13.1 Logical Reasoning with Diagrams

Mateja Jamnik

Summary

This chapter examines the role of diagrams in reasoning. It exposes the tension between formal and informal use of diagrams. Diagrams are a complementary representation to symbolic formulae. Since choosing the right representation may be essential to solving a problem, we study diagrams as one such suitable choice. We analyze algorithmic and computational approaches to formalizing the logic of diagrams and present some modern implementations of diagrammatic reasoning systems.

1. Diagrams through Time

Humans have used diagrams to convey reasoning since the ancient times of Aristotle and Euclid. In fact, until the end of the 19th century, diagrams were considered a legitimate formal tool for explaining the rationale of the solution to a problem. They were a solution in themselves. But while Euler (1768), Venn (1881), and Peirce (1933) were devising logics for their diagrams, Hilbert, Gödel, Russell, and Wittgenstein formalized symbolic logic. This formal movement cemented the notion that only sentences that can be deduced from core axioms through a process of inference are considered a proof of a theorem:

> A theorem is only proved when the proof is completely independent of the diagram. The proof must call step by step on the preceding axioms. The making of figures is [equivalent to] the experimentation of the physicist, and experimental geometry is already over with [laying down of the] axioms. (Hilbert, 1894/2004)

Diagrams, in any form, have until the rise of symbolic logic not been formalized in this Hilbertian way: there were no axiomatic diagrams,[1] there were no formalized inference steps of diagrammatic manipulations, and moreover, mathematics was full of erroneous solutions due to faulty or deceptive diagrams.[2] Yet mathematicians continued to use diagrams to aid their reasoning and to convey the intuition of a solution.

The prejudice against the formal status of diagrams and visual methods of reasoning with them was finally overturned by Shin's seminal work on the formalization of Venn diagrams (Shin, 1994). Shin devised a formal syntax and semantics of particular forms of Venn diagrams, and proved her logic sound and complete. This legitimized and spurred on renewed interest in studying and devising systems for reasoning with diagrams, some of which will be surveyed here.

This chapter examines the role that diagrams play in human reasoning from the *computational*, *algorithmic*, and *implementational* points of view.[3] We argue that diagrams can encode the part of human reasoning that carries the intuition of the solution to a problem and can, moreover, convey the proof without needing to be supplemented by symbolic logic. Human rationality and reasoning are intimately related, as the reasoning process provides the rationale for a human understanding of, and belief in, the correctness of a solution. Our interest does not lie in machine-oriented approaches to reasoning, for which symbolic logic is typically used (with perhaps a small exception in the case of the motivation for natural deduction). Instead, we study diagrammatic reasoning and systems as one plausible way to model human-oriented approaches to rationales and therefore reasoning. Machine-oriented approaches to reasoning are typically motivated by the search for a categorical answer to the question of whether a conjecture is a theorem, by the speed of finding a proof, or by the number of proofs that the system is able to find. By contrast, human-oriented approaches to reasoning seek solution explanations that are understandable to humans. This human-oriented approach to modeling reasoning is motivated by the goal of general artificial intelligence (AI) to make machines more human-like in their reasoning and in the solutions that they find.[4] The hope is that diagrammatic reasoning systems will give us insights into human reasoning, which can then be modeled on machines to make them more human-like. As AI systems become ever more ubiquitous in our interactions with the world, it is important to build systems that humans find amenable and understandable. The new legal requirement for explainability makes the combination of sound reasoning and accessibility very timely. In part, this means capturing the human intuition and rationale of a solution in some way—the diagram presents not only a solution but also an explanation of why the solution is a correct one. We aim to demonstrate that this human-like reasoning with diagrams can be computationally modeled in logically formal ways on machines.

2. Representations

The importance of problem representation has been acknowledged by many researchers (Amarel, 1968; Simon, 1996). Pólya wisely pointed out in his books *How to Solve It* (1957) and *Mathematical Discovery* (1965) that reasoning about a problem depends on how one represents the problem in the first place. J. R. Anderson's (1978) mimicry argument adds the inseparability of the format of a representation and the processes that operate on this representation. This is even more immediately apparent when modeling reasoning with diagrams: representing the problem in the right way may lead to a trivial solution that captures the intuition and truthfulness in an obvious and accessible way. Representing the problem inadequately may prevent us from ever finding a solution. In contrast to machines, which typically use a single fixed representation for a problem, most humans are good at choosing the representation that is just right to enable them to solve a new problem. Moreover, given the right representation, human problem solving is dramatically improved, and humans recognize the benefits of such changes (Cheng, Lowe, & Scaife, 2001). Take, for example, the famous problem of the mutilated checkerboard in figure 13.1.1. It is called "mutilated" because two diagonal corners have been removed. The question is, Can you cover this mutilated checkerboard with dominoes?

