On the Readability of Diagrammatic Proofs

Gem Stapleton[⋆]
[⋆]University of Brighton
Brighton, UK
g.e.stapleton@bton.ac.uk

Mateja Jamnik[†]
[†]University of Cambridge
Cambridge, UK
mateja.jamnik@cl.cam.ac.uk

Judith Masthoff[‡]
[‡]University of Aberdeen
Aberdeen, UK
j.masthoff@abdn.ac.uk

Abstract

Recently, much effort has been placed on developing diagrammatic logics, with a focus on obtaining sound and complete reasoning systems. A hypothesis of the diagrammatic reasoning community is that many people find diagrammatic proofs easier to read than symbolic proofs. This hypothesis has not been thoroughly tested, although significant effort has been directed towards understanding what makes diagrams more readable than symbolic formulae. We are interested in how to automatically find readable diagrammatic proofs. To achieve this aim, significant research is required that builds on the existing state-of-the-art. This extended abstract summarizes our plans for research on this topic.

1 Introduction

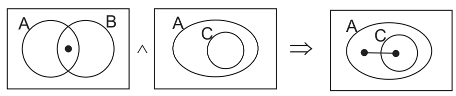
Diagrammatic reasoning has only recently, in the last decade, enjoyed research attention which demonstrated that diagrams *can* be used for formal reasoning. However, diagrammatic proofs have been so far constructed without much attention paid to their readability: a major hypothesis of the diagrammatic reasoning community, which is often assumed, is that they are more readable than symbolic proofs due to their intuitive visual appeal. This may not be the case for an arbitrary diagrammatic proof, and some diagrammatic proofs will be more appealing, easier to understand, or more economical with information than other diagrammatic proofs. In order to take full advantage of the widely acknowledged advantages of diagrammatic reasoning, we need to understand what constitutes, and how to create, readable diagrammatic proofs.

In symbolic reasoning, theorem provers have come a long way in devising strategies, criteria and heuristics for constructing proofs that are human readable, yet these symbolic proofs are still complex for humans to understand. By contrast, diagrammatic reasoners have paid virtually no attention to this readability issue. This extended abstract discusses a plan of work for addressing this issue in a principled way by producing a framework of readability criteria that can be employed in diverse diagrammatic reasoners in the form of heuristics or other higher level strategies. Indeed, we hope that diagrammatic readability criteria, and any principles learned about readability, will extend to symbolic reasoning systems too. Our main hypothesis is: *readable diagrammatic proofs can be constructed automatically by devising and employing readability criteria.* This represents a major challenge in diagrammatic and automated reasoning.

2 Diagrammatic Proofs

We demonstrate the kind of diagrammatic proofs we plan to examine, some possible readability criteria, and how these can be used to guide the search for a more readable diagrammatic proof with a toy example. Initially we will focus our work on the domain of spider diagrams [4] which are used to prove theorems that can be expressed in monadic first order logic with equality. Thus, spider diagrams make statements which provide constraints on set cardinality, such as $|A - B| = 2$, $|A \cap B| \geq 3$, and $A \subseteq C$ (equivalently, $|A - C| = 0$). Diagrams of this type are frequently seen in mathematics text books to intuitively illustrate set theory concepts. Spider diagrams are based on Euler diagrams, augmenting them with shading and so-called spiders. Visually, spiders are trees (dots connected by lines), placed in regions of the diagram; each spider represents the existence of an element in the set represented by the region in which it is placed, thus providing lower bounds on set cardinality. Shading is used to place upper bounds on set cardinality: in a shaded region, all elements are represented by spiders. Here is an example theorem expressed symbolically and using spider diagrams:

$$\exists x\,(x \in A \cap B \wedge C \subseteq A) \Rightarrow \exists x\,\big((x \in A - C \vee x \in A \cap C) \wedge C \subseteq A\big)$$



Two diagrammatic proofs of this theorem are shown in Fig. 1. Notice that the second proof is not only shorter than the first one, but also that its diagrams consist of fewer elements and can therefore be considered to be less cluttered. Spider diagrams are very strongly related to other systems, such as those in [7], and form the basis of more expressive notations making them an ideal choice for investigating readability. We have developed theorem provers for spider diagrams and their Euler diagram fragment but they do not yet incorporate readability criteria except for finding shortest proofs [8].

