Toward an Ontology for Finite Algebras

Bella Manoim
ASC Laboratory
Reem-Kayden 110
Bard College
Annandale-on-Hudson, NY 12504
bm458@bard.edu

Robert W. McGrail
ASC Laboratory
Reem-Kayden 110
Bard College
Annandale-on-Hudson, NY 12504
mcgrail@bard.edu

ABSTRACT

The authors present a case for an ontology of finite algebras. This vocabulary is a direct response to the limitations of the formats employed by first-order model searchers, such as Mace4, and specialized software, such as UACalc. It will support a semantically rich format for algebra storage and interchange intended to improve the efficiency of computational discovery processes in universal algebra. The class of finite quandles is considered as a case study in order to understand some of the challenges of designing such a knowledge base.

1. REPRESENTING ALGEBRAS

Current research projects within the Bard College Laboratory for Algebraic and Symbolic Computation (ASC) concern finite quandles [6], a class of algebras [2] inspired by the crossover arithmetic of three dimensional knots [3]. A finite algebra $\mathbf{A}$ is a pair $\langle A, F\rangle$, where A is a finite set and F a set of finitary operations on A. The finite algebra $\mathbf{A}$ is a quandle when F consists of a single binary operation $*$ that satisfies idempotence, right cancellation, and right self-distributivity. That is, for any $a, b, c \in A$,

- $a * a = a$;
- $a * c = b * c$ implies $a = b$; and
- $(a * b) * c = (a * c) * (b * c)$.

For example, Figure 1 is the Cayley table of the Tait quandle.

$*$	0	1	2
0	0	2	1
1	2	1	0
2	1	0	2

Figure 1: The Tait Quandle $\mathbf{Q} = \langle\{0,1,2\},\{*\}\rangle$

1.1 Algebra Formats

ASC lab researchers employ many symbolic software systems during the mathematical discovery process. The first-order model searcher Mace4, in conjunction with the other LADR-based programs [8], eases the process of finding and isolating finite algebras with particular properties. Its native format represents algebras as legal Prolog terms [10]. For example, Figure 2 displays the Mace4 version of the Tait quandle of Figure 1. Mace4 algebras can be read directly by SWI-Prolog and easily translated into expressions in the Mathematica computer algebra system [11]. As a consequence, most of the ASC lab's custom quandle software is designed in one of these two languages.

```
interpretation(3,[number=5, seconds=0],[
       function(*(_,_),[
                  0,2,1,
                  2,1,0,
                  1,0,2])
]).
```

Figure 2: Mace4 Version of the Tait Quandle

The UACalc system [4] can determine certain important characteristics of an algebra, such as its congruence and subalgebra lattices. It uses a specialized XML [1] format with no accompanying schema or documentation. The markup only records an algebra's operations in table form without the results of any UACalc computations. It is clear that the sole purpose of this format is to serialize algebras to a file.

1.2 Sharing Algebras

The effective sharing of algebras between researchers, rather than between software systems, requires a more comprehensive approach. Indeed, a thoughtful solution could reap great rewards. For example, one of the more important quandles discovered by ASC lab members was the fruit of a Mace4 run that lasted for several days. Making this algebra publicly available allows others to leverage this investment of computing time. A natural, distributed sharing method is simply to post algebras in these formats on the World Wide Web.

However, neither the Mace4 nor the UACalc formats add any unique identifiers to the internet search lexicon, so even reasonably sophisticated internet searches are likely to isolate few algebras in these forms. Another weakness of both the Mace4 and UACalc formats is the lack of any notations to indicate classification or other properties. For example, the Mace4 version of the Tait quandle of Figure 2 makes no indication that it is a quandle.

2. ALGEBRA ONTOLOGY

The authors of this work propose an ontology for finite algebras that will specify classes of universal algebras, such as groups, loops, and quasigroups, and properties applicable to finite algebraic systems, such as solvability for groups, distributivity for lattices, and tractability for general algebras. It will also record known logical relationships between classes, between classes and properties, and between properties.

