Artificial Intelligence 234 (2016) 78–119

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Games for query inseparability of description logic knowledge bases ☆

Elena Botoeva [a], Roman Kontchakov [b,*], Vladislav Ryzhikov [a], Frank Wolter [c], Michael Zakharyaschev [b]

[a] KRDB Research Centre, Free University of Bozen–Bolzano, Italy
[b] Department of Computer Science and Information Systems, Birkbeck, University of London, UK
[c] Department of Computer Science, University of Liverpool, UK

ARTICLE INFO

Article history:
Received 17 May 2015
Received in revised form 7 November 2015
Accepted 15 January 2016
Available online 22 January 2016

Keywords:
Description logic
Knowledge base
Conjunctive query
Query inseparability
Games on graphs
Computational complexity

ABSTRACT

We consider conjunctive query inseparability of description logic knowledge bases with respect to a given signature—a fundamental problem in knowledge base versioning, module extraction, forgetting and knowledge exchange. We give a uniform game-theoretic characterisation of knowledge base conjunctive query inseparability and develop worst-case optimal decision algorithms for fragments of *Horn*-$\mathcal{ALCHI}$, including the description logics underpinning *OWL 2 QL* and *OWL 2 EL*. We also determine the data and combined complexity of deciding query inseparability. While query inseparability for all of these logics is P-complete for data complexity, the combined complexity ranges from P- to ExpTime- to 2ExpTime-completeness. We use these results to resolve two major open problems for *OWL 2 QL* by showing that TBox query inseparability and the membership problem for universal conjunctive query solutions in knowledge exchange are both ExpTime-complete for combined complexity. Finally, we introduce a more flexible notion of inseparability which compares answers to conjunctive queries in a given signature over a given set of individuals. In this case, checking query inseparability becomes NP-complete for data complexity, but the ExpTime- and 2ExpTime-completeness combined complexity results are preserved.

1. Introduction

A description logic (DL) knowledge base (KB) consists of a terminological box (TBox) and an assertion box (ABox). The TBox represents conceptual knowledge by providing a vocabulary for a domain of interest together with axioms that describe semantic relationships between the vocabulary items. To illustrate, consider the following toy TBox $\mathcal{T}_a$, which defines a vocabulary for the automotive industry:

$$Minivan \sqsubseteq Automobile,$$

☆ This paper is an invited revision of a paper which first appeared at the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014).

* Corresponding author.
E-mail addresses: botoeva@inf.unibz.it (E. Botoeva), roman@dcs.bbk.ac.uk (R. Kontchakov), ryzhikov@inf.unibz.it (V. Ryzhikov), wolter@liverpool.ac.uk (F. Wolter), michael@dcs.bbk.ac.uk (M. Zakharyaschev).

http://dx.doi.org/10.1016/j.artint.2016.01.010

$$Hybrid \sqsubseteq Automobile,$$
$$Automobile \sqsubseteq \exists poweredBy.Engine,$$
$$Hybrid \sqsubseteq \exists poweredBy.ElectricEngine \sqcap \exists poweredBy.InternalCombustionEngine,$$
$$ElectricEngine \sqsubseteq Engine,$$
$$InternalCombustionEngine \sqsubseteq Engine.$$

The first two axioms say that minivans and hybrids are automobiles, the third one claims that every automobile is powered by an engine, and the fourth axiom states that every hybrid is powered by an electric engine and also by an internal combustion engine. Thus, the TBox introduces, among others, the concept names (sets) *Minivan*, *Automobile* and *Engine*, states that the concept *Minivan* is subsumed by the concept *Automobile* and uses the role name (binary relation) *poweredBy* to say that automobiles are powered by engines. TBoxes, often called ontologies, are represented in many applications using the syntax of the Web Ontology Language OWL 2 (www.w3.org/TR/owl2-overview).

