Informal Human Mathematical Reasoning

Mateja Jamnik
Computer Laboratory
University of Cambridge
www.cl.cam.ac.uk/users/mj201

We encounter mathematics in every aspect of our lives. Some of the deepest and greatest insights into reasoning were made in mathematics. Hence, it is not surprising that emulating such powerful reasoning on machines is one of the important and difficult aims of artificial intelligence and automated reasoning. Human mathematicians often use diagrams to better convey problems and generate intuitive and easily understandable solutions. They also often learn general solutions from examples of solutions to related problems. Sometimes, they may use analogy or symmetry in solving problems. My research is in the exploration of the nature of such informal reasoning.

Informal human reasoning is very powerful, yet its potential has largely not been exploited in the design of *mechanised reasoning systems* (i.e., systems which use some logic formalism to (semi-) automatically solve problems). This can perhaps be explained by the fact that we do not have a deep understanding of informal techniques and their use in problem solving. In order to advance further the state of the art of automated reasoning systems, I think it is important to integrate some of the informal human reasoning techniques with the proven successful formal techniques, such as different types of logic. This will not only make the reasoning systems more powerful, but such systems can then serve as tools with which we can study and explore the nature of human reasoning. My aim is to formalise and emulate, in particular, human reasoning with diagrams and human learning, on machines.

Theorems in automated theorem proving are usually proved with formal logical proofs, so called symbolic proofs. However, there is a subset of problems which humans can prove by the use of geometric operations on diagrams, so called diagrammatic proofs. Figure 1 presents an example of a diagrammatic proof[1] of a theorem concerning the sum of odd naturals $n^2 = 1 + 3 + 5 + \cdots + (2n - 1)$. The proof consists of repeatedly applying *lcuts* to a square (an *lcut* removes an *ell* shape which is formed from two adjacent sides of a square – see Figure 1).

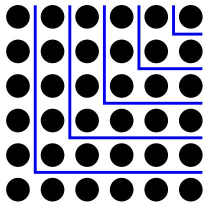

Figure 1: Figure 1: oddint.eps $n^2 = 1 + 3 + 5 + \cdots + (2n-1)$

Notice that an *ell* represents an odd natural number since both sides of a square of size n are joined ($2n$), but the joining vertex was counted twice (hence $2n-1$). We showed how such diagrammatic reasoning about mathematical theorems can be automated, and demonstrated the approach with the diagrammatic reasoning system called DIAMOND.[2]

In DIAMOND, concrete, rather than general diagrams are used to prove particular instances of a universal statement (e.g., in the example in Figure 1, the instance is $n = 6$). The ”inference steps” of a diagrammatic proof are formulated in terms of geometric operations on the diagram (e.g., the *lcut*s in the diagrammatic proof in Figure 1). A general schematic proof of the universal statement is induced from these proof instances by means of the constructive omega-rule. Schematic proofs are represented as recursive programs which, given a particular diagram, return the proof for that diagram. It is necessary to reason about this recursive program to show that it outputs a correct proof. One method of confirming that the abstraction of the schematic proof from the proof instances is sound is proving the correctness of schematic proofs in the meta-theory of diagrams.

DIAMOND can tackle only theorems which can be expressed as diagrams. However, there are theorems which may require a combination of symbolic and diagrammatic reasoning steps in the same proof attempt, so called heterogeneous proofs. I am currently investigating how a system could automatically reason about such proofs, and learn them in general from examples of proofs. An example below demonstrates a heterogeneous proof that consists of a combination of symbolic and diagrammatic inference steps. The theorem states an inequality: $\frac{a+b}{2} \geq \sqrt{ab}$ where $a, b \geq 0$. The first few symbolic steps of the proof are:

$$\begin{array}{rcl} \dfrac{a+b}{2} & \geq & \sqrt{ab} \\ & \downarrow & \text{square both sides of } \geq \\ \dfrac{(a+b)^2}{2^2} & \geq & ab \\ & \downarrow & \times 4 \text{ on both sides of } \geq \\ (a+b)^2 & \geq & 4ab \end{array}$$

$$
\begin{array}{rcl}
 & \downarrow & \\
a^2 + 2ab + b^2 & \geq & 4ab
\end{array}
$$

The second part of the proof[1], which is presented in Figure 2, shows diagrammatically the inequality $a^2 + 2ab + b^2 \geq 4ab$.

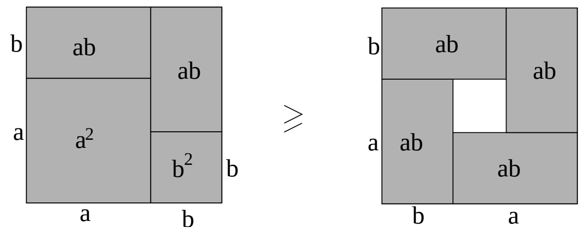

Figure 2: Figure 2: ineq.eps $a^2 + 2ab + b^2 \geq 4ab$

Rather than learning low-level proofs, I aim for a system that can learn diagrammatic *proof plans*. *Proof planning*[3] is an approach to theorem proving which uses high-level *proof methods* rather than low-level logical inference rules to find a proof of a conjecture at hand. It is a technique that searches for a proof plan, and a proof plan consists of some combination of proof methods. A *proof method* specifies and encodes a general reasoning strategy that can be used in a proof, and typically represents a number of individual inference rules (e.g., mathematical induction can be represented as a proof method). Our heterogeneous proof plans will be formed from geometric operations plus symbolic inference steps. The system will be able to learn such proof methods from examples of the use of lower-level methods, and eventually, it will be able to learn new diagrammatic and heterogeneous proof plans.

The hope is that ultimately learning new, general and complex proof methods and proof plans may lead to the discovery of new and interesting proofs of theorems of mathematics that use diagrams for inferencing.

1. R.B. Nelsen. **Proofs without Words: Exercises in Visual Thinking**. The Mathematical Association of America, 1993.

2. M. Jamnik. **Mathematical Reasoning with Diagrams: From Intuition to Automation**. CSLI Press, Stanford, 2001.

3. A. Bundy. *The use of explicit plans to guide inductive proofs.* In R. Lusk and R. Overbeek, editors, 9th Conference on Automated Deduction, pages 111–120. Springer Verlag, 1988.