Utility and Feasibility of Reasoning beyond Decidability in Semantic Technologies

Sebastian Rudolph and Michael Schneider
Institute AIFB, Karlsruhe Institute of Technology, DE
rudolph@kit.edu
FZI Research Center for Information Technology, Karlsruhe, DE
schneid@fzi.de

Abstract. Semantic Web knowledge representation standards such as RDF and OWL have gained momentum in the last years and are widely applied today. In the course of the standardization process of these and other knowledge representation formalisms, decidability of logical entailment has often been advocated as a central design criterion. On the other hand, restricting to decidable formalisms inevitably comes with constraints in terms of modeling power. Therefore, in this paper, we examine the requirement of decidability and weigh its importance in different scenarios. Subsequently, we discuss a way to establish incomplete – yet useful – reasoning support for undecidable formalisms by deploying machinery from the successful domain of theorem proving in first-order predicate logic. While elaborating on the undecidable variants of the ontology language OWL 2 as our primary examples, we argue that this approach could likewise serve as a role model for knowledge representation formalisms from the Conceptual Structures community.

1 Introduction

Today, the Semantic Web serves as the primary testbed for practical application of knowledge representation. A plethora of formalisms for representing and reasoning with Web knowledge has been designed and standardized under the auspices of the World Wide Web Consortium (W3C). While the early days of this endeavor saw ad-hoc and semantically underspecified approaches, interoperability requirements enforced their evolution into mature logical languages with clearly specified formal semantics. In the process of defining more and more expressive such formalisms, an often-debated requirement is decidability of logical entailment, i.e. the principled existence of an algorithm that decides whether a body of knowledge has a certain proposition as a consequence. While it goes without saying that such an algorithm is clearly useful for all kind of querying or knowledge management tasks, results established back in the 1930s show that this property does not hold for all types of logics [6][24]. In particular, in many expressive knowledge representation formalisms (most notably first-order predicate logic), entailment is undecidable.
Hence, whenever a knowledge representation formalism is to be designed, the trade-off between decidability and expressivity has to be taken into account. An examination of the Semantic Web languages hitherto standardized by the W3C yields a mixed picture in that respect: logical entailment in the basic data description language RDF [13] and its light-weight terminological extension RDF Schema [4] is decidable (although already NP-complete in both cases). Within the OWL 2 language family [17], only the most expressive variant OWL 2 Full [21] is undecidable, whereas OWL 2 DL [16,15] as well as its specified sublanguages (called tractable profiles) OWL 2 EL, OWL 2 QL, OWL 2 RL [14] are decidable (the latter three even in polynomial time). On the other hand, for the rule interchange format RIF [12], only the very elementary core dialect RIF-Core [2] is decidable whereas already the basic logic dialect RIF-BLD [3] – and hence every prospective extension of it – turns out to be undecidable.

This small survey already shows that decidability is far from being a common feature of the standardized Semantic Web languages. However, extensive reasoning and knowledge engineering support is currently only available for the decidable languages in the form of RDF(S) triple stores or OWL 2 DL reasoners hinting at a clear practitioners’ focus on these languages.

In this paper, we argue that inferencing support is important and feasible also in formalisms which are undecidable and we provide an outlook how this can be achieved, referring to our recent work on reasoning in undecidable Semantic Web languages as a showcase. We proceed as follows: Section 2 will remind the reader of the important notions from theoretical computer science. Section 3 proposes a schematic classification of inferencing algorithms by their practical usefulness. Section 4 distinguishes cases where decidability is crucial to enable “failsafe” reasoning from cases where it may make sense to trade decidability for expressivity. After these general considerations, we turn to variants of OWL to demonstrate ways to provide reasoning support for undecidable Semantic Web formalisms. To this end, Section 5 gives an overview of OWL syntaxes and the associated semantics. Section 6 shows two different ways of translating OWL reasoning problems into first-order logic and Section 7 briefly reports on our recent work of employing FOL reasoners in that context. In Section 8 we discuss ramifications of our ideas for Common Logic. Section 9 concludes. An extended version of this paper with examples of reasoning in diverse undecidable languages is available as technical report [20].

2 Recap: Decidability and Semidecidability

Let us first recap some basic notions from theoretical computer science which are essential for our considerations. From an abstract viewpoint, a logic is just a (possibly infinite) set of sentences. The syntax of the logic defines how these sentences look like. The semantics of the logic is captured by an entailment relation $\models$ between sets of sentences $\Phi$ and sentences $\varphi$ of the logic. $\Phi \models \varphi$ then means that $\Phi$ logically entails $\varphi$ or that $\varphi$ is a logical consequence of $\Phi$. Usually, the logical entailment relation is defined in a model-theoretic way.