A Study on the Correspondence between FCA and ELI Ontologies

Melisachew Wudage Chekol, Amedeo Napoli

A Study on the Correspondence between FCA and $\mathcal{ELI}$ Ontologies

Melisachew Wudage Chekol and Amedeo Napoli

LORIA (INRIA, CNRS, and Université de Lorraine), France
{melisachew.chekol,amedeo.napoli}@inria.fr

Abstract. The description logic $\mathcal{EL}$ has been used to support ontology design in various domains, and especially in biology and medecine. $\mathcal{EL}$ is known for its efficient reasoning and query answering capabilities. By contrast, ontology design and query answering can be supported and guided within an FCA framework. Accordingly, in this paper, we propose a formal transformation of $\mathcal{ELI}$ (an extension of $\mathcal{EL}$ with *inverse roles*) ontologies into an FCA framework, i.e. $K_{\mathcal{ELI}}$, and we provide a formal characterization of this transformation. Then we show that SPARQL query answering over $\mathcal{ELI}$ ontologies can be reduced to lattice query answering over $K_{\mathcal{ELI}}$ concept lattices. This simplifies the query answering task and shows that some basic semantic web tasks can be improved when considered from an FCA perspective.

1 Introduction

Knowledge discovery in data represented by means of objects and their attributes can be done using formal concept analysis (FCA) [5]. Concept lattices can reveal hidden relations within data and can be used for organizing and classifying data. A survey of the benefits of applying FCA to Semantic Web (SW) and vice versa has been proposed in [8]. As mentioned in that paper, a few of these benefits ranges from knowledge discovery, ontology completion, to computing subsumption hierarchy of least common subsumers. Additionally, studies in [3] and [7] are based on FCA for managing SW data while finite models of description logics (as $\mathcal{EL}$) are explored in [1]. All these studies propose methods for analysing SW data within FCA. Nevertheless, none of them offers a practical way of representing SW data within a formal context which is the basic data structure for FCA. We deem it necessary to provide a mathematically founded method to formalize the representation and the analysis of SW data based on FCA.

In this work, we focus on $\mathcal{ELI}$ (an extension of $\mathcal{EL}$ with inverse roles) ontologies. $\mathcal{EL}$ is one of OWL 2 profiles (OWL 2 $\mathcal{EL}$) which is mainly used for designing large biomedical ontologies such as SNOMED-CT[1] and the NCI thesaurus[2]. These two ontologies have large concept hierarchies that can be queried

[1] http://www.ihtsdo.org/snomed-ct/
[2] http://ncit.nci.nih.gov/

with SPARQL. However, including inferred data in query answering requires either a reasoner to infer all implicit information or query rewriting using property paths (that enable navigation in a hierarchy) [6]. The latter obliges the user to know the nuts and bolts of SPARQL. To overcome these difficulties, we reduce SPARQL query answering in $\mathcal{ELI}$ ontologies into query answering in concept lattices along with the transformation of the queried ontology into a formal context. Then, the resulting concept lattice provides support for query answering (but this does not replace SPARQL) and also for visualization and navigation of relations within SW data.

Overall, we work towards (i) a formal characterization of the transformation of ontologies into a formal context, (ii) translating the difficulty of SPARQL query answering over ontologies into query answering over concept lattices, and finally (iii) providing organization of SPARQL query answers with concept lattices.

2 Transforming $\mathcal{ELI}$ Ontologies into Formal Contexts

A formal context represents data using objects and their attributes. Formally, it is a triple $K = (G, M, I)$ where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$ is a binary relation. A derivation operator ($'$) is used to compute *formal concepts* of a context. Given a set of objects A, a derivation operator $'$ computes the maximal set of attributes shared by objects in A and is denoted by A' (this is done dually with set of attributes B). A formal concept is a pair (A, B) where $A' = B$ and $B' = A$. A set of formal concepts ordered with the set inclusion relation form a *concept lattice* [5].

One difficulty of transforming DL ontologies into formal contexts is mainly due to the fact that while DL languages are based on the *open world assumption (OWA)*, FCA relies on the *closed world assumption (CWA)*. The former permits to specify only known data whereas the later demands that all data should be explicitly specified. To slightly close the gap between these two worlds, we provide a transformation that maintains a DL semantics into an FCA setting.

To transform an $\mathcal{ELI}$ ontology $\mathrm{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ into a formal context $K = (G, M, I)$, the schema axioms in the TBox become background implications [4]. Then, individuals in the ABox correspond to objects in G, class names in the ABox and TBox yield attributes in M, and ABox assertions create relations between objects and attributes $I \subseteq G \times M$. Here, we consider acyclic TBoxes to avoid that class names become objects in a context. The following table gives a summary of the correspondence.

$\mathcal{ELI}$ $\mathrm{O} = \langle \mathcal{T}, \mathcal{A} \rangle$	FCA formal context $K_{\mathcal{O}} = (G, M, I)$ + background implications $\mathcal{L}$
$\mathcal{T} = \{C \sqsubseteq D\}$	$\{C \rightarrow D\} \in \mathcal{L}$
$\mathcal{A} = \{C(a),$	$a \in G$, $C \in M$, and $(a, C) \in I$
$R(a, b)\}$	$a, b \in G$, $\exists R.\top, \exists R^{-}.\top \in M$, $(a, \exists R.\top) \in I$, and $(b, \exists R^{-}.\top) \in I$

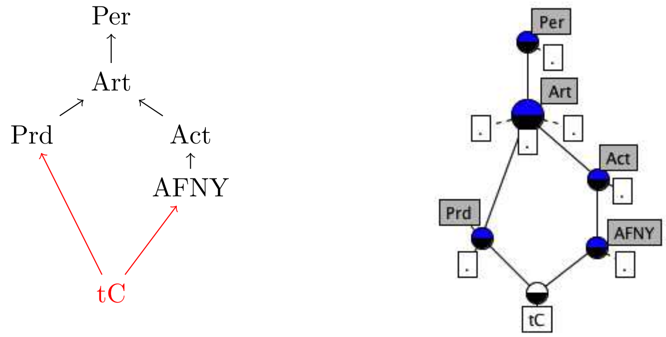

Fig. 1: The ontology in Example 1 and its associated concept lattice.

