Understanding Justifications for Entailments in OWL

Matthew Horridge
The University of Manchester
horridge@cs.man.ac.uk

Bijan Parsia
The University of Manchester
parsia@cs.man.ac.uk

Ulrike Sattler
The University of Manchester
sattler@cs.man.ac.uk

ABSTRACT

Recent work in explanation of entailments in ontologies has focused on justifications and their variants. While in many cases, just presenting the justification is sufficient for user understanding, and in all cases justifications are much better than nothing, we have empirically identified cases where understanding *how* a justification supports an entailment is inordinately difficult. Indeed there are naturally occurring justifications that people, with varying expertise in OWL, cannot understand. To address this problem, we have developed a novel conceptual framework for *justification oriented proofs*. Given a justification for an entailment in an ontology, intermediate inference steps, called lemmas, are automatically derived, that bridge the gap between the axioms in the justification and the entailment. The proof shows in a stepwise way how the lemmas and ultimately the entailment follow from the justification. At the heart of the framework is the notion of a "complexity model", which predicts how easy or difficult it is for a user to understand a justification, and is used for selecting the lemmas to insert into a proof. This poster and demo presents this framework backed by a prototype implementation.

1. INTRODUCTION

Over the past few years there has been a significant amount of interest in the area of explaining entailments in OWL ontologies. Without some kind of tool support, it can be very difficult, or even impossible, to work out why entailments arise in ontologies. Even in small ontologies, that only contain tens of axioms, there can be multiple reasons for an entailment, none of which may be obvious. It is for this reason that there has recently been a lot of focus on generating explanations for entailments in ontologies. In the OWL world, *justifications* are a popular form of explanation for entailments. Justifications are minimal subsets of an ontology that are sufficient for an entailment to hold [2, 3]. Virtually all mainstream ontology editors such as Protégé-4, Swoop, Top Braid Composer, and the NeOn Toolkit provide support for generating justifications as explanations for arbitrary entailments. Justifications have proved enormously useful for understanding and debugging ontologies. In [2], Kalyanpur presents a user study which showed that the availability of justifications had a significant positive impact on the ability of users to successfully diagnose and repair an ontology.

Despite the utility of justifications, and the fact that the availability of a justification, as a form of an explanation, is nearly always better that nothing, in certain cases people can struggle to understand how a justification supports an entailment. Indeed, as part of this work, a user study was carried out, which showed that people with varying levels of expertise in OWL can find justifications difficult or even impossible to understand [1].

Figure 1: A justification for $\mathsf{Person} \sqsubseteq \bot$

$$
\begin{array}{l}
\mathsf{Person} \sqsubseteq \neg\mathsf{Movie} \\
\mathsf{RRated} \sqsubseteq \mathsf{CatMovie} \\
\mathsf{CatMovie} \sqsubseteq \mathsf{Movie} \\
\mathsf{RRated} \equiv (\exists \mathsf{hasScript.ThrillerScript}) \\
\qquad \sqcup (\forall \mathsf{hasViolenceLevel.High}) \\
\mathsf{Domain}(\mathsf{hasViolenceLevel},\ \mathsf{Movie})
\end{array}
$$

The work presented in this poster focuses on solving this problem using *Justification Oriented Proofs*. Given a justification, the framework presented in this paper transforms it into a *Justification Oriented Proof*. Intermediate inference steps, called lemmas, are automatically derived to bridge the gap between the axioms in the justification and the entailment. A justification oriented proof shows in a stepwise way how the lemmas, and ultimately the entailment, follow from the justification. At the heart of the framework is the notion of a "complexity model", which predicts how easy or difficult it is for a user to understand a justification, and is used for selecting the lemmas to insert into a proof.

2. BACKGROUND AND MOTIVATION

In order to give a flavour of why it can be difficult for people to understand justifications, two example justifications are presented in Figures 1 and 2. The reader is encouraged to work through and understand these justifications in order to appreciate the issues.

Consider the justification shown in Figure 1, which is for the entailment $\mathsf{Person} \sqsubseteq \bot$ (i.e. Person is unsatisfiable). This is a justification for an entailment from a real ontology about movies that was posted to the Protégé mailing list. The person who posted it could not understand why Person was unsatisfiable, even when presented with the justification. When this justification was shown to a range of people, some of them having worked with OWL for several years, many of them struggled to understand it. In fact, some of the people claimed that it was not a justification for the entailment.

Figure 2: A justification for Newspaper(DailyMirror)

InverseProperties(hasPet, isPetOf)
isPetOf(Rex, Mick)
Domain(hasPet, Person)
Male(Mick)
reads(Mick, DailyMirror)
drives(Mick, Q123ABC)
Van(Q123ABC)
$\mathsf{Van} \sqsubseteq \mathsf{Vehicle}$
WhiteThing(Q123ABC)
$\mathsf{Driver} \equiv \mathsf{Person} \sqcap \exists \mathsf{drives.Vehicle}$
$\mathsf{Driver} \sqsubseteq \mathsf{Adult}$
$\mathsf{Man} \equiv \mathsf{Adult} \sqcap \mathsf{Male} \sqcap \mathsf{Person}$
$\mathsf{WhiteVanMan} \equiv \mathsf{Man} \sqcap \exists \mathsf{drives.(Van} \sqcap \mathsf{WhiteThing)}$
$\mathsf{WhiteVanMan} \sqsubseteq \forall \mathsf{reads.Tabloid}$
$\mathsf{Tabloid} \sqsubseteq \mathsf{Newspaper}$

Figure 2 presents example justification for the entailment Newspaper(DailyMirror) (read as DailyMirror is an instance of Newspaper). When this justification was shown to people, many of them were put off by the number and variety of axioms in the justification. Those that weren't put off reported that they found it very difficult to work through.

