iCon: A Diagrammatic Theorem Prover for Ontologies

Zohreh Shams and Mateja Jamnik
Department of Computer Science and Technology
University of Cambridge, UK
{zohreh.shams, mateja.jamnik}@cl.cam.ac.uk

Gem Stapleton and Yuri Sato
Centre for Secure, Intelligent and Usable Systems
University of Brighton, UK
{y.sato, g.e.stapleton}@brighton.ac.uk

Abstract

Concept diagrams form a visual language that is aimed at non-experts for the specification of ontologies and reasoning about them. Empirical evidence suggests that they are more accessible to ontology users than symbolic notations typically used for ontologies (e.g., DL, OWL). Here, we report on iCon, a theorem prover for concept diagrams that allows reasoning about ontologies diagrammatically. The input to iCon is a theorem that needs proving to establish how an entailment, in an ontology that needs debugging, is caused by a minimal set of axioms. Such a minimal set of axioms is called an entailment justification. Carrying out inference in iCon provides a diagrammatic proof (i.e., explanation) that shows how the axioms in an entailment justification give rise to the entailment under investigation. iCon proofs are formally verified and guaranteed to be correct.

Introduction

Ontologies are used by diverse stakeholders and domain experts. However, domain experts are often not familiar with symbolic notations in which ontologies are expressed. To address this issue, by appealing to the long-held assumption that using diagrams makes modelling and knowledge representation accessible (e.g., Euclid’s Elements, (Sowa 1984)), there have been attempts (Lembo et al. 2016; Brockmans et al. 2004; Falco et al. 2014; Liepins, Grasmanis, and Bojars 2014) to express ontologies using graphical notations. Concept diagrams (CDs) (Stapleton, Compton, and Howse 2017) are a recent visual language that was developed to specify ontologies, and covers all of OWL 2 except assertions involving ObjectHasSelf, DatatypeRestriction or constraining facets (Stapleton, Compton, and Howse 2017). Empirical studies (Hou, Chapman, and Blake 2016; Alharbi et al. 2017; Sato et al. 2018) demonstrate the accessibility of CDs compared to competing diagrammatic and symbolic notations, including OWL and description logics (DL) (Baader, Horrocks, and Sattler: 2009) and SOVA (Itzik and Reinhartz-Berger 2014). Unlike CDs, some graphical notations for ontologies are based on conceptual modelling languages such as Entity-Relationship (ER) schemas. However, with the exception of (Lembo et al. 2016), they require a combination of diagrammatic and textual formulae. Different from (Lembo et al. 2016), the design of CDs is motivated by the need to be accessible to people without training in formal languages/logic (Hou, Chapman, and Blake 2016; House et al. 2011).

Our contribution is the design of a theorem prover, iCon, with which proofs can be constructed in the graphical and accessible language of CDs. The input to iCon is a theorem that needs proving to establish how an entailment follows from specified axioms. iCon’s inference engine is equipped with diagrammatic versions of symbolic inference rules for OWL (OWL2 2018). According to the World Wide Web Consortium (W3C) (W3C 2018), these rules (listed in (W3CInf 2018)) provide a useful starting point for the practical implementation of ontology reasoners. iCon produces diagrammatic proofs that explain how the axioms give rise to the entailment.

The Concept Diagram Language

Concept diagrams consist of rectangles, closed curves, and shading (as seen in Euler and Venn diagrams) as well as additional objects such as dots, solid arrows and dashed arrows; for a formalisation, see (Stapleton et al. 2013), here we introduce the notation by example.

