Verification of Context-Sensitive Knowledge and Action Bases

Diego Calvanese¹, İsmail İlkan Ceylan², Marco Montali¹, and Ario Santoso¹

¹ Free University of Bozen-Bolzano, Italy
lastname@inf.unibz.it
² Technische Universität Dresden, Germany
ceylan@tcs.inf.tu-dresden.de

Abstract. Knowledge and Action Bases (KABs) have been recently proposed as a formal framework to capture the dynamics of systems which manipulate Description Logic (DL) Knowledge Bases (KBs) through action execution. In this work, we enrich the KAB setting with contextual information, making use of different context dimensions. On the one hand, context is determined by the environment using context-changing actions that make use of the current state of the KB and the current context. On the other hand, it affects the set of TBox assertions that are relevant at each time point, and that have to be considered when processing queries posed over the KAB. Here we extend to our enriched setting the results on verification of rich temporal properties expressed in \( \mu \)-calculus, which had been established for standard KABs. Specifically, we show that under a run-boundedness condition, verification stays decidable and does not incur in any additional cost in terms of worst-case complexity. We also show how to adapt syntactic conditions ensuring run-boundedness so as to account for contextual information, taking into account context-dependent activation of TBox assertions.

1 Introduction

Recent work in the areas of knowledge representation, databases, and business processes [15,26,4,10,19] has identified the need for integrating static and dynamic aspects in the design and maintenance of complex information systems. The static aspects are characterized on the one hand by the data manipulated by the system, and on the other hand by possibly complex domain knowledge that may vary during the evolution of the system. Instead, dynamic aspects are affected by the processes that operate over the system, by executing actions that manipulate the state of the system. In such a setting, in which new data may be imported into the system from the outside environment, the system becomes infinite-state in general, and the verification of temporal properties becomes more challenging: indeed, neither finite-state model checking [14] nor most of the current techniques for infinite-state model checking apply to this case.

Knowledge and action bases (KABs) [4] have been introduced recently as a mechanism for capturing systems in which knowledge, data, and processes are combined and treated as first-class citizens. In particular, KABs provide a mechanism to represent semantically rich information in terms of a description logic (DL) knowledge base (KB) and a set of actions that manipulate such a KB over time. Additionally, actions
allow one to import into the system fresh values from the outside, via service calls. In this setting, the problem of verification of rich temporal properties expressed over KABs in a first-order variant of the \( \mu \)-calculus has been studied. Decidability has been established under the assumptions that in the properties first-order quantification across states is restricted, and that the system satisfies a so-called run-boundedness condition. Intuitively, these ensure that along each run the system cannot encounter (and hence manipulate) an unbounded number of distinct objects. In KABs, the intensional knowledge about the domain, expressed in terms of a DL TBox, is assumed to be fixed along the evolution of the system, i.e., independent of the actual state. However, this assumption is in general too restrictive, since specific knowledge might hold or be applicable on specific, context-dependent circumstances. Ideally, one should be able to form statements that are known to be true in certain cases, but not necessarily in all.

Work on representing and formally reasoning over contexts dates back to work on generality in AI see \[20\]. Since then, there has been some effort in knowledge representation and in DLs to devise context-sensitive formalisms, ranging from multi-context systems \[5\] to many-dimensional logics \[18\]. An important aspect in modeling context is related to the choice of which kind of information is considered to be fixed and which context dependent. Specifically, for DLs, one can define the assertions in the TBox \[2,13\], the concepts \[5\], or both \[24,18\] as context-dependent. Each choice addresses different needs, and results in differences in the complexity of reasoning.

We follow here the approach of \[2,13\], and introduce contextualized TBoxes, in which each inclusion assertion is adorned with context information that determines under which circumstances the inclusion assertion is considered to hold. The relation among contexts is described by means of a lattice in \[2\] and by means of a directed acyclic graph in \[13\]. In our case, we represent context using a finite set of context dimensions, each characterized by a finite set of domain values that are organized in a tree structure. If for a context dimension \( d \), a value \( v_2 \) is placed below \( v_1 \) in the tree (i.e., \( v_2 \) is a descendant of \( v_1 \)), then the context associated to \( v_1 \) is considered to be more general than the one for \( v_2 \), and hence whenever context dimension \( d \) is in value \( v_2 \), it is also in value \( v_1 \).

Starting from this representation of contexts, we enrich KABs towards context-sensitive KABs (CKABs), by representing the intensional information about the domain using a contextualized TBox, in place of an ordinary one. Moreover, the action component of KABs, which specifies how the states of the system evolve, is extended in CKABs with context changing actions. Such actions determine values for context dimensions in the new state, based on the data and the context in the current state. In addition, also regular state-changing actions can query, besides the state, also the context, and hence be enabled or disabled according to the context. Notably, we show that verification of a very rich temporal logic, which can be used to query the system evolution, contexts, and data, is decidable for run-bounded CKABs. We also discuss how to recast the syntactic condition of weak acyclicity \[4\], which ensures run-boundedness, to the case of CKABs.