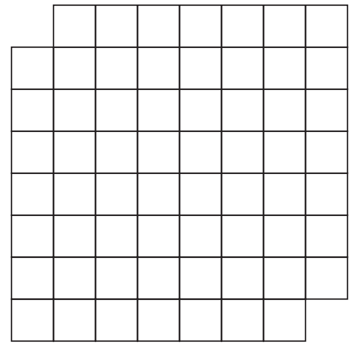


Figure 13.1.1
Mutilated checkerboard. Can it be covered with dominoes?

One could try many different ways to tile the board to find out if it can be covered. However, if we change the representation and color the board black and white like a chessboard (figure 13.1.2), then it is immediately obvious that the mutilated checkerboard cannot be covered with dominoes, since by mutilating it, we have removed two squares of the same color. But we need the same number of black and white squares since each domino covers one black and one white square. This is obvious for humans, but how would a machine go about finding this helpful representation?

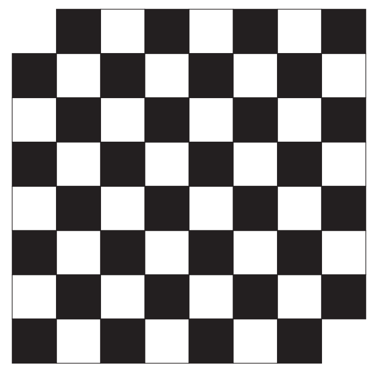


Figure 13.1.2
No, this mutilated checkerboard cannot be covered with dominoes, because it has more black than white squares, but there must be the same number of black and white squares since a domino covers one of each.

It is difficult to see how to use Pólya's advice about representations in computational reasoning systems. Unlike humans, machines' representations are usually fixed by their programmer, and while they can be from a wide range, they are typically a single symbolic representation. What machines don't currently do is reason about what system of representation to adopt. They deal in local parameters of whatever representation they are programmed in. Enabling a reasoning system to choose the right representation in problem solving is a big open challenge that has only recently started to be addressed (Raggi et al., 2020). Its importance lies in the relation between the problem representation and the computational complexity of finding its solution: imagine symbolically describing and solving the mutilated checkerboard example, and contrast this with the diagrammatic solution in figure 13.1.2. Larkin and Simon (1987) extensively discussed the importance of diagrams in many domains of reasoning: they claim that "a diagram is (sometimes) worth ten thousand words." Many so-called diagrammatic proofs without words can be found in Nelsen's books *Proofs without Words: Exercises in*

Visual Thinking (1993) and *Proofs without Words II: Exercises in Visual Thinking* (2001).

There is strong evidence from cognitive psychology that diagrams have many properties that often enable humans to process them with immediate comprehension. Johnson-Laird (1983) and Hegarty and Just (1993) argue that humans, at least in some cases, use diagrams in their mental models of a situation. It is important to recognize that the wrong level of granularity or too much visual clutter can impede cognitive processes (Alqadah, Stapleton, Howse, & Chapman, 2014; Knauff, 2013), but if diagrams capture the logical structure of the problem, they can be helpful in reasoning and problem solving (see chapter 13.3 by Knauff, this handbook). Diagrams concisely store information, explicitly represent the relations among the elements of the diagram, and support a lot of perceptual inferences that are very easy for humans. Shimojima (1996), for example, characterized a property of diagrams called "free rides": expressing some chosen information in a diagram often results in expressing other pieces of information that are consequences of the chosen information; informally, these other pieces of information are free rides. Figure 13.1.3b gives an example of how a Euler diagram representation of expressions in symbolic logic in figure 13.1.3a gives rise to a free ride about the fact that *R* and *Q* are disjoint: this can directly be read off from the diagrams in figure 13.1.3b, whereas it must be inferred from the symbolic sentences in figure 13.1.3a. Notions such as free rides give one possible explanation of why humans find diagrams so accessible.

Modeling this inherently human ability to choose or change appropriate representations in computational systems is an open research question. To address it, we need to find out which cognitive processes humans use to select representations, what criteria they use to choose them, and how we can model this ability on machines. The choice of representation, whether symbolic or diagrammatic (or even among a number of different symbolic or diagrammatic representations), will depend on whether the target audience is a human (and on their level of expertise) or a machine. Such a computational system could then readily assess the human and capitalize on the explanatory power of diagrammatic representations.

3. Logical Systems of Diagrams

Despite the persistent mistrust in the logical status of diagrams, the end of the 20th century started to see a redressing of this issue (M. Anderson, Meyer, & Olivier, 2001; Chandrasekaran, Glasgow, & Narayanan, 1995). Examples include formalized logical systems of diagrams (Hammer, 1995; Howse, Stapleton, & Taylor, 2005; Shin, 1994). This directly abolished the widely held Hilbertian theoretical objections to diagrams being used in proofs. Some of this seminal work is presented here.

