# 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.[1]

*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; Howse et al. 2011).

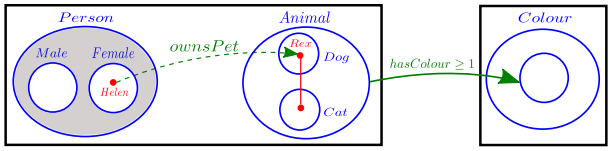


Figure 1: CD Examples

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 la-



[1]Note that we distinguish between visualisation facilities offered by popular ontology editing tools such as Protégé (Protege 2018), that merely visualise ontology axioms already specified in symbolic notations, and those we review here that allow specifying and editing ontologies using diagrammatic notations directly.

belled Helen. The dot, and more generally dots connected by lines, is called a *spider*. For examples, due to the location of spider Rex, we can see that Rex is either a Cat or a Dog. The region outside of Male and Female is shaded. This means that there is no person who is neither a Female nor a Male. The dashed arrow ownsPet connects the dot Helen to Rex. This means that Helen owns Rex as a her pet, but she can own other pets too. Unlike, dashed arrows, solid arrows mean that the source is related to *only* the target. The solid arrow labelled hasColour connects Animal to the unnamed circle inside Colour, meaning that the colours 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 disjoint 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 $(d_1 \wedge \cdots \wedge d_m) \Rightarrow d$, where $d_i$ and $d$ are CDs. This means that we want to prove that if $d_1, \cdots, d_m$ (the premises) hold then $d$ (the conclusion) holds. Let the set of premises in each proof state be $Prm(\Delta_i)$, and the proof goal be $d$. Proofs in iCon are constructed by applying inference rules to the premises of the initial proof state $Prm(\Delta_0)$ in a forward reasoning manner. A theorem is proved when the application of inference rules makes $Prm(\Delta_0)$ 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 $\Delta_{basic}$.

Applying a single inference rule to a proof state $\Delta$ is denoted by $\frac{\Delta}{\Delta'} Rule$, 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 *Identity* that is applied to the basic proof state and concludes the proof: $\frac{\Delta_{basic}}{\top} Identity$.

[2]Available at https://github.com/ZohrehShams/iCon.

$$Cat \sqsubseteq \forall isPetOf \cdot Female \quad isPetOf(Rex, Alex)$$
$$Dis(Male, Female) \quad Cat(Rex) \quad Male(Alex)$$

Figure 2: The inconsistent set of axioms

**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* ($\frac{(d_1 \wedge d_2) \Rightarrow d}{d_1 \Rightarrow d}, \frac{(d_1 \wedge d_2) \Rightarrow d}{d_2 \Rightarrow d}$) and *Identity* which was mentioned in the previous section. If $d$ and $d'$ are isomorphic[3] CDs, we can apply rule *Identity* and infer $\top$: $\frac{d \Rightarrow d'}{\top}$.

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 contrst to a symbolic proof, which typically is inaccessibe to domain experts who are not proficient in symbolic languages, this results in a diagrammatic proof, whcih 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** $Cat \sqsubseteq \forall isPetOf \cdot Female \wedge Cat(Rex) \wedge isPetOf(Rex, Alex) \wedge Male(Alex) \wedge Dis(Male, Female) \Rightarrow \bot$

The symbolic proof of this theorem using W3C inference rules in presented in Figure 3. The first inference rule used in the proof is:

$$\frac{X \sqsubseteq \forall P \cdot Y \wedge X(u) \wedge P(u, v)}{Y(v)} cls-avf$$

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:

$$\frac{Dis(X, Y) \wedge X(u) \wedge Y(u)}{\bot} cax-dw$$

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]Isomorphisim 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.

$$\dfrac{\dfrac{Cat \sqsubseteq \forall isPetOf \cdot Female \wedge Cat(Rex) \wedge isPetOf(Rex, Alex) \wedge Male(Alex) \wedge Dis(Male, Female)}{Female(Alex) \wedge Male(Alex) \wedge Dis(Male, Female)}\, cls-avf}{\bot}\, cax-dw$$

Figure 3: A symbolic proof for Theorem 1

Figure 4 shows a digrammatic version of the same proof as in Figure 3, but unlike the symbolic proof, the diagrammatic one reveals how the interaction between the axioms in the justification brings about an undesired entailment.

