

Towards Explicative Inference for OWL

Marvin R.G. Schiller and Birte Glimm

Institute of Artificial Intelligence, University of Ulm, Germany
{marvin.schiller,birte.glimm}@uni-ulm.de

Abstract. Automated reasoning in OWL is capable of inferences that are nontrivial for people to understand. We argue that the understanding of inferences would benefit from stepwise explanations. To build a system that supports such explicative inference, we propose a framework based on inference rules and proof tactics for OWL ontologies. In particular, the goal is to present inferences in a suitable and adaptable way to human users, and to predict whether certain inferences are harder to understand than others. This article outlines the conception of this framework and its benefits whose implementation is currently work in progress.

1 Introduction

Inferences in OWL vary in their degree of difficulty for people to understand. This was recently shown in experiments, where subjects who were either considered competent in OWL or who considered themselves competent in OWL were found to dismiss inferences in spite of them being logically valid (see [1,2]). One task where it is relevant to understand why certain entailments hold (when in fact, they are undesired or counterintuitive) is ontology debugging, as posited by [3].

Previous work has argued for reasoning services that present inference steps in a transparent manner, e.g. [4], to facilitate understanding. One problem that has been addressed is the identification of *justifications*, that is, to make explicit minimal sets of axioms within an ontology that are sufficient for a given entailment to hold. Presenting such a precise version of justification to the user clearly has a benefit, as demonstrated by [5]. Further work has argued for making intermediate reasoning steps explicit, such as [6] and [4]. McGuinness et al. [4] even consider the presentation of explanations at “different levels of granularity and focus”, where different segments of a proof for an entailment are rewritten to provide “an abstraction”. Further techniques that are used in the context of explanations for entailments are graphical representations of proofs, and natural language output, usually generated using patterns.

An effective explanation in the spirit of the above work presupposes a conception of what kinds of explanations are understandable to particular groups of users (or in general), and where the potential difficulties are. That is, the difficulties of (purported) OWL experts to understand some entailments but not others should inform the design of such a system, to enable support (and transparency) where needed. Consequently, such a system should be measured by the degree to which it makes a difficult task easier. However, the difficulties

that are inherent to the understanding of certain patterns of inference have not been investigated in sufficient depth. Horridge et al. [2] present a model for the “cognitive complexity” of entailments. It received some support in a preliminary experimental evaluation, but more work is clearly required to establish a theoretical underpinning and a more robust evaluation for such kinds of models. In a similar vein, in previous work we have investigated suitable levels of detail and conciseness of proofs in a mathematical assistant system employed in the context of (general mathematical) proof tutoring [7]. It remains to be shown in how far this approach can be usefully applied to OWL reasoning.

The objective of this article is to outline the design of a system currently under development to facilitate inferences in OWL that are “explicative”, in the spirit of the above stipulations. It is based on the assumption that the best way to generate explanations of entailments in OWL are rule-based inference systems. The hypothesis that the stepwise presentation of derivations is under circumstances more useful than the presentation of “justifications” alone (i.e. the axioms from which the entailment is derived) is yet to be tested, one of the objectives to be addressed with the completed system. Furthermore, we are interested not only in formal properties of inference rule sets (such as soundness, of course), but in their plausibility for the user, i.e. whether the user would find it natural or difficult to make a particular inference, or to understand a given inference. A second hypothesis to be addressed is whether the difficulty of an entailment in OWL can be predicted more precisely on the basis of derivations in a suitable rule system than by looking at properties of the axioms and the derived formula alone (the model in [2] makes such a restriction), or by using an off-the-shelf reasoner, for instance a tableau-based reasoner. A third objective is to enable inference rules that can be decomposed further into more fine-grained steps, to result in a hierarchical proof structure. Such rules that encode commonly used proof patterns are referred to as tactics in the theorem proving community. The idea to base proof presentations on hierarchical proofs has previously been implemented in some systems, see [8,9]. The main benefit is that the same proof can be presented at different levels of detail or conciseness, depending on the user’s choice [9], or some model of appropriate step size [7]. One objective of this work is to investigate tactics for OWL reasoning, and to establish the benefits of such an approach for explanation.