In Figure 1, there is one concept diagram containing two boundary rectangles. Within each rectangle, spatial relationships are used to convey information. For example, Person and Animal represent disjoint sets, since the two corresponding curves are disjoint. We can also see that Helen is a Female person, due to the location of the (red) dot l-
denoted by $\Delta$. Prm$_{\text{initial proof state}}$ Prm$_0$ that if where $\Delta$ Prm$_{\text{final proof state}}$, say $\Delta$. Animals have Colour Male Helen belled spider by lines; is called a proof of rectangles, the diagram does not assert that Colour that an animal can have cannot be outside the set Colour. Together with the arrow annotation $\geq 1$, this means that all animals have at least one colour. Note that, due to the use of rectangles, the diagram does not assert that Colour is disjunct from the other sets visualised here. Lastly, we note that $iCon$ makes use of $\top$ and $\bot$ to represent ‘true’ and ‘false’ respectively, as a simple shorthand for valid and contradictory diagrams.

**iCon: System Description**

$iCon$ is a diagrammatic theorem prover$^2$ for CDs, with which, for the first time, explanatory proofs for ontology entailments can be constructed in a graphical and accessible language. The input to $iCon$ is a theorem that needs proving to establish how an ontology entailment (i.e., theorem) follows from its justification axioms. The result of carrying out inference in $iCon$ is a diagrammatic proof that explains this. In what follows, we explain the two main components of $iCon$, namely its reasoning engine and its graphical user interface.

**Reasoning Engine**

The $iCon$ reasoning engine (i) contains a collection of inference rules; (ii) handles the application of inference rules to diagrams; and (iii) manages proofs.

**Proof** A proof in $iCon$ starts with the initial proof state, denoted by $\Delta_0$, which is of the form $(d_1 \land \ldots \land d_m) \Rightarrow d$, where $d_i$ and $d$ are CDs. This means that we want to prove that if $d_1, \ldots, d_m$ (the premises) hold then $d$ (the conclusion) holds. Let the set of premises in each proof state be Prm$_0$($\Delta_0$), and the proof goal be $d$. Proofs in $iCon$ are constructed by applying inference rules to the premises of the initial proof state Prm$_0$($\Delta_0$) in a forward reasoning manner. A theorem is proved when the application of inference rules makes Prm$_0$($\Delta_0$) identical to the proof goal $d$. The final proof state, say $\Delta_k$, is of the form $d \Rightarrow d$, which is trivially true, and is referred to as the basic proof state $\Delta_{\text{basic}}$.

Applying a single inference rule to a proof state $\Delta$ is denoted by $\Delta \Rightarrow$ Rule, where the result is a proof state $\Delta'$, such that Prm($\Delta$) syntactically and semantically entails Prm($\Delta'$) (i.e., Prm($\Delta$) $\vdash$ Prm($\Delta'$) and Prm($\Delta$) $\models$ Prm($\Delta'$)). An exception is the inference rule $\text{Identity}$ that is applied to the basic proof state and concludes the proof: $\Delta_{\text{basic}} \models \text{Identity}$.  

---

$^2$Available at https://github.com/ZohrehShams/iCon.

**Inference Rules** Since $iCon$ is a purpose built tool for ontology reasoning, the basis for its diagrammatic inference rules is the ontology community’s standard set of inference rules. These rules are introduced by W3C and listed in (W3CInf 2018). In order to construct a proof for a justification-entailment pair, we are equipping $iCon$’s inference engine with diagrammatic versions of the symbolic inference rules for OWL (OWL2 2018). In addition to diagrammatic inference rules, $iCon$ has two logical inference rules, namely: Conjunction Elimination $(d_1 \land d_2) \Rightarrow d$ and Identity which was mentioned in the previous section. If $d$ and $d'$ are isomorphic$^3$ CDs, we can apply rule Identity and infer $\top$: $d \Rightarrow \bot$.

Diagrammatic inference rules rewrite the diagrams representing the premises of a proof state in order to make them identical to the goal of the proof state. In contrast to a symbolic proof, which typically is inaccessible to domain experts who are not proficient in symbolic languages, this results in a diagrammatic proof, which is empirically-evidenced to be more accessible. To demonstrate this in more detail, in what follows we present a symbolic and a diagrammatic proof of a theorem that aims at debugging an undesired entailment of an ontology (i.e., an inconsistency).

