The Algebraic Essence of K-Rep

Frank J. Oles, Eric K. Mays and **Robert A. Weida**

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598
oles@watson.ibm.com

1 Introduction

The K-Rep system is an industrial strength knowledge representation system, based on description logic and developed at IBM Research. Its principal application has been to the creation of large medical lexicons as part of clinical information systems. Terminologies involving more than 100,000 separately defined terms have been contructed with K-Rep. The use K-Rep in this arena is described in [MW+]. In this paper, we wish to focus on how algebraic techniques have played a fundamental role in the implementation of K-Rep.

Usually, KL-ONE type languages (see [BS85]), such as K-Rep, are explained informally as logics or they are given an informal set-theoretic semantics. To be more precise about it, a knowledge base (KB) expressed in such a language is treated as a set of logical axioms. Given such a set of axioms, we can ask about those assertions that are true in all models. This leads to treating questions of subsumption of concepts with respect to a KB as questions of subset inclusion of corresponding sets in all models of the KB.

There can be another point of view, one that we took in implementing K-Rep, one that leads to a structural – as opposed to a logical or model-theoretic – approach to computing subsumption. If a description logic can be used to describe a taxonomy or classification scheme consisting of concepts related by subsumption, then what we wish to do is to realize the subsumption relation as the partial order of a meet-semilattice. According to this view, a KB is a presentation via generators and defining relations of an *algebra of semantic concepts.* Thus, we bring to bear on knowledge representation the machinery of the field of universal algebra. In the case at hand, the kind of algebra that results has no established name, but it is a meet-semilattice with additional structure. We call the partial ordering of the algebra *semantic subsumption.* We have been able to use the structure of the algebra denoted by a KB to efficienctly represent concepts in a way that facilitates the computation of subsumption.

Historically, what happened with K-Rep was that an implemented early version existed and proved useful. In that version, subsumption was computed by means of a structural analysis of the concepts involved. Then one of us (Oles) conversant with formal semantics and universal algebra joined the K-Rep project. At that point, the aim was to extend the language by adding the *some construct:* $\exists\ R : C$. Because the algebra that would enter into the structural analysis was not immediately clear, this proved to be nontrivial. From our understanding of set-theoretic models of knowledge bases, we developed a set of algebraic axioms to use in analyzing the structure of concepts. We then massaged these into the code. The result was a very successful system that had added functionality and could be seen to valdiate the mathematical idea of regarding a K-Rep KB as a presentation of an algebra. However, an attempt to give a really precise, formal mathematical semantics for K-Rep is something we are only now trying to do. In so doing, we have come to realize that the existing formalisms of universal algebra aren't quite powerful enough to justify everything we have apparently done in practice! Thus, some ideas that we wish were theorems are only conjectures, but the implemented system gives us confidence that the conjectures are true.

A more complete exploration of the idea of regarding a knowledge base as a presentation of an algebra in a setting where all the conjectures do indeed turn out to be theorems can be found in [O96]. There the reader can get a better appreciation of what it means to give a formal proof that an implementation technique in this area is correct. The computer scientist who needs to brush up on universal algebra can see [W92].

2 KRP

The entire K-Rep system is a bit too cumbersome to discuss here. More to the point, some of its features have no bearing on the issues we wish to take up. Consequently, we will now introduce a description logic *KRP,* standing for *K-Rep Prime,* which has all the features reminiscent of KL-ONE found in the K-Rep system. In addition to avoiding the nuts and bolts of the K-Rep system, by focusing on *KRP*, we will not address the nature of *facets,* which may be attached to a concept name or a role name, because neither are they supposed to affect how a newly introduced concept is *classified,* i.e., how it is to be inserted into an existing taxonomy.

For the language *KRP*, we posit the existence of two countably infinite disjoint sets of identifiers: **I**, to be used as names of concepts, and **R**, to be used as names of roles. Of course, K-Rep itself doesn't force users into this kind of straightjacket, which is reminiscent of ancient versions of FORTRAN, but it is a convenient theoretical simplification. The following is a specific abstract syntax for *KRP*, in which the top level nonterminal **B** defines knowledge bases, **A** defines terminological axioms, **C** defines syntactic concepts, and **N** is the set of nonnegative integers:

B ::= `emptyKB` | **B** ; **A**

A ::= `define-primitive-concept` **I C**
| `define-concept` **I C**
| `disjoint-primitives` **I I**

C ::= **I** | $\top$ | $\bot$ | $(\mathbf{C} \wedge \mathbf{C})$ | $(\forall\, \mathbf{R} : \mathbf{C})$
| $(\exists\, \mathbf{R} : \mathbf{C})$ | (`atmost` **N C**)
| (`atleast` **N C**)

Here `emptyKB` is the KB corresponding to the empty sequence of terminological axioms. In general, KB's are *sequences*, not just sets of terminological axioms, to reflect the idea that the representation of a knowledge base is to be constructed incrementally as terminological axioms are processed one by one. Terminological cyles are not supported. The first two terminological axioms serve to define concept names, and the third declares that two primitive concepts are to be disjoint. The constructs for creating compound concepts are fairly standard.