2 A Framework for Explicative Inference

The goal for the presented framework is to provide a suitable representation for derivations, as the basis for proof presentation, and for using these derivations as a model for estimating the complexity/effort of making the entailment represented by the derivation. As the formula language, the presented framework focuses on the constructs supported by OWL 2 [10] by directly making use of the OWL API [11].¹

¹ <http://owlapi.sourceforge.net>

$$\begin{array}{c}
\text{R17} \frac{\text{dom}(r0, X) \quad \forall r0.\perp \sqsubseteq X}{\top \sqsubseteq X} \\
\text{(a)}
\end{array}
\qquad
\begin{array}{c}
\text{R17(left)} \frac{\Gamma, \top \sqsubseteq X, \text{dom}(r0, X), \forall r0.\perp \sqsubseteq X \vdash \Delta}{\Gamma, \text{dom}(r0, X), \forall r0.\perp \sqsubseteq X \vdash \Delta} \\
\text{(b)}
\end{array}$$

Fig. 1. (a) Example inference from the empirical study of Nguyen et al. [1], and (b) corresponding sequent style inference rule in forward direction (i.e. manipulating the antecedent)

2.1 Foundations

As a representation formalism for explicative inference, we use sequent style proofs and sequent inference rules. This has the advantage that different sets of inference rules can be used. On the one hand, this includes a set of sequent rules for the \mathcal{ALC} fragment from [12], proposed specifically for the explanation of subsumptions in \mathcal{ALC} . On the other hand, we interpret the entailments investigated in an experiment by Nguyen et al. [1] as inference rules. As an example, consider one of the 51 inference problems considered by Nguyen et al., the following entailment (as worded in the experiment):

Everything that has a worship leader is a fomorian.
Everything that has no worship leader at all is a fomorian.
Everything is a fomorian.

The representation of the above entailment as an inference rule is shown in Fig. 1, where (a) is the interpretation of the entailment according to [1], and (b) the interpretation as a sequent style inference rule (here, we only consider the “left hand side” variant of the rule, i.e. forward direction of inference application). In Fig. 1 (b), Γ and Δ represent formulae in the antecedent and succedent that are not manipulated by the inference. As it is common, we construct sequent proofs from the endsequent (the sequent containing all premises and the conclusion(s) to be shown) “upwards” (in terms of notation), this explains the different configuration of rules (a) and (b). To avoid the contraction rule², the principal formulae of R17(left) are also part of the side formulae.

The construct $\text{dom}(r, X)$ is an abbreviation³ provided by *SHOIQ* (cf. e.g. [13]), and is defined as $\exists r.\top \sqsubseteq X$. This construct, instantiated with some imaginary concepts and roles, i.e. $\text{dom}(\text{hasWorshipLeader}, \text{Fomorian})$, was verbalized by Nguyen et al. as “Everything that has a worship leader is a fomorian.” in the natural language presentation of the entailment. Note that this is logically equivalent to $\text{dom}(\text{hasWorshipLeader}, \text{Fomorian})$ but not literally congruent.

The entailment quoted above was correctly judged as valid by 39 out of 50 participants in the experiment by Nguyen et al., i.e. 11 participants wrongly

² In fact, structural rules (which include the contraction rule) are often not considered useful for proof explanation, for example see [12].

³ We write $\text{dom}(r, X)$ instead of $\text{domain}(r, X)$ (the latter being the usual syntax, e.g. in [13]) to save space in figures.

judged the entailment not to hold. Using the Nguyen et al. rule set to justify entailments obviously results in different proofs from the ones using more concise rule sets such as those presented in [12]. The relative difficulties of the entailments/rules investigated by Nguyen et al. are reflected in their facility indices (the proportion of correct responses from experiment participants, interpreted as an estimate of the entailments’ difficulties of being understood by the participants). These can potentially be used to determine whether the employed inferences in a derivation are to be considered to be difficult or easy. How these difficulties combine (whether they are purely additive, or whether their specific combination matters, as suggested by [2, 7]) is not clear, however, and will be one of the research questions to be addressed in future work with the presented framework.

