Modeling Abstract Algebra as an OWL Ontology (Student Abstract)

Michael Vance
University of St. Thomas, 2215 Summit Ave, St. Paul, MN 55105
samsonv12@icloud.com

Abstract

Description logic ontologies serve to model classifications and structural relationships, and to represent and reason about domain knowledge. Modeling the basic classification of abstract algebraic structures as an ontology demonstrates the difficulties presented by their logical semantics and sheds light on the limitations to accurately model further topics in algebra and related mathematical domains.

Introduction

The Web Ontology Language (OWL) 2 DL is the fragment of OWL 2 corresponding to the class of description logics (DLs), which are decidable fragments of first order logic, limited to only unary and binary predicates. Thus, an ontology serves to model a domain of interest by object classes and binary relationships between objects. Furthermore, a reasoner is capable of interpreting given axioms to classify individuals and answer queries within the domain (Baader et al. 2017). This last capability is particularly interesting to the mathematical domain of abstract algebra.

Mathematical concepts are sparse in the literature of ontologies. Only one other project was found categorizing mathematical vocabulary, but not implementing the concepts themselves (Nevzorova et al. 2014). This project instead intends to model the concepts and definitions of the algebraic domain, using the $\mathcal{SHIQ}$ DL profile, meaning it includes transitive roles ($\mathcal{S}$) for relations like equality, role hierarchies ($\mathcal{H}$) for structure classification, role inverses ($\mathcal{I}$) and qualified number restrictions ($\mathcal{Q}$) for various properties.

Abstract algebra is a field of pure mathematics that serves to classify operated sets into various structure types with similar properties. For example, the familiar set of integers, operated by the standard addition and multiplication, is the prototype of the "commutative ring" structure. Any other set also classified as a commutative ring will also satisfy the same (or analogous) theorems that the integers satisfy. An ontology of this domain, in junction with the reasoner capabilities, could model and decide the classification of any proposed set and infer the theoretic implications that follow therefrom, creating a knowledge base for algebra that can be integrated with and applied to other mathematical domains and their practical purposes as well. An example query could be whether elements of Structure A (which may serve a specific applied purpose) satisfy the Zero Product property (that for elements $a,b,$ if $ab=0$ then either $a=0$ or $b=0$), which is used often to solve real-valued polynomials. The reasoner would classify Structure A, and if it is a "domain" or "field", it satisfies the property such that a proposed equation within the structure can be solved thereby.

However, in the basic classification modeled using OWL 2 ontology editor, Protégé 5.5.0, certain algebraic properties presented difficulty in accurately and efficiently modeling via the semantics of the language. This could prove problematic in the modeling and application of algebraic domains and the greater mathematical paradigm within the constraints of description logics.

Primer on Algebra

A brief overview of the ontology domain is necessary to discuss the issues further. Structures consist of a set of elements, often but not necessarily numbers, and either one or two closed operators, that is, the operators cannot generate a new element outside the set. Operators need not be the standard addition and multiplication. The first property considered in classification is commutativity: that for any operator * and two elements, a and b, the equality $a*b=b*a$ holds. Next is associativity. Similar to commutativity, for any operator * and three elements $a,$ b and c, the equality $a*(b*c)=(a*b)*c$ holds, where parentheses denote order of operation. An operator's identity is the unique element which, when operated with another element, generates that latter element; in variable notation, some identity e such that $e*a=a*e=a$ for all elements a. Another property is invertibility, that an element has some unique element for which the operated product of the two equals the identity; in variable, for some element u, an inverse v such that $u*v=v*u=e$. For structures with two operators, which are labeled one additive and one multiplicative, the two must satisfy the distributive property: for two operators * (multiplicative) and & (additive) and elements $a,$ b and c, the equality

$a*(b\&c)=(a*b)\&(a*c)$ holds (Cuoco and Rotman 2013). There are a total of four properties considered in single operated structures and nine in double operated ones, by which the satisfaction or not of properties classifies the structure.