In Figure 2, there are five axioms that count as a justification for an inconsistency in some ontology$^4$. Below is the theorem that needs proving to show why and how the inconsistency is caused:

**Theorem 1** 
\[
\text{Cat } \sqsubseteq \forall \text{isPetOf } \land \text{Cat(Re)} \land \text{isPetOf(Re, Alex)} \land \text{Male(Alex)} \Rightarrow \bot
\]

The symbolic proof of this theorem using W3C inference rules in presented in Figure 3. The first inference rule used in the proof is:
\[
\frac{X \subseteq \forall P \cdot Y \land X(u) \land P(u, v)}{Y(v)}
\]

which expresses that if $X$ is only related to $Y$ under $P$, and $u$ is of type $X$, and $u$ is related to $v$ under $P$, we can conclude that $v$ is of type $Y$. The second inference rule is:
\[
\frac{\text{Dis}(X, Y) \land X(u) \land Y(u)}{\bot}
\]

which says that if two sets are disjoint and there is an element that belongs to both of them then we have a contradiction and can infer false.

---

$^3$Isomorphism of CDs is defined in the same fashion as that of Spider Diagrams (Stapleton et al. 2004).

$^4$The off-the-shelf reasoners can give a justification for any inconsistency, in the form of a minimal set of axioms that has caused the inconsistency, however no causal conection that leads to the inconsistency is provided.
The first four and the last three inference steps in the proof explained above, represent a diagrammatic version of the four inference rules (W3CInf 2018). In doing so, we are currently conducting more empirical studies to inform us about the most accessible diagrammatic representation for non-experts.

Graphical User Interface
iCon’s GUI enables inputting CDs and proof states in an abstract textual representation format. It then visualises them, based on the algorithm for Euler diagrams in (Stapleton et al. 2012). The GUI also enables the construction of a diagrammatic proof by offering users different inference rules to apply to any diagram or elements within it with a point and click mechanism. The successful application of inference rules transforms the diagrams in the proof state, and generates a new one that is then visualised.

Proof states are stored as indexed trees. When an inference rule is applied, the tree for the proof state in which the diagram is situated is traversed in the search of the diagram(s) that is the target of inference. If this diagram(s) and the possible element(s) chosen from it satisfy all the requirements for a sound application of the rule, the rule is applied and the affected diagram(s) is transformed appropriately.

Conclusion and Future Work
Building an error-free, high-quality ontology is not an easy task. There are ontology reasoners which generate justifications for entailments that follow from an ontology, so that the undesired ones can be eliminated through debugging. But these justifications remain opaque to ontology engineers. Here, we reported on an ontology reasoner, iCon, that has two main advantages over existing approaches. First, it is capable of generating an explanation in terms of a proof that exposes how the interaction between the axioms in the justification brings about an undesired entailment. Second, its explanations are in a diagrammatic language for which empirical studies suggest more accessibility than symbolic notations. Indeed, iCon is the first tool that can provide a diagrammatic explanation for debugging ontologies.

A future goal is to evaluate the accessibility of iCon through usability studies with ontology engineers. Another avenue for future work is taking iCon from an interactive theorem prover toward an automated one to automatically generate diagrammatic explanations for undesired ontology entailments. We have already experimented (Shams et al. 2018) with the use of tactics (Harrison, Urban, and Wiedijk 2014). Tactics are programs that encapsulate sequences of inference rules to achieve a higher level of abstraction and automation. We are continuing this line of work for automation in iCon.

Other technical improvements are also in the pipeline. One is devising a more effective visualisation layout algorithm that preserves the shape and location of the invariant parts of diagrams before and after applying inference rules. Another one is developing a drag and drop visual tool for constructing diagrammatic theorems (to replace the current abstract textual representation input method).
Figure 4: A diagrammatic proof for Theorem 1. Which proof is more explanatory: the diagrammatic proof in this figure or the symbolic one in Figure 3?
Acknowledgements

This research was funded by a Leverhulme Trust Research Project Grant (RPG-2016-082) for the project entitled Accessible Reasoning with Diagrams.

References