As an example for the representation of such a proof in our system, consider an entailment taken from [13]. Given the following (justification) axioms, it holds that $Person \sqsubseteq \perp$ (i.e. the class *Person* is unsatisfiable):⁴

$$\begin{aligned}
RRated &\equiv (\exists hasScript. ThrillerScript \sqcup \forall hasVL. High) \\
&\quad dom(hasVL, Movie) \\
Person &\sqsubseteq \neg(Movie) \\
RRated &\sqsubseteq CatMovie \\
CatMovie &\sqsubseteq Movie
\end{aligned}$$

Fig. 2 shows the corresponding proof when a rule set based on the entailments in [1] is used. Rules that are numbered (e.g. R1) directly correspond to the entailments listed in [1]. Rules that are named are provided additionally in our framework, and in the case of the example entailment, they are needed to suitably represent the derivation. A list of the inference rules that are used in the example is provided in the appendix.

Fig. 2 (a) shows the proof tree (for ease of presentation, we use the Gentzen-style tree, rather than reproducing the full sequents at each step, the latter being the actual representation used in our system). The proof presentation is generated via simple pretty printing, and results in a linear proof in Fig. 2 (b). The sequent representation would be a suitable starting point for natural language generation (beyond the simple pretty printing we currently use) to facilitate the verbalization of proofs, but this is outside the scope of this paper (and often it is not necessary, for example in the case of ontology debugging, where derivations consisting of concise formulae might be more useful for an ontology engineer to quickly understand the source of a problem than natural-language text). The graphical representation of proof trees also has its benefits (see, for example, [4]). So far, we have not connected our framework to a tool that visually lays out the proof trees, which might be useful in its own right.

⁴ We write “*hasVL*” as a shorthand for the role “*hasViolenceLevel*”.

$$\begin{array}{c}
\frac{RRated \equiv (\exists hasScript. ThrillerScript \sqcup \forall hasVL.High)}{(\exists hasScript. ThrillerScript \sqcup \forall hasVL.High) \sqsubseteq RRated} \text{R1} \\
\frac{CatMovie \sqsubseteq Movie \quad RRated \sqsubseteq CatMovie}{RRated \sqsubseteq Movie} \text{R12} \quad \frac{\forall hasVL.High \sqsubseteq RRated}{\forall hasVL.\perp \sqsubseteq RRated} \text{R4} \quad \forall\text{-weakening} \\
\frac{\frac{dom(hasVL, Movie) \quad \forall hasVL.\perp \sqsubseteq Movie}{\top \sqsubseteq Movie} \text{R17}}{Person \sqsubseteq \neg(Movie) \quad \top \sqsubseteq Movie} \text{DisjointFromTopSynonym} \\
\frac{}{Person \sqsubseteq \perp}
\end{array}
\tag{a}$$

Obtain $RRated \sqsubseteq Movie$ by R12 from
 $CatMovie \sqsubseteq Movie$ and $RRated \sqsubseteq CatMovie$

Obtain $(\exists hasScript. ThrillerScript \sqcup \forall hasVL.High) \sqsubseteq RRated$ by R1 from
 $RRated \equiv (\exists hasScript. ThrillerScript \sqcup \forall hasVL.High)$

Obtain $\forall hasVL.High \sqsubseteq RRated$ by R4 from
 $(\exists hasScript. ThrillerScript \sqcup \forall hasVL.High) \sqsubseteq RRated$

Obtain $\forall hasVL.\perp \sqsubseteq RRated$ by \forall -weakening from
 $\forall hasVL.High \sqsubseteq RRated$.