Example 1. Consider the transformation of the following $\mathcal{ELI}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ into a formal context $K_{\mathcal{O}}$ and its background implications $\mathcal{L}$.

$$\mathcal{T} = \{\text{ActorsFromNewYork (AFNY)} \sqsubseteq \text{Actor (Act)},$$
$$\text{FilmProducer (Prd)} \sqsubseteq \text{Artist (Art)}, \text{Actor} \sqsubseteq \text{Artist}, \text{Artist} \sqsubseteq \text{Person (Per)}\}$$
$$\mathcal{A} = \{\text{tomCruise (tC)} \in \text{ActorsFromNewYork}\}$$

$K_{\mathcal{O}}$	AFNY	Prd	Act	Art	Per
tC	x				

$$\mathcal{L} = \{\ \text{AFNY} \rightarrow \text{Act},\ \text{Prd} \rightarrow \text{Art},$$
$$\text{Act} \rightarrow \text{Art},\ \text{Art} \rightarrow \text{Per}\ \}$$

Construction of a concept lattice: In [4], an algorithm to construct a concept lattice from a formal context w.r.t. background implications is provided. This technique is employed here and the concept lattice associated with the formal context and background implications of Example 1 is depicted in Figure 1.

This is a simple example but SNOMED-CT and NCIT are much larger than that but not more complex. Let us how a concept lattice based on an $\mathcal{ELI}$ ontology can be queried.

3 SPARQL query answering over ontologies vs query answering over concept lattices

SPARQL query answering over $\mathcal{ELI}$ ontologies can be considered from the point of view of query answering over $K_{\mathcal{ELI}}$ concept lattices. Querying concept lattices amounts to fetching the objects given a set of attributes as query constants and to fetch the attributes given a set of objects as query constants or terms [2]. Query terms can be connected using the logical operators: *and*, *union*, and *set difference* to form a complex term.

SPARQL query answering over ontologies can be done in two main ways: (1) *Materialization* amounts to computing all implicit data before evaluating the query. This can be done by using a DL reasoner. (2) *Query rewriting* amounts to converting a query into another one using property paths and schema axioms.

Query answering over $\mathcal{ELI}$ ontologies with SPARQL appears to be harder than query answering over $K_{\mathcal{ELI}}$ concept lattices. SPARQL requires expensive tasks such as materialization and query rewriting but its expressive power is better than lattice querying. By contrast, lattice querying is practically sufficient to retrieve instances for $\mathcal{ELI}$ ontologies as shown by the following example.

Example 2. Let us consider the evaluation of the SPARQL query Q on the ontology $\mathcal{O}$ (in Figure 1) and its materialization $\mathcal{O}'$. Q = select all objects, elements who are artists = `SELECT ?x WHERE` $\{$`?x a Artist .`$\}$. Under simple entailment evaluation of a SPARQL query, the answer of Q over $\mathcal{O}$ is empty. To get non-empty answers for Q, one can evaluate Q over the materialization of $\mathcal{O}$ that we call $\mathcal{O}'$, where $Q(\mathcal{O}') = \{\text{tomCruise}\}$. Another way is to rewrite Q into Q' = `SELECT ?x WHERE` $\{$`?x a/rdfs:subClassOf`* `Artist .`$\}$. Q' selects all instances of Artist and that of its subclasses by navigating through the subclass relation. Then, the evaluation of $Q'(\mathcal{O})$ returns $\{\text{tomCruise}\}$.

By contrast, Q is converted into a lattice query as $q(x) = (x, Artist)$. The evaluation of this query over a concept lattice $K_{\mathcal{O}}$ obtained from $\mathcal{O}$ (Figure 1) is $Q'(K_{\mathcal{O}}) = \{\text{tomCruise}\}$, as it is sufficient to return all objects which are instances of Artist or any of its subconcept.

4 Discussion

It can be convenient to use FCA as a guideline for designing and querying $\mathcal{ELI}$ ontologies. In addition, FCA provides visualization and navigation capabilities. The present work does not apply to all ontologies but seems to be well suited to $\mathcal{ELI}$ ontologies. We plan to extend and experiment the proposed approach, especially with real-world and large datasets.

References

1. Baader, F., Distel, F.: Exploring finite models in the description logic EL gfp. In: ICFCA. pp. 146–161. Springer (2009)
2. Carpineto, C., Romano, G.: Concept data analysis: Theory and applications. Wiley (2004)
3. d'Aquin, M., Motta, E.: Extracting relevant questions to an RDF dataset using formal concept analysis. In: Proceedings of the sixth international conference on Knowledge capture. pp. 121–128. ACM (2011)
4. Ganter, B.: Attribute exploration with background knowledge. Theoretical Computer Science 217(2), 215 – 233 (1999)
5. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
6. Glimm, B.: Using SPARQL with RDFS and OWL entailment. Reasoning Web. Semantic Technologies for the Web of Data pp. 137–201 (2011)
7. Kirchberg, M., Leonardi, E., Tan, Y.S., Link, S., Ko, R.K., Lee, B.S.: Formal concept discovery in semantic web data. In: ICFCA. pp. 164–179. Springer-Verlag (2012)
8. Sertkaya, B.: A survey on how description logic ontologies benefit from FCA. In: CLA. vol. 672, pp. 2–21 (2010)