The initial proof state, shows the diagrammatic representation of Theorem 1, where F, C, M and isP, stand for Female, Cat, Male and isPetOf. In the first inference step, Alex is copied from the second (from left) diagram to the first diagram and the secod diagram is deleted. Since there is a solid arrow from Cat to a subset of Female, every element of Cat is related to that subset via the same arrow. This has been made explicit via the next inference step, where the solid arrow is sourced at Rex. Due to smentic of solid arrow, Rex can *only* be related to the target of this arrow under isPetOf relation. Therefore if Rex is related to Alex, Alex has to be in the target of the solid arrow in the first diagram. Thus, $AddSpidertoSolidArrowImage$ copies Alex from the second diagram to the first one, followed by deleting the second diagram. $DeleteSyntax$ rule then allows deleting any extra piece of syntax from the first diagram to decrease the clutter while preserving what is needed for the next inference step. $CopyCurve$ is the next inference rule that copies curve Male from the second diagram to the first one, with respect to the location of spider Alex . The second diagram is deleted too. Now, in proof state 5, there are two diagrams, one expressing that Alex is a Male and a Female, and the other one expressing Male and Female are disjoint. The disjointness means that the intersection between Male and Female is empty which is expressed by shadig in CD language. By applying $AddAllMissingZones$ this has been made explicit in the seocond diagran. Proof state 6, clearly highlights the contradiction by having the same zone (intersection of Male and Female) both as non-empty and empty.

The first four and the last three inference steps in the proof explained above, represent a diagrammatic version of $cls-avf$ and $cax-dw$, respectively. We use one possible mapping of $cls-avf$ and $cls-dw$, and there might exist several other mappings, because any symbolic inference rule may give rise to several diagrammatic ones. Since iCon is designed to provide a graphical and accessible explanation for ontology reasoning tasks, we base these choices on evidence from our cognitive empirical studies about what humans find more accessible, such as (Shams et al. 2018). There are currently 18 inference rules in iCon. We are expanding this set to capture all of the W3C OWL 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).