Obtain $\forall hasVL.\perp \sqsubseteq Movie$ by R12 from
 $RRated \sqsubseteq Movie$ and $\forall hasVL.\perp \sqsubseteq RRated$

Obtain $\top \sqsubseteq Movie$ by R17 from
 $dom(hasVL, Movie)$ and $\forall hasVL.\perp \sqsubseteq Movie$

Obtain $Person \sqsubseteq \perp$ by DisjointFromTopSynonym from
 $Person \sqsubseteq \neg(Movie)$ and $\top \sqsubseteq Movie$

(b)

Fig. 2. Rule-based derivation of the entailment in [13, p.221]

2.2 Tactics for OWL

Tactics represent commonly used proof patterns. In this work, we define a tactic as a high-level inference rule (represented as a sequent style rule), for which an alternative derivation exists with the same premises and the same conclusion, but made up of several inference steps at a lower level. The more detailed version of the derivation is called the *expansion* of the tactic, and can be considered an alternative explanation of the inference.

To represent such hierarchical proofs with several levels, we employ the generic conception of the proof data structure PDS developed for the mathematical assistant system Omega [14] (however, in our work we do not make use of proof forests at this stage, which are also part of the PDS concept). More precisely, proofs are represented as directed acyclic graphs, where sequent for-

mulae are linked to their premises via justification⁵ nodes. There can be several justifications for the same sequent, which are ordered hierarchically. As an illustrative example, we consider the entailment in Fig. 2 again. Rule R17 (as shown in Fig. 1) is implemented as a tactic, and expansion results in two inference steps using the inference rules \top -Cases and Def.Domain (R17 is part of the catalog of entailments in [1], whereas \top -Cases and Def.Domain are extra rules that are not part of Nguyen et al.’s catalog), as shown in Fig. 3. The inference rules that constitute \top -Cases and Def.Domain are presented in Fig. 4. In Fig. 3, rectangular boxes represent sequents, and rounded boxes the inference rules that “justify” the derivations, which connect a sequent to its premise sequent(s) (in the figure, from left to right). For ease of readability, those formulae that are added to a sequent by the latest rule application are printed in bold. When the last step of the expansion of a tactic finally links to the premise sequent of that tactic, it may happen that some of the formulae that were derived intermittently are not considered (formally, this corresponds to weakening). In our example, this is the case for $\exists \text{hasVL}.\top \sqsubseteq \text{Movie}$. However, not much is lost – if a proof is complete using a particular tactic, of course it remains complete when the expansion is added in this way. The dashed arrow represents a hierarchical ordering between the justifying inference rules for the sequent on the left in the figure.

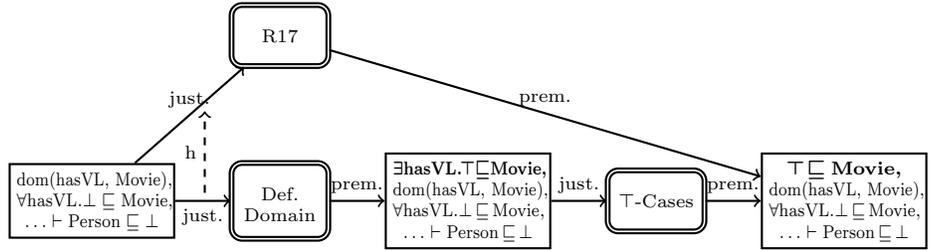


Fig. 3. Expansion of tactic R17 in the running example from Fig.2

When tactics are used, there is a choice between different levels for presenting the derivation/proof to the user. In the example, instead of presenting the step using R17, the two more fine-grained steps using \top -Cases and Def.Domain can be shown alternatively. Any complete proof alternative that can be generated from this hierarchical representation is called a “view” in the terminology of [14]. Previous work has explored the usefulness of enabling users to switch between such views in a mathematical assistant system [9]. Another prospect is to use measures for the difficulty (or “cognitive complexity” [13]) to make an informed selection of the right level(s) of justifications for a particular user, in the spirit

⁵ Note that here, “justification” takes a slightly different meaning than in [5]; a sequent is considered justified when it follows from one or several premise sequent(s) by means of an inference application.

$$\text{T-Cases} \frac{\Gamma, \top \sqsubseteq X, \exists r. \top \sqsubseteq X, \forall r. \perp \sqsubseteq X \vdash \Delta}{\Gamma, \exists r. \top \sqsubseteq X, \forall r. \perp \sqsubseteq X \vdash \Delta}$$

(a)

$$\text{Def.Domain (left)} \frac{\Gamma, \exists r. \top \sqsubseteq X, \text{dom}(r, X) \vdash \Delta}{\Gamma, \text{dom}(r, X) \vdash \Delta}$$

(b)

Fig. 4. Inference rules employed in the example in Fig. 3

of [7]. For example, if the model for difficulty/complexity determines that R17 is understandable to the user (possibly depending on some form of user model, or the user’s choice, or a general assumption about the typical user of the system), R17 is shown, whereas if it is considered too difficult, it can be split up into the two rule applications at the lower level.

