

$$Planar \Leftrightarrow \neg(\exists \text{ (Diagram 1) }) \wedge \neg(\exists \text{ (Diagram 2) })$$


$$Path(C, x, y) \Leftrightarrow [x \in C \wedge y \in C \wedge (\forall u \in C. u \neq y \rightarrow edge(u, v)) \\ \wedge \neg \exists D \subset C. \forall u \in D \exists v \in D. edge(u, v)]$$

$$Connected \Leftrightarrow \forall x, y \exists C. Path(C, x, y)$$

$$Connected_subgraph(C) \Leftrightarrow \forall x, y \in C \exists D \subseteq C. Path(D, x, y)$$

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

$$Three_connected \Leftrightarrow \forall x, y. Connected_subgraph(V \setminus \{x, y\})$$

$$Polyhedron \Leftrightarrow Planar \wedge Three_connected$$

$$Polycyclic_cage \Leftrightarrow Polyhedron$$

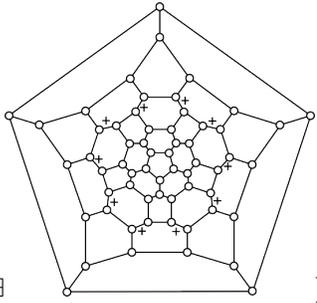
$$Face(C) \Leftrightarrow Cycle(C) \wedge Connected_subgraph(V \setminus C) \\ \wedge \forall u, x, y, z \in C. edge(u, x) \wedge edge(u, y) \wedge edge(u, z) \rightarrow (x = y \vee x = z \vee y = z)$$

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

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

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

$$Fullerene \Leftrightarrow Carbon_Allotrope \wedge Polycyclic_cage \wedge Cubic \\ \wedge \forall C. Face(C) \rightarrow Pent(C) \vee Hex(C)$$

$$Closed_nanotube \Leftrightarrow Fullerene \wedge (\exists \text{ (Diagram 3) })$$


$$Open_nanotube \Leftrightarrow Carbon_Allotrope \wedge Polyhedron \wedge Cubic \\ \wedge \exists B, C. Face(B) \wedge Face(C) \wedge B \neq C \\ \wedge \forall D. Face(D) \rightarrow B = D \vee C = D \vee Hex(D)$$

$$\begin{aligned}
\text{Graphene} &\Leftrightarrow \text{Carbon_Allotrope} \wedge \text{Planar} \\
&\wedge \exists B. \text{Face}(B) \wedge (\forall x \in B. \text{degree}_2(x) \vee \text{degree}_3(x)) \\
&\wedge \forall C. \text{Face}(C) \rightarrow B = C \vee (\text{Hex}(C) \wedge \forall x \in C. \text{degree}_3(x)) \\
\text{Moebiusstrip} &\Leftrightarrow \neg \text{Planar} \wedge \exists B. \text{Face}(B) \wedge \forall C. \text{Face}(C) \rightarrow B = C \vee \text{Hex}(C)
\end{aligned}$$

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:

Theorem 1 (Kuratowski). *A finite graph is **planar** (i.e. a graph that can be embedded in the plane) if and only if it does not contain a minor that is a subdivision of K_5 (the complete graph on five vertices) or $K_{3,3}$ (complete bipartite graph on six vertices, three of which connect to each of the other three).*

With a model checker like [1], we can then check whether a chemical graph (e.g. given by an MDL MOLfile⁵) falls into a graph class defined by a MSOL theory.

We can also check the subclass relation between two graph classes defined by MSOL theories, which corresponds to logical entailment. Although in general undecidable, logical entailment in second-order logic can be checked with the automated theorem prover LEO-II [3], available at <http://www.ags.uni-sb.de/~leo/>.

How can this specification of graph classes be related to existing ontologies of molecules such as ChEBI [7] 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 a *grounded ontology*, which means that class names like fullerene are equipped with a formal MSOL specification, which is then subject to automatic analysis such as model and subclass checking. The general correspondences are as follows:

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

To enable a formal interlinking of these different layers, we need a KR framework that provides strong support for logical heterogeneity, which we sketch next.

4.2 A very brief sketch of the Hyperontology framework

The hyperontology framework harnesses the fact that several heterogeneous logics and formalisms are used for designing ontologies. This approach is based on the theory of institutions (i.e. abstract model theory) and formal structuring techniques from algebraic specification theory. The features most relevant for the present application are the following, paraphrasing [14]:⁶

- The ontology designer can use OBO or OWL to specify most parts of an ontology, and can use first-order (or even higher-order) logic where needed.
- The overall ontology can be assembled from (or split up into) semantically meaningful parts (‘modules’) that are systematically related by structuring mechanisms supporting re-use of such parts in different settings.
- Here, various concepts of ‘ontological module’ are covered, including simple imports (extensions) and union of theories, as well as conservative and definitional extensions.

