UNDERSTANDING BASIC DL REASONING

Executive Summary

Network Inference is the leading innovator in enterprise software for applied inference and reasoning capabilities. Traditional enterprise application integration (EAI), enterprise information integration (EII) and vocabulary management systems are ripe with opportunities for these advanced inference and reasoning strengths. Today, Network Inference's flagship product Cerebra supports a variety of reasoning methods that are scientifically sound and complementary to one another. These methods include: T-Box reasoning, A-Box reasoning, representative A-Box reasoning (RDBMS queries) and datatype reasoning (D-Box). Cerebra represents the fusion of these methods into a unified enterprise solution that simplifies the complex problem of integrating data, process and business rules.

Description Logic View of the Universe

The description logic (DL) perspective contends that all things in the universe can be represented as an unspecified number of data elements, termed "instances", with every pair possibly related via any number of different directed edges, termed "properties" or "roles".

The field of description logic, however, generally concerns these instances and properties only indirectly. DL languages are concerned primarily with grouping the potentially infinite set of instances which could exist in the universe into sets, termed "classes" or "concepts". The power of DL is in its ability to combine and manipulate these classes in order to encode knowledge about the universe. Complex classes can be formed with the use of the boolean operators as well as existential and universal concept constructors, cardinality restrictions, etc. The resulting classes can then be used within axioms to assert that certain sets are necessarily empty, for example: no modern creature is both a mammal and a fungus (the intersection of the two is empty); a taxpayer is filing either singly or jointly (and thus the class of taxpayers doing neither is empty).

Understanding Boxed Ontology Sets

The ontology resulting from such class definitions and axioms can then be processed to produce sound, complete, and decidable inferencing results. Given any complex class definition, it can be determined whether the knowledge in the ontology implicitly forces this class to be empty, whether the subclasses and superclasses of the new class can be found, or any other deterministic answer set. Note that the previously mentioned examples take note only of classes and the potential role relations between members of those classes. All the knowledge encoded in this way is sometimes called a "terminology", which includes both the vocabulary of terms which can be used as well as the semantic meanings of those terms. As a consequence, these aspects of an ontology are termed the "terminological box" or T-Box.

A collection of knowledge which moves beyond general relations among classes and axioms that form a terminology may focus on instances and assertions that are possible among them. In addition to such knowledge about the sets themselves, data about specific instances which are explicitly known to exist can be asserted. Such assertions are actually much simpler than the complex semantic axioms of the T-Box. Beyond simply asserting that some instance exists or that two different names for an instance actually refer to the same thing, all such statements of either of the form "instance X is a member of class C" (where C may be any complex class construction) or "instances X and Y are related via role R." A collection of such statements about instances is termed an "assertional box" or A-Box.

There are two very important items to note about A-Box statements. First, A-Box axioms clearly require the vocabulary made available by a T-Box – without classes and roles no relevant statements can be made about any instances. Second, A-Box statements do not have any direct impact on the implications of a T-Box: the assertions that there exists an instance "John" of the class "Person" and an instance "Fido" of the class "Dog" and that John is related to Fido via the role "owns" does not imply that every Person owns a Dog or that every Dog has an owner. Nothing can be inferred beyond the (possibly unusual) state of affairs between these two specific instances.

A-Box assertions are useful for two purposes. In the simplest case, a set of A-Box data can be asserted and an inference engine can determine whether this data contradicts the semantics described in the associated T-Box. More robustly, an A-Box can be used to store a full collection of raw data for a system, and an advanced reasoner can retrieve collections of instances which meet certain criteria. Such an interface should form a true superset of the functionality provided in a traditional relational database, however in our case we can extend predicates to full complex class constructors. Further, relational databases generally assume a form of "negation as failure"--every predicate is explicitly either true or false. A DL ontology, however, groups predicates into those which must be true, those which must be false, and those which cannot be proven either true or false with available information.

T-Box reasoning can actually be used to provide a sensible simulation of A-Box assertions: each instance can be represented as a new class, and assertions about the instance can be translated to class axioms. The resultant reasoning provides much of the functionality of a true A-Box, however there are significant distinctions between a class and an instance. Chief among these is the fact that an instance represents exactly one element, while a class may contain zero, one, or more members. Reasoning about an instance as though it were a class thus results in "incomplete" inferences. More to the point, retrieval-style queries for classes are generally limited to a single variable, and generally allow only "provable" predicates. True A-Box reasoning coupled with a robust instance query interface allows for sound and complete results to complex expressive queries. Network Inference supports complete A-Box. In addition, for extremely large data sets, we provide a form of simulation for retrieval query that uses RDBMS instances – this is sometimes referred to as representative A-Box capabilities.

Future DL Directions and Pragmatic Enterprise Capabilities

Extremely expressive Description Logic languages sometimes provide an additional feature which forms a bridge between the separate worlds of T-Box and A-Box reasoning. "Nominals" (invoked using the "oneOf" construct in OWL) allow the creation of a class definition based on explicit enumeration of the class's elements. Nominals thus provide a way to form "closed world" realms within the open world universe of the T-Box. Such constructions do allow A-Box assertions to impact the T-Box (since T-Box axioms can now be written in terms of elements of the A-Box).

Straightforward T-Box and A-Box reasoning attract great interest in academic circles, however, in practice such systems have proven somewhat limited. It has become standard to extend DL languages with "datatypes". The semantic model is equivalent to adding a very large number of A-Box assertions to create and define memberships among new sets of instances for each of the possible integers, floating point numbers, and strings. For practical reasons, the actual assertion of this near-infinite axiom set is not performed; instead the A-Box is broken into a traditional instance A-Box and a separate software component specific to datatype reasoning, occasionally referred to as the "datatype box" or "D-Box." OWL and other DL languages often include special features to assist in working with ontologies containing datatypes (including implicit role range constraints to simplify ontology construction and visualization). Datatype optimizations allow for not only efficient reasoning, but also a more expressive language allowing the equivalent of nominals for data values. Because of this nominal bridge from the D-Box to the T-Box and the implementation independent of an A-Box reasoner, datatype reasoning is often included in T-Box-only reasoners. Network Inference fully supports D-Box reasoning capabilities.

Summary Conclusions

Advancements in reasoning capabilities, specifically description logic (DL) based approaches, hold much promise for a more adaptive data environment. Unlike XML, UML, and relational systems, data that has been modeled in these expressive formats (OWL-DL for example) allows enterprise users to execute complex, rules-driven applications on top of models rather than compiled code. Future directions will simultaneously incorporate additional advanced hybrid reasoning rules systems as well as simpler methods for leveraging the volumes of already existing corporate data. Network Inference will continue to lead the innovation, standardization, and adoption of advanced reasoning capabilities on all of these fronts.

About the Author

Robert Shearer

Mr. Shearer is a Lead Architect with Network Inference and the principal innovator of Cerebra's core description logic runtime infrastructure. He is an acknowledged commercial expert in advanced reasoning capabilities with a background in natural language processing. Mr. Shearer is an active member of the World Wide Web Consortium where he participates in the Resource Description Framework Data Access Working Group – helping to drive momentum towards commercially viable approaches for querying ontology. Aside from his product engineering responsibilities, Mr. Shearer devotes significant time working with Professor Ian Horrocks of the University of Manchester, also Chief Scientist at Network Inference, to research the next generation of capabilities for the Cerebra product line.