In essence, while a justification gathers together the axioms, or *premises*, for an entailment, it is left up to the person reading the justification to figure out how these premises interplay with each other to give rise to the entailment in question. In some cases, for example those presented in Figures 1 and 2, the "gap" between seeing the premises and understanding how they give rise to the entailment is too large. So much so, that it is very difficult, or even impossible, for a person to understand the justification. It is arguable, that what is needed, is some kind of "proof", that explicates the various *steps* involved in understanding how a justification supports an entailment. Therefore, this work presents *Justification Oriented Proofs* as a possible solution.

3. JUSTIFICATION ORIENTED PROOFS

The main idea behind a justification oriented proof is depicted in Figure 3. The numbered rectangles represent axioms, with the rightmost rectangle, labelled η, representing the entailment of interest. The shaded rectangles labelled with "1" – "5" represent exactly the axioms that appear in the original justification $\mathcal{J}$ for the entailment (and are therefore in the ontology as asserted axioms). Hence $\mathcal{J} = \{1, 2, 3, 4, 5\}$ is a justification for η with respect to the ontology that entails η. The idea behind a justification oriented proof, is to augment a justification with helpful intermediate steps, called *lemmas*. This produces a weakly connected directed acyclic graph, with one sink node that represents the entailment of interest and a source node for each axiom in the justification. In the example shown in 3, axiom 6 is a lemma for axioms 1, 2 and 3 (conversely, axioms 1, 2 and 3 are a justification for axiom 6). Axiom 7 is a lemma for axioms 3, 4 and 5 (conversely axioms 3, 4 and 5 are a justification for axioms 7). Together axioms 6 and 7 constitute a justification for η i.e. the entailment. Notice that axiom 3 participates in different justifications for different lemmas. The main idea of this approach, is that compared to the justification $\mathcal{J}$, it is easier to see and understand why axioms 6 and 7 entail η, and it is also easy to see why axioms 1, 2 and 3 entail axiom 6, and why axioms 3, 4 and 5 entail axiom 7. More over, the ability to spot that $\mathcal{J}$ entails axioms 6 and 7, and to realise the part these axioms play, may be key to understanding how $\mathcal{J} \models \eta$, yet they may not be at all salient to a person looking at $\mathcal{J}$. In other words, the process of understanding why a justification supports an entailment, is transformed to understanding how simpler subsets of the justification result in intermediate entailments, and understanding how these intermediate entailments fit together to give rise the main entailment of interest.

Figure 3: A schematic of a Justification Oriented Proof

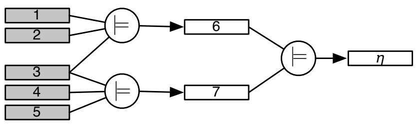

Figure 4: A schematic of a justification oriented proof for the justification shown in Figure 1

$\mathbf{Person} \sqsubseteq \neg\mathbf{Movie}$
$\top \sqsubseteq \mathsf{Movie}$
  $\forall \mathsf{hasViolenceLevel}.\bot \sqsubseteq \mathsf{Movie}$
    $\forall \mathsf{hasViolenceLevel}.\bot \sqsubseteq \mathsf{RRated}$
      $\mathbf{RRated} \equiv (\exists \mathbf{hasScript.ThrillerScript})$
        $\sqcup(\forall \mathbf{hasViolenceLevel.High})$
    $\mathsf{RRated} \sqsubseteq \mathsf{Movie}$
      $\mathbf{RRated} \sqsubseteq \mathbf{CatMovie}$
      $\mathbf{CatMovie} \sqsubseteq \mathbf{Movie}$
  $\exists \mathsf{hasViolenceLevel}.\top \sqsubseteq \mathsf{Movie}$
    Domain(hasViolenceLevel, Movie)

An example of a justification oriented proof is shown in Figure 4. Note that the presentation style used here is merely for illustrative purposes, it is designed to give a flavour of the kinds of lemmas that get introduced into a proof rather than as an end user presentation device. The axioms shown in bold are the axioms in the original justification, all other axioms correspond to lemmas. The axioms at each level of indentation form a justification for the axiom that is above them.

More example justification oriented proofs and a demonstration of the software for computing them will be available at the poster/demo session.

4. REFERENCES

[1] M. Horridge, B. Parsia, and U. Sattler. Lemmas for justifications in OWL. In *Description Logics (DL 2009)*, 2009.
[2] A. Kalyanpur. *Debugging and Repair of OWL Ontologies*. PhD thesis, The Graduate School of the University of Maryland, 2006.
[3] S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI International Joint Conference on Artificial Intelligence*, 2003.