2.3 Explanation Generation

The envisaged process for the generation of explanations with the presented system is a three stage process. First, for a given entailment and an ontology, a standard tableau-based reasoner (currently Hermit⁶) is called to find a minimal set of justifications according to [15]. Then, in the second stage, a simple proof search procedure is used that applies the tactics and sequent inference rules to find a proof as described above. This tandem construction of using two different inference mechanisms (both tableau-based and sequent rule-based) is used since a justification is a better starting point for constructing the sequent derivation than the entire ontology. Further work will investigate in how far the construction of the sequent-style proofs with tactics can be informed by the tableau proof obtained in the first step. For instance, Borgida et al. [12] have shown in the context of the \mathcal{ALC} fragment that the trace of the (efficient) tableau construction can be used to build sequent-style proofs for the purpose of explanation without any further search, but this result needs to be extended to the other patterns of inference available in the system presented here. We consider the work in [12] a good basis to further explore this issue.

The third stage consists in producing readable output from the generated proof(s). This involves the selection within the hierarchical levels of a proof (i.e. the selection of the “view”), and the generation of a human-readable presentation. At current, only a relatively simple pretty-printer is used (resulting in the output illustrated in Fig. 2), but a graphical interface (to draw the proofs as graphs, and to visualize hierarchical layers) or a natural language output facility would be useful additions.

⁶ www.hermit-reasoner.com/

3 Discussion

This work combines several ideas and hypotheses that have been previously investigated by related work more or less in isolation;

- that for the purpose of explanations, entailments in OWL are suitably represented by explicit derivations, rather than pointing to the relevant axioms alone,
- that different kinds of inference schemata are more difficult to perform and/or to understand than others,
- that hierarchical proofs offer the flexibility to adapt the level of abstraction of derivations to the needs of the user.

This echoes efforts in the general community of mathematical assistant systems and automated reasoning that have sought to facilitate human-oriented proofs, as opposed to “machine-oriented” proof mechanisms [16]. Such work has pointed to challenges that are specific to such systems, in particular with respect to efficiency. So far, the efficiency of our tandem construction for proof generation (fast search for justifications using a standard tableau-based reasoner, and relatively slow rule-based proof search with large rule sets on the simplified search problem) has not been evaluated, an important topic that remains for future work. In fact, it is to be expected that relatively rich rule sets (such as the 51-rule set generated from the inferences in [1]) will result in a large branching factor, and that naive, non-heuristic search will reach the limits of acceptable performance fairly quickly.

While the work presented here argues for the benefits of explicit derivations to explain entailments, in contrast to the “justification-oriented” approach (e.g. [15]), the question of which one of those competing approaches is the most suitable is not likely to result in an all-or-nothing answer. For example, justification-based explanations certainly have an advantage in situations where, for instance, the logical structure of the problem is of a well-known or repetitive form (e.g., a chain of subsumptions, or even simple OWL 2 EL justifications with chains of existential restrictions and atomic subsumptions). In such a case, any additional information (such as showing a derivation) would most likely be more harmful than helpful. The argument should rather be addressed by a comparison of the two approaches taking into account the background of the user and the complexity of the derivations to be presented.

A further and important topic for empirical investigation to be addressed in future work is to determine the advantage of using hierarchical proofs/tactics (and empirically investigated rule sets such as those from [1]) for the explanation of entailments over previous approaches using sequent-style derivations, such as [12].