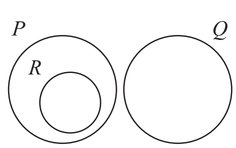
While Peirce (1933) began to formalize a variant of Venn diagrams and diagrammatic inference rules on them, it was Shin (1994) who was the first to show that two particular systems of Venn diagrams can have a formal syntax, semantics, and model theory. A *Venn diagram* (Venn, 1881) expresses all possible logical relations between a fixed finite number of sets. It does so by depicting sets as regions inside closed curves, which overlap in all possible ways (see figure 13.1.4). Venn diagrams use shading to represent empty sets.

Shin (1994) formalized a form of Venn diagrams, called "Venn-I," where she used shading to express set-emptiness (as in the original diagrams by Venn), x-sequences to express that a set is *not* empty, and linked x-sequences to express disjunction (similar to Peirce[5]). Shin's further Venn-II system introduced connecting lines between diagrams to express disjunctive information. Figure 13.1.5 shows statements in both Venn-I and Venn-II systems.

Importantly, Shin equipped both systems with formal syntax, semantics, as well as sound and complete inference rules and showed that Venn-II is expressively equivalent to monadic first-order logic. This was the first

$$(P \cap Q) = \varnothing$$
$$R \subseteq P$$

(a) Set-theoretic symbolic formulae.



(b) Euler diagram.

Figure 13.1.3
Free ride: in Euler diagrams, the fact that *R* is disjoint from *Q* can be read off, whereas this has to be inferred from the symbolic formulae.

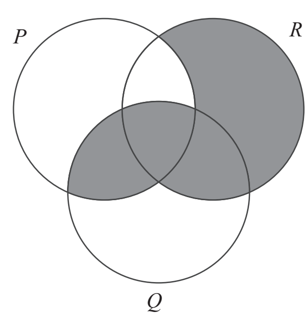


Figure 13.1.4
Venn diagram representation of the information in figure 13.1.3.

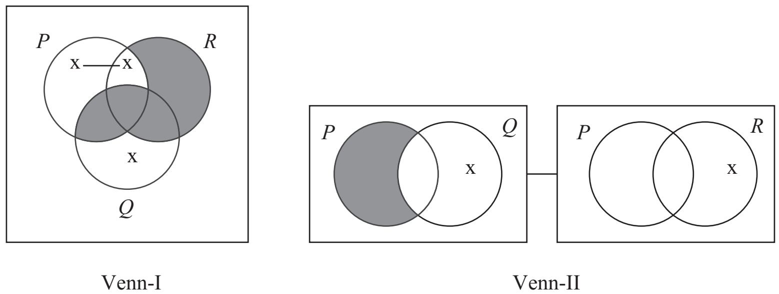


Figure 13.1.5
The Venn-I diagram expresses the information in figure 13.1.4 and also that there is at least one *Q* that is neither *P* nor *R* and there is at least one *P* that is not *Q*. The Venn-II diagram expresses that either *P* is a subset of *Q* and $Q \backslash P$ is not empty, or $R \backslash P$ is not empty.

time that a diagrammatic system for reasoning received a Hilbertian treatment. Shin showed that diagrams in a purely diagrammatic reasoning system can be used as formal tools.

Hammer (1995) extended Shin's work to *Euler diagrams*: he defined their formal syntax and semantics, and proved that a set of diagrammatic inferences were sound and complete. Euler diagrams (Euler, 1768) use closed curves that can overlap, be totally enclosed by other curves, or do not overlap at all, that is, exclude each other, to represent intersection, subset, and disjoint sets, respectively (see figure 13.1.3b).

Spider diagrams adapt Hammer's Euler and Shin's Venn-II diagrams. They are based on Euler diagrams, but instead of using x-elements to indicate nonempty regions, they use "spiders," that is, dots connected by lines, to represent the existence of distinct elements. Spider diagrams use shading: in a shaded region, all elements are denoted by spiders. So, a shaded region with no spiders represents the empty set (see figure 13.1.6).

The syntax and semantics of spider diagrams were formally defined in Howse et al. (2005). The language of spider diagrams is also accompanied by sound and complete inference rules, which result in the logic of spider diagrams. This logic is expressively equivalent to monadic first-order logic with equality (Stapleton, Howse, Taylor, & Thompson, 2004). A set of sound inference rules for spider diagrams was defined and proved complete in Urbas, Jamnik, and Stapleton (2015). Thus, spider diagrams are more expressive than Venn-II diagrams.

The revived interest in diagrams has seen other domains formalizing a rigorous syntax and semantics of diagrams for reasoning. Some are equipped with proved formal properties of soundness (which is a minimal requirement for using them in proofs) and some even with completeness. Examples include analogical reasoning in Grover (Barker-Plummer & Bailin, 1997), diagrammatic reasoning about arithmetic in Diamond (Jamnik,

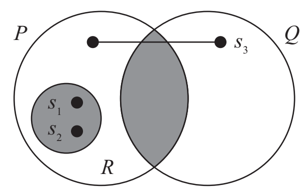


Figure 13.1.6
Spider diagram: *R* is a subset of *P* and has exactly two elements, s_1 and s_2; *Q* is disjoint from *P* (and *R*); element s_3 is either in *P* or in *Q*, but not in both.

