Decentralized Reasoning on a Network of Aligned Ontologies with Link Keys

Jérémy Lhez[1], Chan Le Duc[1], Thinh Dong[2], and Myriam Lamolle[1]

[1] LIASD, Université Paris 8 - IUT de Montreuil, France
{lhez,leduc,lamolle}@iut.univ-paris8.fr
[2] University of Danang, Vietnam
dnnthinh@kontum.udn.vn

1 Introduction

Reasoning on a network of aligned ontologies has been investigated in different contexts where the semantics given to correspondences differs from one to another. In this paper, we introduce a new semantics of correspondences which is weaker than the usual one and propose a procedure for reasoning over a network of aligned ontologies with link keys [1] in a decentralized manner, i.e. reasoning can be independently performed on different sites This process allows to reduce polynomially global reasoning to local reasoning.

To achieve such results for a network of ontologies expressed in the description logic $\mathcal{ALC}$, the semantics of a correspondence, denoted $C \rightarrow D$ where C and D are concepts in ontologies O_i and O_j respectively, is defined as an implication of concept unsatisfiabilities (i.e. unsatisfiability of D implies unsatisfiability of C) rather than a concept subsumption as usual. This weakened semantics allows to reduce the reasoning complexity over a network of aligned ontologies since (i) only individual equalities and concept unsatisfiabilities such as $a \approx b$, $C \sqsubseteq \bot$ can be propagated from one to another ontology, and (ii) if a concept is locally unsatifiable in an ontology then it remains unsatisfiable when adding to the ontology individual equalites or concept unsatisfiabilities. The weakened semantics would be relevant for correspondences between ontologies of different nature. Given two ontologies about `equipment` and `staff` and a correspondence $\mathsf{Computer} \rightarrow \mathsf{Developer}$ between them. With this correspondence, the weakened semantics tells us that if there is no developer then there is no computer. The standard semantics is irrelevant in this case.

We use $\langle \{O_i\}_{i=1}^n, \{A_{ij}\}_{i=1,j=2,i<j}^n \rangle$ to denote a network of ontologies where each O_i is an ontology expressed in $\mathcal{ALC}$ and each A_{ij} contains individual correspondences, link keys with the usual semantics, or concept correspondences with the weakened semantics. Such a network is *consistent* if there is a model $\mathcal{I}_i$ of each ontology O_i which satisfies all correspondences in each A_{ij}. We will present our algorithms for a network composed of two ontologies O_1, O_2 and an alignment A_{12}. These algorithms can be straightforwardly extended to a general network of aligned ontologies.

2 Decentralized reasoning by propagating

As discussed in Section 1, the weakened semantics of alignments allows us to decompose checking consistency of a network into two steps which consist in propagating knowledge from one to another ontology.

Propagating individual equalities. This step discovers all inferred individual equalities by applying link keys in the alignment and using local reasoners for entailment on ontologies. For example, a new individual correspondence $a \approx b$ will be discovered and added to A_{12} if $\{\langle P, Q\rangle\}\mathsf{linkkey}\langle C, D\rangle \in A_{12}$, $c \approx d \in A_{12}$ and $P(a, c) \in O_1$, $Q(b, d) \in O_2$. A new equality $a \approx c$ will be discovered and added to O_1 if $a \approx b, c \approx b \in A_{12}$. When a local reasoner is called to check whether $O_i \models a \approx b$, it needs only one O_i for reasoning.

Propagating concept unsatisfiabilities. This step uses local reasoners associated with ontologies to discover from each ontology O_i new unsatisfiable concepts which can result from unsatisfiable concepts in O_j via concept correspondences. For instance, if $O_1 \models C \sqsubseteq \bot$ and $C \leftarrow D \in A_{12}$ then a new axiom $D \sqsubseteq \bot$ will be added to O_2. As the previous step, each local reasoner needs only one O_i for reasoning.

The main algorithm executes these two steps until either an inconsistency is found, or a stationary state is reached. If O_1, O_2 are consistent, and A_{12} does not contain any pair $a \approx b, a \not\approx b$, then the network itself is consistent. Our algorithm runs in polynomial time in the size of the network since the propagation procedures add only axioms and assertions which are composed of (sub-)concepts and named individuals occurring in the ontologies and alignments. Moreover, these algorithms never remove anything from the network.

Implementation and tests. The algorithms has been implemented and integrated within DRAOn [2]. HermiT [3] is used for local reasoners. We performed some tests with several datasets available from the OAEI web site. We compared the performances of DRAOn under the IDDL semantics [2] and the weakened semantics. Better performance has been observed for the latter. For instance, checking consistency of the network composed of SNOMED, FMA and the alignment took 81 seconds under the weakened semantics while it took greater than 15 minutes under the IDDL semantics. We also added to the alignments some link keys, and ran other tests to validate the implementation of our algorithm.

References

1. Gmati, M., Atencia, M., Euzenat, J.: Tableau extensions for reasoning with link keys. In: Proceedings of the 11th International Workshop on Ontology Matching. (2016) 37–48
2. Le Duc, C., Lamolle, M., Zimmermann, A., Curé, O.: Draon: A distributed reasoner for aligned ontologies. In: Informal Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE). (2013) 81–86
3. Shearer, R., Motik, B., Horrocks, I.: HermiT: A Highly-Efficient OWL Reasoner. In: Proc. of the 5th Int. Workshop on OWL (OWLED). (2008)