Approximate Terminological Reasoning for the Semantic Web

Heiner Stuckenschmidt, Frank van Harmelen, Perry Groot
Vrije Universiteit Amsterdam

A strength of the current proposals for the foundational languages of the Semantic Web (RDF Schema, DAML+OIL), it that they are all based on formal logic. However, this reliance on logics is not only a strength but also a weakness. Traditionally, logic has always aimed at modeling idealized forms of reasoning under idealized circumstances. Clearly, this is not what is required under the practical circumstances of the Semantic Web. Instead, the following are all needed:

- reasoning under time-pressure
- reasoning with other limited resources besides time
- reasoning that is not "perfect" but instead "good enough" for given tasks under given circumstances
- reasoning-algorithms that do not behave as yes/no oracles, but that instead display anytime behaviour

It is tempting to conclude that symbolic, formal logic fails on all these counts, and to abandon that paradigm. However, research in the past few years has developed methods with the above properties while staying within the framework of symbolic, formal logic.

Approximation Techniques

Knowledge Compilation In order to avoid complexity at run-time, knowledge compilation aims at explicating knowledge hidden in a logical model in a pre-processing step. Derived facts are added to the original theory as axioms avoiding the need to deduct them again. In the case of ontological reasoning, implicit subsumption and membership relations are good candidates for a compilation. For example, implicit subsumption relations in a DAML+OIL ontology could be identified using a description logic reasoner, the resulting more complete hierarchy could be encoded in RDF schema and used by systems that do not have the ability to perform complex reasoning.

Language Weakening The idea of language weakening is based on the well-known trade-off between the expressiveness and the reasoning complexity of a logical language. By weakening the logical language a theory is encoded in, we are able to tread the completeness of reasoning against runtime. The logic that underlies DAML+OIL for example is known to be highly intractable, existing reasoners therefore use a slightly weaker logic that still allows to compute most deductions. This idea can be further extended by starting with a very simple language and iterating over logics of increasing strength supplementing previously derived facts.

Approximate Deduction Instead of modifying the logical language, approximations can also be achieved by weakening the notion of logical consequence. This can be done by restricting the size of definitions that are considered or by allowing failures on parts of the vocabulary used. On the semantic web, using the latter techniques will often even be necessary if agents that only share parts of their vocabulary. In this case approximate deduction can be used to compute at least those deductions that solely depend on the shared vocabulary instead of failing to derive anything.

Expected Benefits

Scalability: Approximate Terminological Reasoning can be used to reduce complex reasoning tasks to simpler one, e.g. by choosing a less expressive encoding language. Using different simplification techniques with increasing exactness, we can produce an anytime behaviour of terminological reasoning engines. While abandoning completeness, such a techniques will help to scale up logical reasoning to realistic application domains on the semantic web.

Robustness: Approximate Terminological Reasoning can be used to cope with erroneous, inconsistent or missing information for example by restricting the vocabulary of an application domain to a maximal handable subset. This ability will be very important if we assume a large scale use of ontologies on the semantic web that will be dominated by sloppy ontologies that claim neither completeness nor consistency.

Interoperability: Approximate Terminological Reasoning can help to compare, align and integrate heterogeneous ontologies in complex application scenarios. When sharing or re-using ontologies that are not designed to be used together, we will often face the situation where classes describe similar, but not equivalent sets of objects or they will structure the domain according to orthogonal dimensions. In these cases, we need approximate reasoning techniques in order to identify similar concepts and to relate heterogeneous concepts.