2001), ontology reasoning with concept diagrams in iCon (Shams, Sato, Jamnik, & Stapleton, 2018), and computer science specification with UML diagrams based on higraphs (Rumbaugh, Jacobson, & Booch, 1999). Similar to the formalization of Venn, Euler, and spider diagrams, all of these systems provide evidence that diagrams can be used in formal proofs and thus show that diagrams can be made Hilbertian.

4. Implementations of Diagrammatic Reasoning Systems

In parallel with theoretical interest in the logic of diagrams revived at the end of the 20th century, several diagrammatic mechanized reasoning systems have been implemented. The motivations for their implementations differ: from cognitive ones about improved accessibility of reasoning systems to proof of concept that diagrams can be formally used for mechanized machine reasoning. They provide one model of reasoning that is more human-like than traditional symbolic machine-oriented approaches: they are both formal and explainable.

In implementing diagrammatic reasoning systems, there are many design choices that need to be carefully considered. For example, how do we devise algorithms for laying out diagrams in the best and clearest possible way, is the set of inference rules sound and complete, what level of abstraction of inference rules is most accessible to humans, what is the notion of a diagrammatic proof/solution, and should diagrams be general or should we exploit their concreteness? Here we demonstrate how some of these choices were made for three particular implementations of diagrammatic reasoning systems: Hyperproof, Diamond, and Speedith.

4.1 Hyperproof

Hyperproof (Barker-Plummer, Barwise, & Etchemendy, 2017; Barwise & Etchemendy, 1991) was the first system that implemented the use of both diagrammatic and symbolic representations to derive conclusions. The diagram is a blocks-world situation depicted in a graphical display: the upper part of Hyperproof's screen in figure 13.1.7.[6] In addition, some symbolic formulae of first-order logic might be given (in the lower part): *Dodec*(*c*) → *Dodec*(*d*) and *Small*(*c*). The aim is to show that some conjecture about the given information, usually expressed symbolically, is a consequence or a nonconsequence of the given information. In the example in figure 13.1.7, we want to determine whether block *c* and block *d* are of the same shape: *SameShape*(*c*,*d*). Hyperproof combines inspecting the diagram and applying logical rules in order to derive the conclusion.

The proofs in Hyperproof are guaranteed to be correct if every step, diagrammatic or symbolic, is evaluated to be true according to the evaluation schema in Kleene logic. They are heterogeneous since they combine symbolic and diagrammatic information. This notion of a heterogeneous proof is novel and stands in contrast to the Hilbertian view of formal proof. Using the blocks-world situation exploits not only diagrammatic representations but also their concreteness property—concrete proof objects can be named and their shape and size inspected. Hyperproof has been successfully deployed in an educational setting to teach first-order logic precisely for these properties: they make the reasoning more accessible to most users.[7]

4.2 Diamond

Diamond is the first system that implemented the construction of purely diagrammatic proofs (Jamnik, 2001). Its domain is natural number arithmetic for inductive theorems. Uniquely, Diamond exploits the concreteness property of diagrams to express a particular concrete instance of a theorem and its proof, rather than the general, universally quantified case. For example, figure 13.1.8 gives a diagrammatic proof for a theorem about the sum of odd natural numbers for a concrete instance $n = 6$. The proof cuts a square of size 6 into a sequence of so-called *ells*,[8] each representing a subsequent odd natural number since it consists of two edges, $2n$, but the joining vertex is counted twice, so $2n - 1$. In