A *well-formed knowledge base* is

1. one such that each identifier in the KB that names a concept that appears in the knowledge base (a) makes its first appearance in a terminological axiom that does not define it in terms of itself, and (b) has no other definition in the KB, and
2. one in which the identifiers in a `disjoint-primitives` axiom have indeed been earlier defined as naming primitive concepts.

We will not attempt to assign meaning to KB's that are not well-formed. We do expect that, if one well-formed knowledge base is a reordering of the terminological axioms of another, then both will have the same semantic meaning.

From the preceding description of syntactic concepts, we extract the signature Ω, whose operators are as follows:

1. the two constants $\top$ and $\bot$,
2. the binary operator $\wedge$,
3. for each role name R, the two unary operators $\forall\, R :$ and $\exists\, R :$, and
4. for each nonnegative integer n, the two unary operators `atmost` n and `atleast` n.

Thus, the collection of all syntactic concepts of *KRP* can be described as the free Ω-algebra generated by **I**.

3 Algebraic Laws Used in Implementation

Given any set-theoretic model of a well-formed KB expressed in *KRP*, there are many ways to make it into an Ω-algebra. The reason that there are many ways is that we are free to interpret the roles not explicitly mentioned in the KB as binary relations on $\top$ in any way we want.

Ω-algebras that arise as models of KB's are of central interest. What algebraic laws are true in all such models? Below is a list of equational laws, schema for equational laws, and, at the end, one slightly non-equational law that hold in all set-theoretic models of *KRP* knowledge bases. Some of the laws may not look like equations, but they really are, because the first three laws are the meet-semilattice axioms, which enable us to regard $C \leq D$ as a shorthand for $C = C \wedge D$. In the following list, $C_1, \cdots, C_n, C, D$, and E are algebra elements (i.e., possible set-theoretic interpretations of concepts), R is a role name, and n and m are nonnegative integers.

1. $(C \wedge D) \wedge E \;=\; C \wedge (D \wedge E)$,
2. $C \wedge D \;=\; D \wedge C$,
3. $C \wedge C \;=\; C$,
4. $C \;\leq\; \top$,
5. $\bot \;\leq\; C$,
6. $\forall\, R : (C \wedge D) \;=\; (\forall\, R : C) \wedge (\forall\, R : D)$,
7. $\exists\, R : (C \wedge D) \;\leq\; \exists\, R : C$,
8. $\forall\, R : \top \;=\; \top$,
9. $\exists\, R : \bot \;=\; \bot$,
10. $(\forall\, R : C) \wedge (\exists\, R : D) \;=\; (\forall\, R : C) \wedge (\exists\, R : (C \wedge D))$,
11. `atmost` $0\ R \;=\; \forall\, R : \bot$,
12. `atleast` $0\ R \;=\; \top$,
13. `atmost` $n\ R \;\leq\;$ `atmost` $m\ R$, if $n \leq m$,
14. `atleast` $n\ R \;\leq\;$ `atleast` $m\ R$, if $m \leq n$,
15. (`atmost` $n\ R$) $\wedge$ (`atleast` $m\ R$) $=\; \bot$, if $n < m$,
16. $(\exists\, R : C) \wedge$ (`atmost 1 R`) $\leq\; \forall\, R : C$, and finally
17. $(\exists\; R \;:\; C_1) \wedge \cdots \wedge (\exists\; R \;:\; C_n) \;\leq$ `atleast` $\kappa(\{C_1, \cdots, C_n\})\ R$, where $n > 0$.

    The function κ depends on the algebra under discussion, but its definition is uniform across all Ω-algebras. κ takes as argument a nonempty finite set $\mathcal{E}$ of algebra elements and produces a natural number. If $\bot \in \mathcal{E}$, then $\kappa(\mathcal{E}) = 0$. Otherwise, $\kappa(\mathcal{E})$ is the largest integer k such that there exists a partition $S_1, \cdots, S_k$ of $\mathcal{E}$ satisfying

    (a) for all $i \in \{1, \cdots, k\}$, $\bigwedge S_i \neq \bot$, and
    (b) for all $i, j \in \{1, \cdots, k\}$ with $i \neq j$, $(\bigwedge S_i) \wedge (\bigwedge S_j) = \bot$.

The authors would like to thank Peter Selinger for simplifying the original version of the seventh law in the list above.

Using ASAMAL as an acronym for All Some At Most At Least, define an *ASAMAL-meet-semilatice* to be *any* Ω-algebra – not just those algebras of sets that arise as models of *KRP* knowledge bases – satisfying the 17 axioms and axiom schemas listed above. ASAMAL-meet-semilattices are apparently not a variety (i.e., equationally defined class) of algebras, atthough we do not have a proof of this fact. However, we now see in retrospect that we pretended that they were a variety. In particular, we pretended that we knew that each set of generators paired with a set of defining relations uniquely determined a unique most general ASAMAL-meet-semilattice generated by those generators and satisfying those defining relations. Formal proofs that the desired theorems are true can almost certainly be extracted from our implementation techniques.