In order to ensure a distributed and extensible knowledge base it will be designed using the OWL [9] plugin for Protégé [5]. Thus a lattice theory expert, for example, will be able to contribute directly to the lattice vocabulary by simply posting some OWL online. She may also serialize an individual algebra along with its major classifications and properties to a variety of semantically rich formats, such as RDF [7], for storage and internet searches.

2.1 Design Challenges

Of course, classes of algebras will be represented by OWL classes and individual algebras by instances. However, the ontology must reflect that not all classes of algebras are the same to universal algebraists. This issue is illustrated below in the realm of finite quandles (see Figure 3).

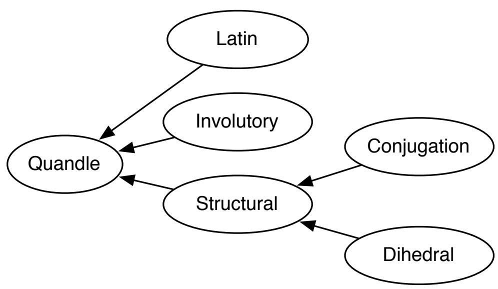

Figure 3: Some Classes of Quandles

A quandle $\mathbf{Q}$ is Latin if $*$ satisfies left cancellation: For any $a, b, c \in Q$, $a*b = a*c$ implies $b = c$. Thus the class of Latin quandles is a first-order axiomatizable subclass of the class of quandles. Such subclasses are among the most natural to universal algebraists. Hence this subclass relationship corresponds to direct subclassing in OWL (Figure 3).

Other subclasses arise via more complex constructions. As an example, consider the class of conjugation quandles. Given a group $\mathbf{G} = \langle G, \{\circ, (\ )^{-1}\}\rangle$, one forms a quandle by letting

$$g * h = h^{-1} \circ g \circ h,$$

for $g, h \in G$. Such a structure is called a group quandle. A conjugation quandle $\mathbf{Q}$ is a subquandle of $\langle G, \{*\}\rangle$ for some group $\mathbf{G}$. Classes constructed via similar means are rarely first-order axiomatizable over the target language. An abstract class for subclasses that arise via some structural method is added to the ontology. This class ("Structural" in Figure 3) acts as an umbrella term for many well-known subclasses of quandles, such as conjugation and dihedral quandles.

One may demonstrate via an elementary yet nontrivial proof that Latin quandles form a subclass of conjugation quandles. However, this is neither an immediate consequence of first-order subclassing, nor the results of a restriction of the conjugation construction to some subclass of finite groups. Therefore, this is not indicated using ordinary subclassing, but instead is specified as a property of Latin quandles.

3. ACKNOWLEDGEMENTS

The authors thank Sven Anderson and S. Rebecca Thomas for several excellent suggestions for improvement.

4. REFERENCES

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, editors. *Extensible Markup Language (XML) 1.0, 4th Edition*. W3C Recommendation, August 2006.

[2] S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Springer Verlag, Berlin, 1981.

[3] P. Cromwell. *Knots and Links*. Cambridge University Press, 2004.

[4] R. Freese and E. Kiss. The UACalc universal algebra calculator. http://www.uacalc.org.

[5] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, January 2003.

[6] D. Joyce. A classifying invariant of knots, the knot quandle. *Journal of Pure and Applied Algebra*, 23:37–66, 1982.

[7] F. Manola and E. Miller, editors. *RDF Primer*. W3C Recommendation, February 2004.

[8] W. McCune. Mace4 reference manual and guide. Technical Report ANL/MCS-TM-264, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 2003.

[9] D. L. McGuinness and F. van Harmelen, editors. *OWL Web Ontology Language Overview*. W3C Recommendation, February 2004.

[10] J. Wielemaker. *SWI-Prolog 5.6 Reference Manual*, April 2006.

[11] S. Wolfram. *The Mathematica Book*. Wolfram Research, Inc., 5th edition, 2003.