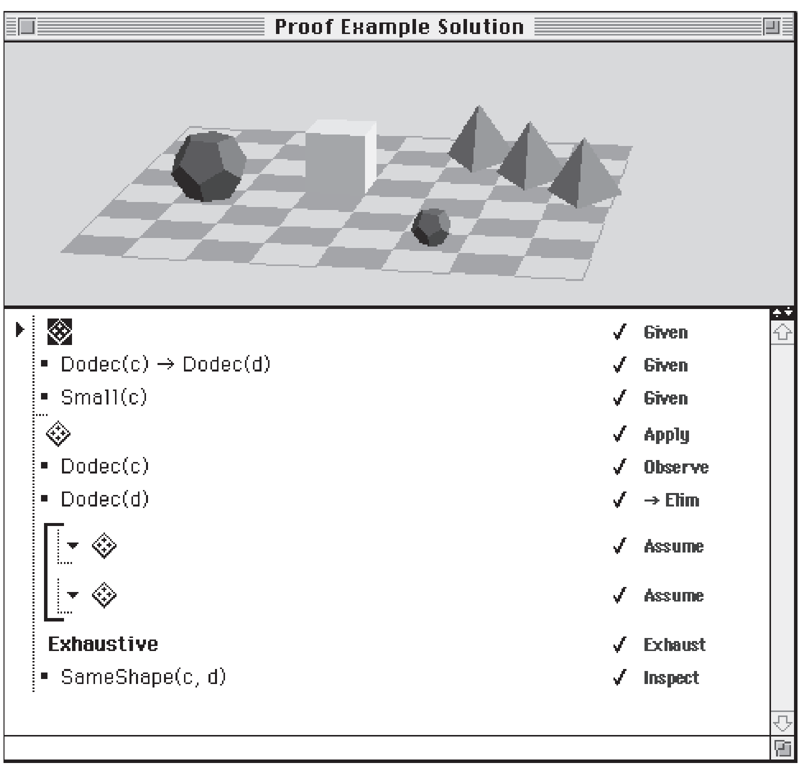
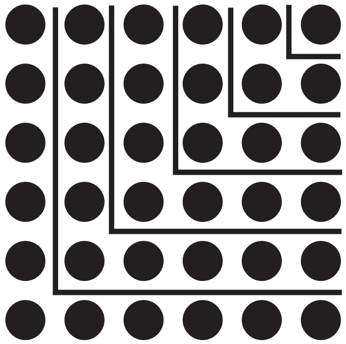


Figure 13.1.7
Hyperproof's proof.



$1 + 3 + 5 + \cdots + (2n - 1) = n^2$.

Figure 13.1.8
Diagrammatic proof of the theorem about the sum of odd natural numbers.

the concrete case of $n = 6$, the inference *lcut* was carried out five times.

Diamond's approach to capturing the generality of the proof while using concrete instances is theoretically justified by the constructive ω-rule (Baker, Ireland, & Smaill, 1992). This rule states that if there is a uniform effective procedure generating every instance of a proof, then we can conclude the universally quantified theorem. One such effective procedure is a recursive program. Therefore, Diamond's general proof is encoded in the recursive program that, upon an input for every value of the universal quantifier, produces a proof for that instance of the theorem.

The work on automation of diagrammatic proofs in Diamond provides important information on proof procedure construction. It demonstrates the importance of representing diagrammatic expressions so that general reasoning techniques can be applied to them. Furthermore, it provides insight into how diagrams and purely diagrammatic inferences can be used in formal proofs.

4.3 Speedith

Speedith (Urbas et al., 2015) is an interactive diagrammatic theorem prover for spider diagrams. It is the first implemented system with the same notion of proof as in a typical symbolic theorem prover: apply inference rules until tautology or contradiction is reached. Instead of symbolic formulae, the theorem is expressed using diagrams. The inference rules are diagrammatic (and some symbolic) and form a sound and complete set—this guarantees that diagrammatic proofs are correct.

Figure 13.1.9 shows an example of a diagrammatic proof as constructed in Speedith. Here, the antecedent of the *Initial proof goal* is a spider diagram that conveys information about the relationships between two elements and two sets, and it proves that the consequent of the *Initial proof goal* follows logically. In particular, the proof establishes that given sets A and B, if there are two elements, one of which is in both A and B and the other is either only in A or only in B, then we can deduce that one element is in A and the other is in B.

5. Discussion

The rationality of human thought has been studied from a number of different angles. Cognitive, psychological, and philosophical perspectives include investigations into mental models, heuristics, deduction, argumentation, beliefs, social epistemology, morality, and learning, among others. This chapter presented an angle stemming from the perhaps less mainstream, yet foundational, artificial intelligence view of one aspect of human rationality, namely, that of human-oriented visual mathematical reasoning. We approached rationality from the algorithmic and implementational view to computationally model reasoning with diagrams. The invention of Hilbertian reasoning at the start of the 20th century firmly rooted the formality of human thought in symbolic logic. This was despite centuries of mathematicians using visual tools such as diagrams to convey theorems and their proofs. The human ability to "see" the problem and its solution in a diagram is one of the fundamental components of the human mathematical cognitive repertoire. Nevertheless, there is a clear tension between symbolic and diagrammatic representations when computationally modeling human intuition in problem solving.