⁵ see http://en.wikipedia.org/wiki/Chemical_table_file

⁶ For technical detail and extensive discussion we have to refer to [14].

- Structuring into modules is made *explicit* in the ontology.
- Institution theory provides ‘logic translations’ between different ontology languages, translating the syntax and semantics of different formalisms. These translations allow in particular the ‘borrowing’ of reasoning and editing tools from one logic to another.
- Re-using (parts of) ontologies whilst renaming (parts of) the signature is handled by *symbol maps* and *hiding symbols*.
- The approach allows heterogeneous refinements: it is possible to prove that an ontology O_2 is a refinement of another ontology O_1 , formalised in a different logic. For instance, one can check if a domain ontology is a refinement of (a part of) a foundational one.

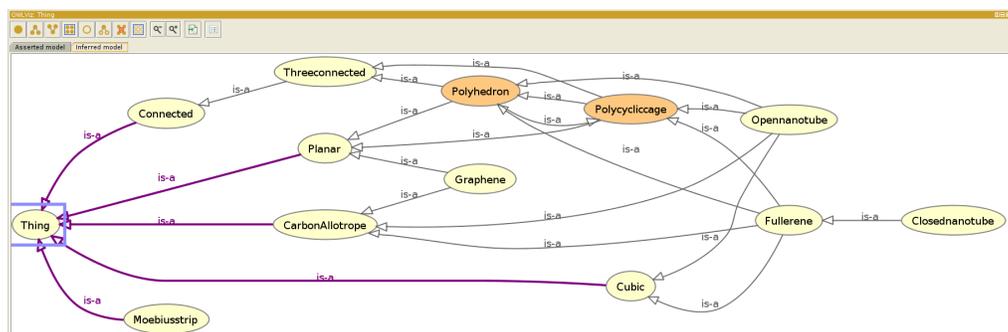
4.3 Information flow between ontological and graph layers

We investigate a threefold information flow between the ontological modelling and the graph-theoretical modelling of chemical graphs based on monadic second order logic (MSOL).

1. An asserted subsumption of classes in the chemical ontology is checked against corresponding MSOL theories. If the graph-theoretical properties are not compatible, a subsumption should be removed, which corresponds to a *debugging step* in the ontology development process.
2. If an implication between graph-theoretical characterisations associated with molecule classes can be established, this suggests a subsumption in the ontological hierarchy, and can be interpreted as an *abductive reasoning step* in the ontology.
3. Finally, giving a set of example molecules (given e.g. by MOL datafiles), by *inductive reasoning*, we can also try to learn a corresponding graph class specified in MSOL.

We plan to integrate these modes of reasoning, as well as suitable reasoning tools, into our general hyperontology framework. 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.

An initial OWL ontology computed from the logical implications among the MSOL axiomatisations looks as follows:



5 Conclusions

Representation of structured objects such as molecules, and reasoning based on aspects of their structure, is still an area of active research and development for ontologists and chemoinformaticians. Chemical ontologies such as ChEBI [7] provide one solution to this problem through the careful manual classification of molecules into classes. 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 approaches, and examine an approach which uses the expressive power of monadic second-order logic (MSOL) to formalise properties that cannot be defined in OWL, and propose the use of the hyperontology framework to link the two heterogeneous formalisms.

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 like ChEBI, thus obtaining a “grounded ontology”, where OWL subclass relations can be verified or inferred by looking at the corresponding graph properties in MSOL.

Interesting future work would be the combination of MSOL knowledge and OWL knowledge in a hybrid reasoner that combines MSOL theorem proving and OWL classification, such that e.g. the OWL classification externally calls the MSOL prover, but minimises the number of calls. Perhaps performance can be improved by singling out efficient fragments of MSOL (for example, propositional dynamic logic would be a candidate).

Another point is the semi-automatic generation of MSOL theories from MOL file 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, such that disambiguation choices also influence subclass properties.

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.

Acknowledgements

Work on this paper has been supported by the DFG-funded collaborative research centre SFB/TR 8 ‘Spatial Cognition’. We thank Berthold Hoffmann, Hans-Jörg Kreowski and Ian Pratt-Hartmann for valuable discussions and comments.