The ABox of a knowledge base is a set of facts storing data about the concept and role names introduced in the TBox. As an example ABox in the automotive domain, we will use the following set of assertions:

$$\mathcal{A}_a = \{\,Hybrid(toyota\_highlander),\ Minivan(toyota\_highlander),$$
$$Minivan(nissan\_note),\ poweredBy(nissan\_note, hr15de),\ InternalCombustionEngine(hr15de)\,\}.$$

Typical applications of KBs in modern information systems use the semantics of the concepts and roles in the TBox to enable the user to query the data in the ABox. This is particularly useful if the data is incomplete or comes from heterogeneous data sources, which is the case, for example, in linked data applications [1] and large-scale data integration projects [2,3], or if the data comprises the web content gathered by search engines using semantic markup [4].

As the data may be incomplete, the open world assumption is adopted when querying a KB $\mathcal{K}$: a tuple $\boldsymbol{a}$ of individuals from $\mathcal{K}$ is a (certain) answer to a query $\boldsymbol{q}$ over $\mathcal{K}$ if $\boldsymbol{q}(\boldsymbol{a})$ is true in every model of $\mathcal{K}$. Since general first-order queries are undecidable under the open-world semantics, the basic and most important querying instrument is conjunctive queries (CQs), which are ubiquitous in relational database systems and form the core of the Semantic Web query language SPARQL (www.w3.org/TR/sparql11-query). In our context, a CQ $\boldsymbol{q}(\boldsymbol{x})$ is a first-order formula $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$ such that $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a conjunction of atoms of the form $A(z_1)$ or $P(z_1, z_2)$, for a concept name A, a role name P, and variables z_1, z_2 from $\boldsymbol{x}, \boldsymbol{y}$.[1] For example, to find minivans powered by electric engines, one can use the CQ

$$\boldsymbol{q}(x) = \exists y\, \big(Minivan(x) \land poweredBy(x, y) \land ElectricEngine(y)\big),$$

with *toyota_highlander* being the only certain answer to $\boldsymbol{q}(x)$ over $(\mathcal{T}_a, \mathcal{A}_a)$.

The problem of answering CQs over KBs has been the focus of significant research in the DL community: deep complexity results have been obtained for a broad range of DLs (see below), new DLs have been introduced with tractable (in data complexity) query answering [5,6], a variety of query answering techniques have been invented [6,7] and implemented in a number of powerful software systems (see, e.g., [8] and references therein).

Apart from developing query answering techniques, a major research problem is KB engineering and maintenance. In fact, with typically large data and often complex and tangled ontologies, tool support for transforming and comparing KBs is becoming indispensable for applications. To begin with, KBs are never static entities. Like most software artefacts, they are updated to incorporate new information, and distinct versions are introduced for different applications. Thus, developing support for KB versioning has become an important research problem [9,10]. As dealing with a large and semantically tangled KB can be costly, one may want to extract from it a smaller module that is indistinguishable from the whole KB as far as the given application is concerned [11]. Another technique for extracting relevant information is forgetting, where the task is to replace a given KB with a new one, which uses only those concept and role names that are needed by the application but still provides the same information about those names as the original KB [12,13]. Finally, the vocabulary of a given KB may not be convenient for a new application. In this case, similarly to data exchange in databases [14]—where data structured under a source schema is converted to data under a target schema—one may want to transform a KB in a source signature to a KB given in a more useful target signature and representing the original KB in an accurate way. This task is known as knowledge exchange [15,16].

In this article, we investigate a relationship between KBs that is fundamental for all such tasks if querying the data via CQs is the main application. Let Σ be a relational signature consisting of a finite set of concept and role names. We say that KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are Σ-*query inseparable* and write $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_2$ if any CQ formulated in Σ has the same answers over $\mathcal{K}_1$ and $\mathcal{K}_2$. Note that even for Σ containing all concept and role names in the KBs, Σ-query inseparability does not necessarily imply logical equivalence: for example, $(\emptyset, \{A(a)\})$ is $\{A, B\}$-query inseparable from $(\{B \sqsubseteq A\}, \{A(a)\})$ but the two KBs are clearly not logically equivalent. Thus, if KBs are used for purposes other than querying data via CQs, then different notions

[1] Since we consider Horn DLs, the results of this article actually apply to unions of CQs (known as UCQs), see Remark 2 below. For simplicity, however, we consider CQs only.