Unlike sentential, symbolic, or linguistic representations, which represent relations between objects in a single dimension by concatenation, diagrammatic representations use multidimensional relations between objects. This has both advantages and drawbacks. On the plus side, once the relations between objects are correctly represented, their properties or new relations can be just read off from the diagram. Indeed, diagrams may give you new information for free that was not originally declared and represented—referred to as "free rides." On the negative side, some relations cannot be expressed in diagrammatic representations by utilizing spatial properties, and if relations are not represented completely, this may lead to unintended and erroneous conclusions. Stenning and Oberlander (1995) argue that diagrams are necessarily concrete, with just a limited capacity for abstraction. Knauff discusses this tension between concreteness and abstraction, and the level of granularity in diagrams, in chapter 13.3 of this handbook. The

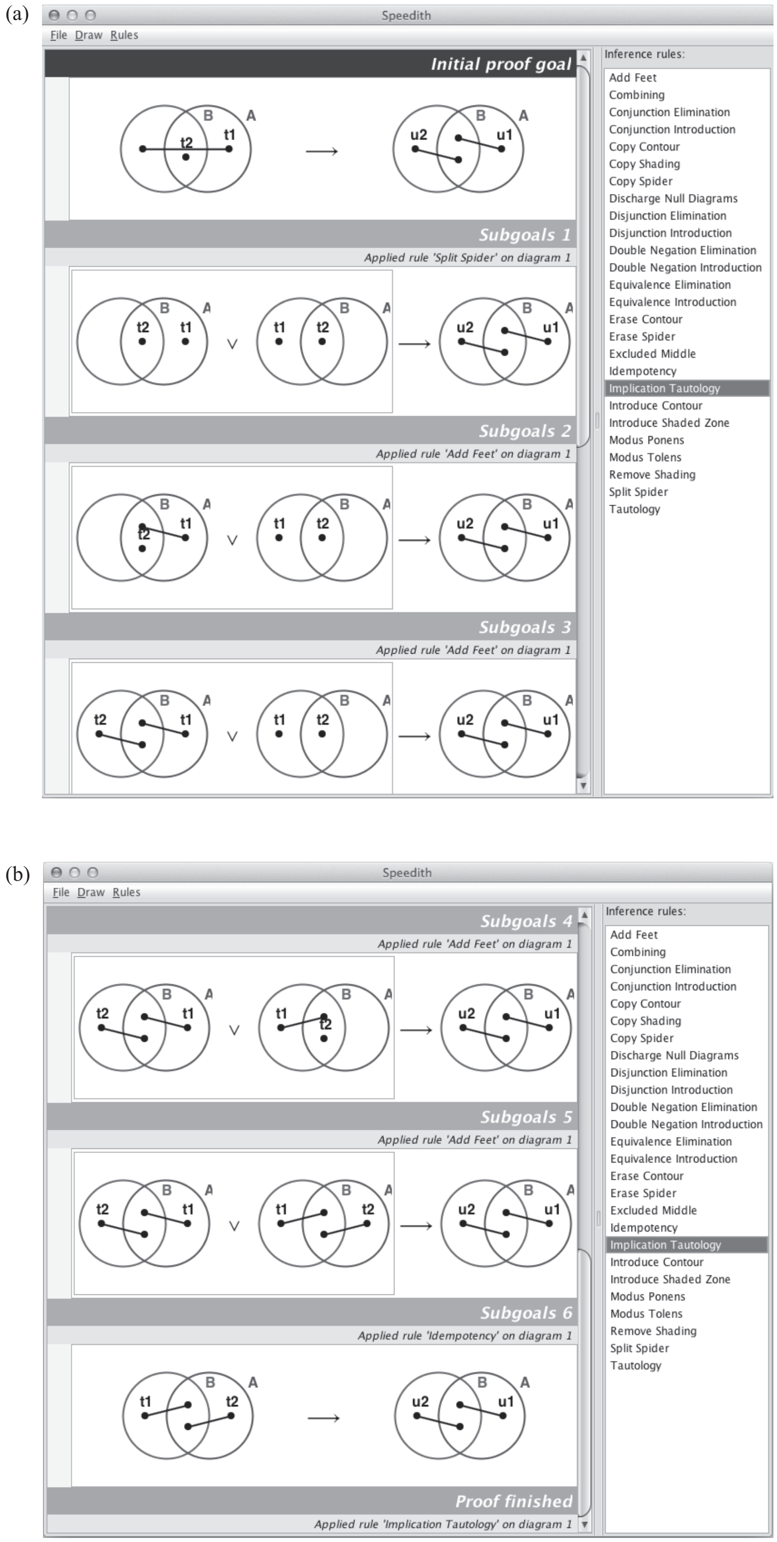


Figure 13.1.9
Screenshot of a proof of a spider-diagrammatic statement as constructed in Speedith.

consequence is that the user has to distinguish intended from unintended concreteness (e.g., the use of $n = 6$ above). Getting this wrong often causes erroneous conclusions. There are numerous examples in the history of mathematics of faulty solutions—Lakatos (1976) and Maxwell (1959) illustrate some better-known ones.

When are diagrammatic representations not expressive enough? The systems we mentioned here are all limited in expressivity in different ways. For example, Venn-I cannot express disjunctive information, spider diagrams can only express monadic first-order logic with equality, and Diamond can only tackle arithmetic theorems expressed as three-dimensional shapes. Diagrams are concrete objects and are thus subject to spatial constraints that sentential representations do not have. This is clearly limiting, and there are sentential logics, and their computational implementations in theorem provers, that are more expressive and indeed very impressive (e.g., Coq's proof of the famous four-color theorem; Gonthier, 2008).