As an example, the integers mentioned above classify as a commutative ring, satisfying all but one of the nine properties: multiplicative invertibility. Commutativity, associativity and distributivity are confirmed by examples. The additive identity is 0, to which any integer can add via its inverse, the negative. The multiplicative identity is 1. Not every integer is capable of multiplying to 1 (in fact, only 1 and its negative are) so that property is not satisfied.

The Algebra Ontology

Modeled in this ontology are several prevalent algebraic structures and the classifying properties to define each. The design of the ontology is intended to determine these properties solely from operation table(s). However, in combination with the ubiquity of infinite sets (like the integers), the Open World Assumption (OWA) of the logical system does not allow conclusive reasoning from operation tables. Thus, in many cases, unless declared, the reasoner cannot classify a structure to be commutative, for example, but, having defined commutativity, will recognize an inconsistency if a non-commutative structure is declared commutative. In the case of these properties, which are defined by an operator over every element, this is circumventable by assuming a structure to be commutative unless proven otherwise.

The ontology's Expression class contains individuals with names like $a*b$, declared to be operated by the operator *, "paired" with elements a and b (order preserving), and evaluating to some single element. Properties defined in terms of related individual elements, the identity and inverse, are defined with the help of these expressions and inverse ($\mathcal{I}$) constructs. The modeled identity definition is limited by the OWA. A subclass of the Element class, it is defined with a maximum cardinality restriction on a superproperty "relates" governing pairings and evaluation. An expression that pairs the same element on both sides of the operator and evaluates to that very element ($a*a=a$) relates to the maximum cardinality of 1 element (necessitating $\mathcal{Q}$ constructs), an identity. However, the reasoner, under the OWA, cannot confirm that an expression does not relate more than is declared, and therefore cannot conclude an element to be an identity. Unlike the previous OWA issue, there are too many elements to assume all to be identities and sort through inconsistencies proving them otherwise. Thus, a user must themself declare the identity element and the reasoner can merely check consistency.

Another prevalent issue, in some cases avoidable and others potentially less so, appeared in the logical semantics. In the algebra definitions presented above, it was necessary to use variables to define many of the properties, something OWL cannot do because the additional expressivity would lead to undecidability. Translating into the logical capabilities of Protégé, the definitions are wanting of a relative clause type expression. For example, an attempt at commutativity used an object property "commutes" to link two expressions that should commute, $a*b$ and $b*a$. An expression is defined to be commutative if it equals the expression *which it commutes*. This italic clause is incompatible with the logical semantics, but an alternative solution was identified using more declarations. Some concepts might not have suitable alternatives, needing excessive amounts of new declarations or failing altogether to accurately model.

The seemingly most difficult concept to model is associativity, presently left undefined. The compound expressions necessary are compatible with the formulation above but quickly expand the number of declarations. Just as commutativity, involving two variables, raised issues with both the OWA and the semantics, associativity does so on a larger scale, involving three variables. Despite this difficulty, associativity seems the least important to model as it is a condition of all the modeled common structures.

Conclusion

The basic classification of algebraic structure is almost completely modeled as an OWL DL ontology using the Protégé editor, but various properties demonstrate limitations with OWL in discussing algebraic concepts. While solutions were found for many cases, some may be possible but not conceived and some may even be impossible. Nevertheless, the ontology can classify declared structures and decide whether they satisfy properties for use by algebraic applications therein. However, for some properties and concepts, the foregoing issues could pose a significant challenge in the modeling of algebra and the greater understanding of mathematics in ontologies and knowledge representation.

Acknowledgments

The project was advised by Dr. Arina Britz of Stellenbosch University, South Africa.

References

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Cuoco, A. and Rotman, J. 2013. *Learning Modern Algebra: From Early Attempts to Prove Fermat's Last Theorem*. Washington D.C.: The Mathematical Association of America.

Nevzorova, O.; Zhiltsov, N.; Kirillovich, A.; and Lipachev, E. 2014. *OntoMathPRO Ontology: A Linked Data Hub for Mathematics*. Switzerland: Springer International Publishing.