References

1. ARNBORG, S. A general purpose msol model checker and optimizer based on boolean function representation. Tech. rep., The Royal Institute of Technology, Stockholm, Sweden, 1994.
2. BAKRY, R., VALLANT, R. M., UL HAQ, M. N., RAINER, M., SZABO, Z., HUCK, C. W., AND BONN, G. K. Medicinal applications of fullerenes. *International Journal of Nanomedicine* 2 (2007), 639–649.
3. BENZMÜLLER, C., PAULSON, L. C., THEISS, F., AND FIETZKE, A. LEO-II - A Cooperative Automatic Theorem Prover for Classical Higher-Order Logic (System Description). In *IJCAR* (2008), A. Armando, P. Baumgartner, and G. Dowek, Eds., vol. 5195 of *Lecture Notes in Computer Science*, Springer, pp. 162–170.
4. CAETANO, E. W. S., FREIRE, V. N., DOS SANTOS, S. G., GALVAO, D. S., AND SATO, F. Möbius and twisted graphene nanoribbons: stability, geometry and electronic properties. *The Journal of Chemical Physics* v128, 164719 (2008).
5. CANCEILL, J., CHAMBRON, J.-C., COLLET, A., DIETRICH-BUCHECKER, C., DURST, H., DUTASTA, J.-P., KOHNKE, F., LOZACH, B., MATHIAS, J.-P., MISUMI, S., SAUVAGE, J.-P., STODDART, J., TOMALIA, D., ZIMMERMAN, S., CHAMBRON, J.-C., DIETRICH-BUCHECKER, C., AND SAUVAGE, J.-P. From classical chirality to topologically chiral catenands and knots. In *Supramolecular Chemistry I Directed Synthesis and Molecular Recognition*, vol. 165 of *Topics in Current Chemistry*. Springer Berlin / Heidelberg, 1993, pp. 131–162.
6. COURCELLE, B., AND ENGELFRIET, J. *Graph structure and monadic second-order logic—A language theoretic approach*. Cambridge University Press, 2011. Forthcoming.
7. DE MATOS, P., ALCÁNTARA, R., DEKKER, A., ENNIS, M., HASTINGS, J., HAUG, K., SPITERI, I., TURNER, S., AND STEINBECK, C. Chemical Entities of Biological Interest: an update. *Nucl. Acids Res.* 38 (2010), D249–D254.
8. HABEL, A., AND PENNEMANN, K.-H. Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science* 19, 2 (2009), 245–296.

9. HABEL, A., AND RADKE, H. Expressiveness of graph conditions with variables. *ECEASST 30* (2010).
10. HAN, D., PAL, S., LIU, Y., AND YAN, H. Folding and cutting dna into reconfigurable topological nanostructures. *Nature Nanotechnology* 5 (2010), 712–717.
11. HASTINGS, J., DUMONTIER, M., HULL, D., HORRIDGE, M., STEINBECK, C., SATTLER, U., STEVENS, R., HÖRNE, T., AND BRITZ, K. Representing chemicals using OWL, description graphs and rules. In *Proc. of OWLED* (2010).
12. KONYK, M., DE LEON, A., AND DUMONTIER, M. Chemical knowledge for the semantic web. In *Proceedings of Data Integration in the Life Sciences (DILS2008)* (Evry, France, 2008), vol. LNBI 5109, Lecture Notes in Computer Science, pp. 169–176.
13. KUTZ, O., LUTZ, C., WOLTER, F., AND ZAKHARYASCHEV, M. \mathcal{E} -Connections of Abstract Description Systems. *Artificial Intelligence* 156, 1 (2004), 1–73.
14. KUTZ, O., MOSSAKOWSKI, T., AND LÜCKE, D. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis* 4, 2 (2010), 255–333. Special Issue on ‘Is Logic Universal?’.
15. LUTZ, C., AND MILICIC, M. A tableau algorithm for DLs with concrete domains and GCI. *Journal of Automated Reasoning* 38, 1–3 (2007), 227–259.
16. RZEPA, H. Molecular möbius strips and trefoil knots. Last accessed 8 May 2011.
17. SHEREMET, M., TISHKOVSKY, D., WOLTER, F., AND ZAKHARYASCHEV, M. A Logic for Concepts and Similarity. *J. of Logic and Computation* 17, 3 (2007), 415–452.
18. TRINAJSTIC, N. *Chemical graph theory*. CRC Press, Florida, USA, 1992.
19. WEISSTEIN, E. W. Polyhedral graph, 2011. Last accessed 8 May 2011.
20. WESTER, M. J., POLLOCK, S. N., COUTSIAS, E. A., ALLU, T. K., MURESAN, S., AND OPREA, T. I. Scaffold Topologies. 2. Analysis of Chemical Databases. *Journal of Chemical Information and Modeling* 48, 7 (2008), 1311–1324.