However, purely sentential logics are limited in representational scope and often unilluminating with regard to solution explanations. The diagrammatic approaches to reasoning presented in this chapter provide a vehicle for widening the scope and expanding the explanatory power of computational systems for reasoning. They work with users in accessible and explanatory ways and can be formal in the Hilbertian sense. They enable us to explore the importance of representational choice on the diversity of inference methods and the ability to actually solve problems in mathematics. Ultimately, the goal is to understand human rationality and, from the AI point of view, to build computational systems that reflect it—this is going to make AI machines more human-like in internal workings and in external interactions with humans.

Notes

1. The distinction between symbolic and diagrammatic methods is not straightforward. While Newton's famous *Philosophiæ Naturalis Principia Mathematica* (*Mathematical Principles of Natural Philosophy*, 1726/2020) is resolutely based on geometric techniques, its contributions are little understood for their diagrammatic beauty—they are usually demonstrated and analyzed using a symbolic calculus into which they were translated very early on.

2. One of the best-known examples of erroneous proofs is Cauchy's visual proof of Euler's theorem, which is well documented in Lakatos's (1976) seminal work on the analysis of mathematical discovery. This "proof" stood for over 50 years before a Hilbertian symbolic formalization was devised that could readily be checked for errors.

3. The computational, algorithmic, and implementational views of rationality are defined in the Introduction to this volume by Knauff and Spohn. Here, our implementational view does not refer to the hardware that embodies rationality but rather to the software that executes reasoning tasks.

4. Current mainstream AI is mostly interested in machine-oriented statistical learning. Nevertheless, modeling human-like reasoning on machines remains one of the foundational motivations of the AI field.

5. Peirce did not use shading for set emptiness, but he introduced x (for nonemptiness) and o (for emptiness) elements that can be linked with lines to express disjunctive information in the diagrams. These diagrams, however, lost somewhat the spatial appeal and ease of reading that Venn diagrams have.

6. The diamond in the lower part of the display denotes the blocks-world situation—clicking the diamond displays the situation graphically in the upper part.

7. Stenning and Oberlander (1995) showed that there are individual differences in users' performance: users who are more visual reasoners do better with visual representations, and sentential reasoners do better with sentential representations.

8. Two perpendicularly joined sides of a square look like a letter L, so they are referred to as *ells*. Splitting an *ell* from a square is thus referred to as an *lcut*.

References

Alqadah, M., Stapleton, G., Howse, J., & Chapman, P. (2014). Evaluating the impact of clutter in Euler diagrams. In T. Dwyer, H. Purchase, & A. Delaney (Eds.), *Diagrams. LNCS 8578* (pp. 108–122). Berlin, Germany: Springer.

Amarel, S. (1968). On representations of problems of reasoning about actions. In D. Michie (Ed.), *Machine intelligence 3* (pp. 131–171). Edinburgh, Scotland: Edinburgh University Press.

Anderson, J. R. (1978). Arguments concerning representations for mental imagery. *Psychological Review*, *85*, 249–277.

Anderson, M., Meyer, B., & Olivier, P. (2001). *Diagrammatic representation and reasoning.* London, England: Springer.

Baker, S., Ireland, A., & Smaill, A. (1992). On the use of the constructive omega rule within automated deduction. In A. Voronkov (Ed.), *International Conference on Logic Programming and Automated Reasoning (LPAR). LNAI 624* (pp. 214–225). Berlin, Germany: Springer.

Barker-Plummer, D., & Bailin, S. (1997). The role of diagrams in mathematical proofs. *Machine Graphics and Vision*, *6*(1), 25–56.

Barker-Plummer, D., Barwise, J., & Etchemendy, J. (2017). *Logical reasoning with diagrams & sentences: Using Hyperproof*. Stanford, CA: CSLI.

Barwise, J., & Etchemendy, J. (1991). Visual information and valid reasoning. In W. Zimmerman & S. Cunningham (Eds.), *Visualization in teaching and learning mathematics* (pp. 9–24). Washington, DC: Mathematical Association of America.

Chandrasekaran, B., Glasgow, J., & Narayanan, N. H. (1995). *Diagrammatic reasoning: Cognitive and computational perspectives*. Cambridge, MA: AAAI Press/MIT Press.

Cheng, P. C.-H., Lowe, R., & Scaife, M. (2001). Cognitive science approaches to understanding diagrammatic representations. *Artificial Intelligence Review*, *15*, 79–94.

Euler, L. (1768). *Lettres à une Princesse d'Allemagne* [Letters to a German princess]. St. Petersburg, Russia: l'Academie Imperiale des Sciences.