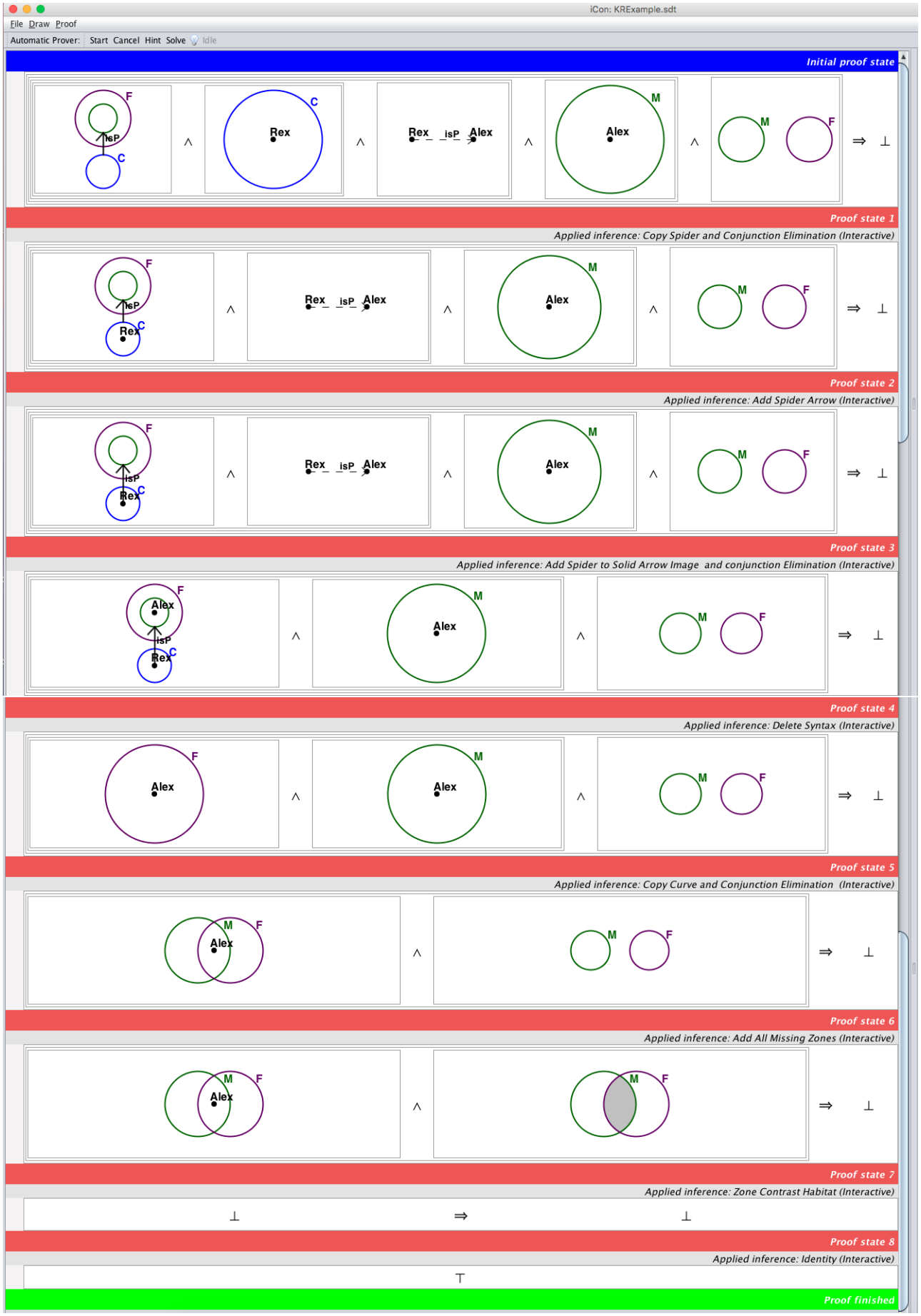


Figure 4: A diagrammatic proof for Theorem 1. Which proof is more explanatory: the diagrammatic proof in this figure or the symbolic one in Figure 3?

## Acknowledgements

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

## References

Alharbi, E.; Howse, J.; Stapleton, G.; Hamie, A.; and Touloumis, A. 2017. Visual logics help people: An evaluation of diagrammatic, textual and symbolic notations. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, 255–259. IEEE.

Baader, F.; Horrocks, I.; and Sattler:, U. 2009. Description logics. In *Handbook on Ontologies*. Springer. 21–43.

Brockmans, S.; Volz, R.; Eberhart, A.; and Löffler, P. 2004. Visual modeling of OWL DL ontologies using UML. In *The Semantic Web*, volume 3298 of *Lecture Notes in Computer Science*, 198–213. Springer.

Falco, R.; Gangemi, A.; Peroni, S.; Shotton, D. M.; and Vitali, F. 2014. Modelling OWL ontologies with graffoo. In *ESWC 2014 Satellite Events*, volume 8798 of *LNCS*, 320–325. Springer.

Harrison, J.; Urban, J.; and Wiedijk, F. 2014. History of interactive theorem proving. In *Computational Logic*, volume 9. Elsevier. 135–214.

Hou, T.; Chapman, P.; and Blake, A. 2016. Antipattern comprehension: An empirical evaluation. In *Formal Ontology in Information Systems*, volume 283 of *Frontiers in Artificial Intelligence*, 211–224. IOS Press.

Howse, J.; Stapleton, G.; Taylor, K.; and Chapman, P. 2011. Visualizing ontologies: A case study. In *International Semantic Web Conference*, 257–272. Springer.

Itzik, N., and Reinhartz-Berger, I. 2014. SOVA - A tool for semantic and ontological variability analysis. In *Joint Proceedings of the CAiSE 2014 Forum and CAiSE 2014 Doctoral Consortium*, volume 1164, 177–184. CEUR-WS.org.

Lembo, D.; Pantaleone, D.; Santarelli, V.; and Savo, D. F. 2016. Easy OWL drawing with the graphol visual ontology language. In *Principles of Knowledge Representation and Reasoning*, 573–576. AAAI Press.

Liepins, R.; Grasmanis, M.; and Bojars, U. 2014. Owlgred ontology visualizer. In *ISWC Developers Workshop 2014*, volume 1268, 37–42. CEUR-WS.org.

2018. The OWL2 web ontology language. `https://www.w3.org/TR/owl2-direct-semantics/`, retrieved May 2018.

2018. Protégé: A free, open-source ontology editor. `http://protege.stanford.edu`, retrieved May 2018.

Sato, Y.; Stapleton, G.; Jamnik, M.; and Shams, Z. 2018. Deductive reasoning about expressive statements using external graphical representations. In *Cognitive Science Society*. Cognitive Science Society. Forthcoming.

Shams, Z.; Jamnik, M.; Stapleton, G.; and Sato, Y. 2018. Accessible reasoning with diagrams: from cognition to automation. In *Diagrams*, LNCS. Springer. Forthcoming.

Sowa, J. F. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Longman Publishing Co., Inc.

Stapleton, G.; Howse, J.; Taylor, J.; and Thompson, S. J. 2004. The expressiveness of spider diagrams. *Logic and Computation* 14(6):857–880.

Stapleton, G.; Flower, J.; Rodgers, P.; and Howse, J. 2012. Automatically drawing Euler diagrams with circles. *Journal of Visual Languages and Computing* 23(3):163–193.

Stapleton, G.; Howse, J.; Chapman, P.; Delaney, A.; Burton, J.; and Oliver, I. 2013. Formalizing concept diagrams. In *Visual Languages and Computing*, 182–187. Knowledge Systems Institute.

Stapleton, G.; Compton, M.; and Howse, J. 2017. Visualizing OWL 2 using diagrams. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, 245–253. IEEE.

2018. World Wide Web Consortium. `https://www.w3.org`, retrieved May 2018.

2018. Reasoning in OWL 2 RL and RDF graphs using rules. `https://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules`, retrieved May 2018.