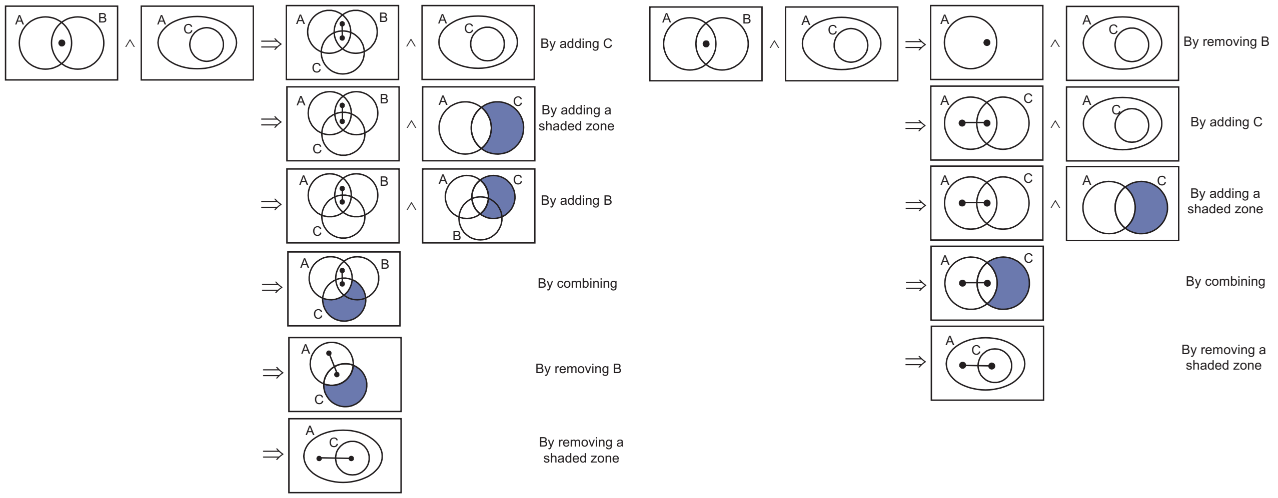


Figure 1: Two diagrammatic proofs.

We propose to devise and use *readability criteria* in order to construct more readable and intuitive diagrammatic proofs than currently possible. Lessons about the representation and use of heuristics, tactics [5], methodicals [6], strategies and so forth in symbolic theorem proving will inform our analysis. In addition, and most importantly, we will also experimentally test what people find more readable in order to identify readability criteria and relevant proof situations for them and also, later, to assess which readability criteria are better than others. Other possible indicators of readability are, for example: (1) diagram clutter: it is easier to read and understand diagrams that have fewer elements in them, (2) length of proof: often shorter proofs are more readable than longer ones, (3) known lemmas: if an inference rule leads to a statement proved before, then this lemma can be used rather than derived again, (4) the size of the step that an inference rule makes: too big steps may be obscure, but too small steps may be tedious, and (5) topological properties matching semantics: diagrams that are better matched to their semantics may be more readable.

Investigating symbolic theorem provers, especially proof planners [1], that already use techniques to guide search [2] will inform our work. In particular, proof planners such as λ*Clam* [6] and ISAPLANNER [3] use methods and methodicals to structure the search. Similarly, ΩMEGA uses control rules and strategies. It is not clear what will be the best framework for employing readability criteria in order to guide the search to produce readable proofs; devising this framework is one of our main goals. Once we have this framework, we will implement it in a diagrammatic theorem prover for spider diagrams. The hope is that any principles learned are general enough that they will extend to symbolic reasoners as well.

Acknowledgements Gem Stapleton is supported by EPSRC grant EP/E011160/1 for the Visualisation with Euler Diagrams project. Mateja Jamnik is supported by an EPSRC Advanced Fellowship GR/R76783/01.

References

[1] A. Bundy. The use of explicit plans to guide inductive proofs. 9th Conf. on Automated Deduction, Springer, 111–120, 1988.

[2] L. A. Dennis, M. Jamnik, and M. Pollet. On the comparison of proof planning systems: LambdaClam, Omega and IsaPlanner. ENTCS 151(1):93–110, 2006.

[3] L. Dixon. *A Proof Planning Framework for Isabelle*. PhD thesis, University of Edinburgh, UK, 2005.

[4] J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. *LMS J. of Computation and Mathematics*, 8:145–194, 2005.

[5] L.C. Paulson. *Isabelle: A generic theorem prover*. LNCS 828, Springer, 1994.

[6] J.D.C. Richardson, A. Smaill, and I. Green. System description: proof planning in higher-order logic with lambda-clam. *15th Conf. on Automated Deduction*, Springer, 129–133, 1998.

[7] S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.

[8] G. Stapleton, J. Masthoff, J. Flower, A. Fish, and J. Southern. Automated theorem proving in Euler diagrams systems. *J. of Automated Reasoning*, 39:431–470, 2007.