Ontology Based Query Answering with Existential Rules
Michaël Thomazo

To cite this version:

HAL Id: lirmm-00925730
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00925730
Submitted on 8 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Ontology Based Query Answering with Existential Rules

Michaël Thomazo
University of Montpellier
France
thomazo@lirmm.fr

1 Framework and objectives

Ontology-Based Query Answering (OBQA) is currently a problem that receives a lot of attention both from knowledge representation and databases communities. The aim is to answer queries that are at least as expressive as conjunctive queries while taking an ontology into account. This is important in order to improve the quality of query answering and interoperability between different sources of data.

The mainstream formalism to deal with ontologies is description logics (DLs) ([Baader et al., 2003]) and DL-Lite ([Calvanese et al., 2007]). Real-world ontologies expressed in these DLs already exists, such as the medical ontology SNOMED-CT (based on EL [Baader, et al., 2009]). Slight adaptation of this algorithm makes it also optimal for UCQs. This class is a subset of b.t.s. which covers most of the known recognizable b.t.s. classes of rules. Slight adaptation of this algorithm makes it also optimal for these subclasses. This work has been published in [Baget et al., 2011a; Thomazo et al., 2012; Thomazo, 2012].

Pure query rewriting approaches suffer from the exponential blow-up of the size of rewritings w.r.t. to the query, even with solely class or role hierarchies. UCQs are then too large to be efficiently dealt with by RDMS. In [König et al., 2012], we show that this is inherent to UCQs by characterizing the smallest rewriting using that shape of formulas. I thus propose, in a paper submitted to IJCAI’13, to use semi-conjunctive queries, which are a more general form of positive existential formula. I present an algorithm for computing such rewritings, and experimentally evaluate their quality by checking the efficiency of evaluation of such queries. First results show that this approach is more efficient than using UCQs.

1 Constraints and equality rules are also needed, but this is out of the scope of this summary.

2See [Baget et al., 2011b] for formal definition of these classes.
3 Related work

I now briefly present some related results.

Recognizability of materialization-based approaches relies on two main criteria: guardedness and acyclicity. Guarded rules (where an atom of the body contains all variables of the body) have been generalized in several ways [Cali et al., 2009; Baget et al., 2011b; Krötzsch and Rudolph, 2011].

Weak-acyclicity [Fagin et al., 2005] has been generalized into super-weak acyclicity [Marnette, 2009] and join acyclicity [Krötzsch and Rudolph, 2011]. An incomparable notion relies on the notion of rule dependency [Baget et al., 2011b; Grau et al., 2012] proposes a semantic condition of acyclicity that generalizes all these notions.

Last, the combined approach mixes materialization and query rewriting: it both extends the data by applying rules (independently of the query) and rewrite the queries with respect to the rules (independently of the data). This has proven to be useful for both DL-Lite and EL ontologies.

Query rewriting approaches are applicable in particular to linear [Cali et al., 2009; Baget et al., 2011b] and (join-)sticky [Gottlob et al., 2011] rules, which are f.u.s.. Several algorithms have been implemented for linear rules or DL-Lite ontologies, rewriting either into a union of conjunctive queries (QuOnto, Requiem [Pérez-Urbina et al., 2009], NyayaGottlob et al., 2011], Iqaros [Venetis et al., 2012], Rapid [Chortaras et al., 2011]) or into a Datalog program (Presto [Rosati and Almatelli, 2010]). [König et al., 2012] proposes an optimal algorithm for any f.u.s.. The question of finding polynomial cases has also been addressed [Kikot et al., 2011].

References