Gonthier, G. (2008). Formal proof—the four-color theorem. *Notices of the American Mathematical Society*, *11*, 1382–1393.

Hammer, E. M. (1995). *Logic and visual information*. Stanford, CA: CSLI Press.

Hegarty, M., & Just, M. (1993). Constructing mental models of amchines from text and diagrams. *Journal of Memory and Language*, *32*, 717–742.

Hilbert, D. (2004). *Grundlagen der Geometrie*. In M. Hallet & U. Majer (Eds.), *David Hilbert's lectures on the foundations of geometry, 1891–1902* (pp. 65–144). Berlin, Germany: Springer. (Original work published 1894)

Howse, J., Stapleton, G., & Taylor, J. (2005). Spider diagrams. *LMS Journal of Computation and Mathematics*, *8*, 145–194.

Jamnik, M. (2001). *Mathematical reasoning with diagrams: From intuition to automation*. Stanford, CA: CSLI.

Johnson-Laird, P. (1983). *Mental models: Towards a cognitive science of language, inference and consciousness*. Cambridge, England: Cambridge University Press.

Knauff, M. (2013). *Space to reason—a spatial theory of human thought*. Cambridge, MA: MIT Press.

Lakatos, I. (1976). *Proofs and refutations: The logic of mathematical discovery*. Cambridge, England: Cambridge University Press.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*(1), 65–99.

Maxwell, E. A. (1959). *Fallacies in mathematics*. Cambridge, England: Cambridge University Press.

Nelsen, R. B. (1993). *Proofs without words: Exercises in visual thinking*. Washington, DC: Mathematical Association of America.

Nelsen, R. B. (2001). *Proofs without words II: Exercises in visual thinking*. Washington, DC: Mathematical Association of America.

Newton, I. (2021). *The mathematical principles of natural philosophy* (C. Leedham-Green, Ed., Trans.) (3rd ed.). Cambridge, England: Cambridge University Press. (Original work published 1726)

Peirce, C. S. (1933). *Collected papers*. Cambridge, MA: Harvard University Press.

Pólya, G. (1957). *How to solve it: A new aspect of mathematical method*. Princeton, NJ: Princeton University Press.

Pólya, G. (1965). *Mathematical discovery*. New York, NY: Wiley.

Raggi, D., Stapleton, G., Stockdill, A., Jamnik, M., Garcia Garcia, G., & Cheng, P. C.-H. (2020). How to (re)represent it? In *32nd IEEE international conference on tools with artificial intelligence (ICTAI)* (pp. 1224–1232). Baltimore, MD: IEEE.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language reference manual*. Reading, MA: Addison-Wesley.

Shams, Z., Sato, Y., Jamnik, M., & Stapleton, G. (2018). Accessible reasoning with diagrams: From cognition to automation. In P. Chapman, G. Stapleton, A. Mokfeti, S. Perez-Kriz, & F. Bellucci (Eds.), *Diagrammatic representation and inference (Diagrams). LNCS 10383* (pp. 247–263). Cham, Switzerland: Springer.

Shimojima, A. (1996). Operational constraints in diagrammatic reasoning. In G. Allwein & J. Barwise (Eds.), *Logical reasoning with diagrams* (pp. 27–48). New York, NY: Oxford University Press.

Shin, S.-J. (1994). *The logical status of diagrams*. Cambridge, England: Cambridge University Press.

Simon, H. A. (1996). *The sciences of the artificial*. Cambridge, MA: MIT Press.

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

Stenning, K., & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science*, *19*, 97–140.

Urbas, M., Jamnik, M., & Stapleton, G. (2015). Speedith: A reasoner for spider diagrams. *Journal of Logic, Language and Information*, *24*(4), 487–540.

Venn, J. (1881). *Symbolic logic*. London, England: Macmillan.

This is a section of doi:10.7551/mitpress/11252.001.0001

The Handbook of Rationality

Edited by: Markus Knauff, Wolfgang Spohn

Citation:

The Handbook of Rationality
Edited by: Markus Knauff, Wolfgang Spohn
DOI: 10.7551/mitpress/11252.001.0001
ISBN (electronic): 9780262366175
Publisher: The MIT Press
Published: 2021

Funding for the open access edition was provided by the MIT Libraries Open Monograph Fund.

The MIT Press

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Stone Serif and Stone Sans by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Knauff, Markus, editor. | Spohn, Wolfgang, editor.
Title: The handbook of rationality / edited by Markus Knauff and Wolfgang Spohn.
Description: Cambridge : The MIT Press, 2021. | Includes bibliographical references and index.
Identifiers: LCCN 2020048455 | ISBN 9780262045070 (hardcover)
Subjects: LCSH: Reasoning (Psychology) | Reason. | Cognitive psychology. | Logic. | Philosophy of mind.
Classification: LCC BF442 .H36 2021 | DDC 153.4/3—dc23
LC record available at https://lccn.loc.gov/2020048455