Another question is in how far such systems should mirror “human” reasoning, in particular since there is controversy on the question “how do people reason” between different schools of thought in cognitive psychology. For example, some have argued (e.g. [17]) that human reasoning is essentially incomplete, since humans were found to refuse certain inferences that are logically valid (this interpretation could be transferred to those inferences that proved to be most

difficult in [1]). For some classes of inferences (e.g. syllogisms), competing kinds of theories exist (for instance, rule-based theories of inference, e.g. [17], vs. mental models [18]). In our current work we submit to inference rules, since they provide a middle ground between their suitability for automated reasoning and their psychological plausibility as determined by cognitive psychologists.

A predominant perspective in previous work on the difficulty of inferences in ontologies (e.g. [1, 2]) is that of difficulty being an essentially static property of inference problems, i.e. individual differences in knowledge, mastery, or learning have not been taken into account so far. Similarly, most experiments on reasoning in cognitive psychology have been performed with untrained subjects, to establish the “natural” ability of people to reason logically. Experiments involving both learning and reasoning are more complex to set up and analyze. However, most courses in logic are built on the assumption that instruction and training will increase the learners’ ability over time, therefore the hypothesis that the difficulty of inferences will decrease with practice seems to be a plausible one. In learning, a process called “rule automatization” [19] is known. Taking this into account, future work should not only study whether certain inferences are generally more difficult for a particular group of people (at a specific level of expertise) than others (which has been done in cognitive psychology, usually with the level of expertise being fixed at “naive”), but also study how the specific difficulty changes over time as a result of experience and learning. The presented framework will be useful to model hypotheses and generate test problems for empirical experiments that help to discriminate between different such hypotheses (for example, whether the relative differences in difficulty between easy and difficult problems for novices are still to be found in experts, or whether these problem-specific differences disappear with practice).

Finally, since this article describes a system that is currently work in progress, future work will examine different aspects of usefulness of such a framework, both from the viewpoint of practitioners (e.g. in the case of ontology debugging), and for a deeper investigation of metrics associated with the difficulty of sequences of inferences in a knowledge representation setting.

Acknowledgements. We acknowledge the support of the Transregional Collaborative Research Center SFB/TRR 62 “A Companion-Technology for Cognitive Technical Systems” funded by the German Research Foundation (DFG). We thank three anonymous reviewers for their useful comments.

References

1. Nguyen, T., Power, R., Piwek, P., Williams, S.: Measuring the understandability of deduction rules for OWL. In: First International Workshop on Debugging Ontologies and Ontology Mappings (WoDOOM12). (2012) 1–12
2. Horridge, M., Bail, S., Parsia, B., Sattler, U.: The cognitive complexity of OWL justifications. In Rosati, R., Rudolph, S., Zakharyashev, M., eds.: Proceedings of the 24th International Workshop on Description Logics (DL 2011), Barcelona, Spain. Volume 745 of CEUR Workshop Proceedings. (2011)
3. Horridge, M., Parsia, B., Sattler, U.: Lemmas for justifications in OWL. In Cuenca Grau, B., Horrocks, I., Motik, B., Sattler, U., eds.: Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK. Volume 477 of CEUR Workshop Proceedings. (2009)
4. McGuinness, D., Ding, L., Glass, A., Chang, C., Zeng, H., Furtado, V.: Explanation interfaces for the semantic web: Issues and models. In: Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI'06). Accessed 19 April 2013 from ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-06-14.pdf.
5. Kalyanpur, A.: Debugging and repair of OWL ontologies. PhD thesis, University of Maryland (2006)
6. Horridge, M., Parsia, B., Sattler, U.: Justification oriented proofs in OWL. In: ISWC'10 Proceedings of the 9th International Semantic Web Conference, November 2010, Shanghai, China, Volume Part I, Springer, Berlin 354–369
7. Schiller, M.R.G.: Granularity analysis for mathematical proofs. *Topics in Cognitive Science* **5** (2013) 251–269
8. Alexoudi, M., Zinn, C., Bundy, A.: English summaries of mathematical proofs. In Benz Müller, C., Windsteiger, W., eds.: Second International Joint Conference on Automated Reasoning – Workshop on Computer-Supported Mathematical Theory Development, Cork, Ireland, July 5, 2004. Number 04-14 in RISC Report Series, RISC Institute, University of Linz (2004) 49–60
9. Fiedler, A.: *P.rex*: An interactive proof explainer. In Goré, R., Leitsch, A., Nipkow, T., eds.: Automated Reasoning – Proceedings of the First International Joint Conference (IJCAR 2001), Siena, Italy, June 18-23, 2001. Number 2083 in LNAI, Springer, Berlin, Heidelberg (2001) 416–420
10. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web* **6**(4) (2008) 309–322
11. Horridge, M., Bechhofer, S.: The OWL API: A java API for OWL ontologies. *Semantic Web Journal* **2**(1) (2011) 11–21
12. Borgida, A., Franconi, E., Horrocks, I.: Explaining ALC subsumption. In Horn, W., ed.: Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), IOS Press, Amsterdam (2000) 209–213
13. Horridge, M.: Justification based explanations in ontologies. PhD thesis, University of Manchester (2011)
14. Autexier, S., Benz Müller, C., Dietrich, D., Meier, A., Wirth, C.P.: A generic modular data structure for proof attempts alternating on ideas and granularity. In Kohlhase, M., ed.: Proceedings of the 5th International Conference on Mathematical Knowledge Management (MKM'05), Bremen, Germany, July 15-17, 2005. Volume 3863 of LNCS, Springer, Berlin, Heidelberg (2006) 126–142
15. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: ISWC'07/ASWC'07 Proceedings of the 6th International