As long as we assume that ASAMAL-meet-semilattices presented by *KRP* knowledge bases exist, it is easy to see that every element of such an algebra can be put into a normal form. Whether or not one concept subsumes another can be determined by comparing the normal forms. Space restrictions prevent us from going into detail, but the normal forms involve antichains of concepts and the Smyth ordering from domain theory pops up.

Our main unproved conjecture is that every ASAMAL-meet-semilattice arises as an algebra of sets. From this, it should follow that any subsumption relation true in all models of a knowledge base can be seen to be true using the axioms above together with the assertions provided by the KB. Another conjecture is that every ASAMAL-meet-semilattice presented by a *KRP* knowledge base is actually a lattice. This is not too surprising a conjecture in that it can be translated as saying that any two concepts have a *least common subsumer.* A careless reading of Proposition 1 of [CBH] might lead one to believe that pairs of concepts in description logics supporting binary conjunction always have a unique least common subsumer when they have at least one common subsumer (e.g., $\top$). This conclusion is unwarranted. The real problem in proving that two concepts have a least common subsumer is to show the existence of some least common subsumer when the two concepts have infinitely many common subsumers. We do not expect to find a simple two-line argument showing that description logics have least common subsumers of arbitrary pairs of concepts, because it would probably also show all bounded meet-semilattices are lattices, which is false.

References

[BS85] Brachman R.J., and J.G. Schmolze, "An overview of the KL-ONE knowledge representation system," *Cognitive Science* 9 (1985), pp. 171-216.

[CBH] Cohen, W.W., A. Borgida, and H. Hirsh, "Computing Least Common Subsumers in Description Logics," AAAI-92, pp. 754-760.

[MW+] Mays, E., R. Weida, R. Dionne, M. Laker, B. White, C. Liang, and F.J. Oles, "Scalable and Expressive Medical Terminologies," to appear in *Proceedings of the 1996 AMIA Annual Fall Symposium.*

[O96] Oles, F.J., "An Application of Lattice Theory to Knowledge Representation," submitted for publication.

[W92] Wechler, W., *Universal Algebra for Computer Scientists,* Springer-Verlag, Berlin, 1992.

On the semantics of epistemic description logics (Extended Abstract)

Riccardo Rosati
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
email: rosati@dis.uniroma1.it

Abstract

The modal description logic $\mathcal{ALCK}$ both constitutes a promising framework for reasoning about actions and allows for the formalization of several non-first-order aspects of KR systems based on DLs. However, other nonmonotonic features of DL-based KR systems, in particular role and concept closure inside the knowledge base, lack an intuitive formalization in this modal framework. To overcome these difficulties, we propose a modification of the semantics for $\mathcal{ALCK}$, which consists in allowing selective minimization of primitive concepts and roles, thus providing for a correct formalization of the notion of role and concept closure.

1 Introduction

Recent research in description logics (DLs) has dealt with the problem of extending the language of such formalisms with modal operators, increasing their expressive power in order to allow for the formalization of notions like knowledge, belief, time, intention and others [Baader and Laux, 1995; Baader and Ohlbach, 1995]. Specifically, a *nonmonotonic* modal extension of DLs has been proposed in [Donini *et al.*, 1992; 1994; 1995]. Such an extension, the autoepistemic description logic $\mathcal{ALCK}$, both constitutes a promising framework for reasoning about actions [De Giacomo *et al.*, 1996] and allows for the formalization of several non-first-order aspects of KR systems based on DLs [Borgida *et al.*, 1989; MacGregor, 1988], like procedural rules, defaults, a weak form of concept definition and some forms of closed-world reasoning [Donini *et al.*, 1994; 1995]. However, other nonmonotonic features of DL-based KR systems, in particular role and concept closure *inside* the knowledge base, lack an intuitive formalization in this modal framework. We illustrate such notions through two examples.

Example 1 (Concept closure). Let Ψ_1 be the following $\mathcal{ALCK}$ knowledge base:

$$\texttt{doctor(Paula)}$$
$$\texttt{lawyer(Marc)}$$
$$\forall\texttt{CHILD.has-blue-eyes(Ann)}$$

Suppose we want to add to Ψ the following informal assertion A: "One of Ann's children is one of the *known* doctors". Now, since Paula is the only known doctor, we want to be able to conclude that Paula is one of Ann's children, and hence

$$\Psi_1 \cup \{A\} \models \texttt{has-blue-eyes(Paula)}$$

The intuitive formalization of A in $\mathcal{ALCK}$ is the following assertion A':

$$A' = \exists\texttt{CHILD.}\mathbf{K}\texttt{doctor(Ann)}$$

Unfortunately, the original $\mathcal{ALCK}$ semantics does not capture the intended meaning of $\Psi_1' = \Psi_1 \cup \{A'\}$. Informally, the closure is formalized through the minimization of the knowledge about the concept doctor. However, in the $\mathcal{ALCK}$ semantics *every* concept and role is minimized, and, in this particular case, the interaction between the minimization of the role CHILD and the minimization of the concepts in