- Semantic Web Conference and the 2nd Asian Semantic Web Conference, Springer, Berlin, Heidelberg (2007) 267–280
16. Bundy, A.: A survey of automated deduction. Technical Report EDI-INF-RR-0001, Edinburgh University (1999)
 17. Rips, L.J.: The psychology of proof : deductive reasoning in human thinking. MIT Press, Cambridge, MA (1994)
 18. Johnson-Laird, P.: Mental models and human reasoning. Proceedings of the National Academy of Science of the United States of America **107**(43) (2010) 18243–18250
 19. Sweller, J.: Cognitive technology: Some procedures for facilitating learning and problem solving in mathematics and science. Journal of Educational Psychology **81** (4) (1989) 457–466

Appendix

In the following, we list the inference rules used in the example derivation in Fig. 2.

$$\frac{\Gamma, X \sqsubseteq Y, X \equiv Y \vdash \Delta}{\Gamma, X \equiv Y \vdash \Delta} \text{R1}^a$$

$$\frac{\Gamma, Y \sqsubseteq Z, (X \sqcup Y) \sqsubseteq Z \vdash \Delta}{\Gamma, (X \sqcup Y) \sqsubseteq Z \vdash \Delta} \text{R4}$$

$$\frac{\Gamma, X \sqsubseteq Z, X \sqsubseteq Y, Y \sqsubseteq Z \vdash \Delta}{\Gamma, X \sqsubseteq Y, Y \sqsubseteq Z \vdash \Delta} \text{R12(left)}$$

$$\frac{\Gamma, \top \sqsubseteq X, \text{dom}(r, X), \forall r. \perp \sqsubseteq X \vdash \Delta}{\Gamma, \text{dom}(r, X), \forall r. \perp \sqsubseteq X \vdash \Delta} \text{R17(left)}$$

$$\frac{\Gamma, \forall r. \perp \sqsubseteq X, \forall r. X \sqsubseteq Y, \vdash \Delta}{\Gamma, \forall r. X \sqsubseteq Y \vdash \Delta} \forall\text{-weakening}$$

$$\frac{\Gamma, X \sqsubseteq \perp, X \sqsubseteq \neg Y, \top \sqsubseteq Y \vdash \Delta}{\Gamma, X \sqsubseteq \neg Y, \top \sqsubseteq Y \vdash \Delta} \text{DisjointFromTopSynonym}$$

^a We assume that n-ary class-equivalence axioms are normalized into binary class-equivalence axioms, such that the rule for the binary case suffices.