THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Psychological validity of schematic proofs

Citation for published version:
Jamnik, M & Bundy, A 2005, Psychological validity of schematic proofs. in *Mechanizing Mathematical Reasoning: Essays in Honor of Joerg H. Siekmann on the Occasion of His 60th Birthday.* vol. LNCS 2605, Lecture Notes in Computer Science, vol. 2605, Springer-Verlag GmbH, pp. 321-341. https://doi.org/10.1007/978-3-540-32254-2_19

Digital Object Identifier (DOI):
10.1007/978-3-540-32254-2_19

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Mechanizing Mathematical Reasoning

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 18. Jan. 2022

Psychological Validity of Schematic Proofs

Mateja Jamnik[1] and Alan Bundy[2]

[1]University of Cambridge Computer Laboratory,
J.J. Thomson Avenue, Cambridge, CB3 0FD, England, UK
Mateja.Jamnik@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~mj201
[2]Centre for Intelligent Systems and their Applications, Division of Informatics,
University of Edinburgh, 80 South Bridge, Edinburgh, EH1 1HN, Scotland, UK
A.Bundy@ed.ac.uk
http://www.dai.ed.ac.uk/homes/bundy/

Abstract. Schematic proofs are functions which can produce a proof of a proposition for each value of their parameters. A schematic proof can be constructed by abstracting a general pattern of proof from several examples of a family of proofs. In this paper we examine several interesting aspects of the use of schematic proofs in mathematics. Furthermore, we pose several conjectures about the psychological validity of the use of schematic proofs in mathematics. These conjectures need testing, hence we propose an empirical study which would either support or refute our conjectures. Ultimately, we suggest that schematic proofs are worthy of a closer and more detailed study and investigation.

1 Introduction

In this paper we study and address several questions about the nature of mathematical proofs. How can a well chosen example often convey the idea of a proof better than the proof itself? How is it possible for proofs to be erroneous, and for such faulty "proofs" to persist for decades? Why are the proofs of some intermediate results less intuitive than the original theorem? We suggest that studying schematic proofs might provide some answers to such questions.

Schematic proofs have been used and studied in various branches of mathematics. Their use has been successfully mechanised in automated mathematical reasoning [1, 2]. We hypothesise that humans often use procedures similar to the construction of schematic proofs. The aim of this paper is to motivate cognitive scientists and cognitive psychologists that schematic proofs are an interesting concept in mathematics and that they are worthy of a closer investigation from a psychological point of view. Such an investigation would shed some light on the nature of human mathematical thought. We examine some interesting aspects of schematic proofs and postulate a number of conjectures about the psychological validity of schematic proofs. We have anecdotal evidence to support our intuitions, however, we have not conducted any systematic experiments. Hence, in §7, we propose an experimental investigation and we suggest some of the questions that such an investigation could attempt to answer.

Schematic proofs are functions, i.e., programs, which output a proof for each value of their parameters, i.e., inputs. That is, they are a way of capturing a family of proofs.[1] For example, consider a trivial theorem, let us call it *multiple addition*, which says that to get a value of an x in $(\ldots((x+a_1)+a_2)+\cdots+a_n) = y$ one has to subtract all the a_i from y. So, more formally, the theorem can be expressed as $((\ldots((x+a_1)+a_2)+\cdots+a_n) = y) \Rightarrow (x = (\ldots((y-a_n)-a_{n-1})-\cdots-a_2)-a_1)$. The schematic proof for this theorem is the following informal program (where we assume that we have definitions of *proof, apply*, etc.):

$$\mathit{proof}(n) = \mathit{apply}\ \ (U+V=W \Rightarrow U=W-V)\ \ n\ \mathit{times}$$

which rewrites n times, terms in the theorem of the form $U+V=W$ to terms of the form $U=W-V$. A schematic output of this program gives a proof of the *multiple addition* theorem (bold blocks represent program execution steps, i.e., applications of rewrite rules on the theorem):

$$
\begin{array}{c}
(((\ldots((x+a_1)+a_2)+\cdots)+a_{n-2})+a_{n-1})+a_n = y \\
\Downarrow \textbf{\textit{apply}}\quad (U+V=W \Rightarrow U=W-V) \\
((\ldots((x+a_1)+a_2)+\cdots)+a_{n-2})+a_{n-1} = y-a_n \\
\Downarrow \textbf{\textit{apply}}\quad (U+V=W \Rightarrow U=W-V) \\
(\ldots((x+a_1)+a_2)+\cdots)+a_{n-2} = (y-a_n)-a_{n-1} \\
\vdots \\
x = (\ldots((y-a_n)-a_{n-1})-\cdots-a_2)-a_1
\end{array}
$$

A procedure that can be used to construct schematic proofs is to prove some special cases of a proposition, extract a pattern from these proofs, and abstract this pattern into a general schematic proof. We give examples of proofs for special cases for the above theorem where $n=2$ and $n=3$. When the schematic proof is given an input 2, then the program is instantiated to $\mathit{proof}(2) = \mathit{apply}\ \ (U+V = W \Rightarrow U=W-V)\ \ 2\ \mathit{times}$. The output of this program is:

$$
\begin{array}{c}
(x+a_1)+a_2 = b \\
\Downarrow \textbf{\textit{apply}}\quad (U+V=W \Rightarrow U=W-V) \\
x+a_1 = b-a_2 \\
\Downarrow \textbf{\textit{apply}}\quad (U+V=W \Rightarrow U=W-V) \\
x = (b-a_2)-a_1
\end{array}
$$

Similarly, when the schematic proof is given an input 3, then the program is instantiated to $\mathit{proof}(3) = \mathit{apply}\ \ (U+V=W \Rightarrow U=W-V)\ \ 3\ \mathit{times}$. The

[1] Schematic proofs are often used as an alternative to mathematical induction (see §2).

output of this program is:

$$
\begin{array}{rl}
((x + a_1) + a_2) + a_3 = b & \\
& \Downarrow \textbf{\textit{apply}} \quad (U + V = W \Rightarrow U = W - V) \\
(x + a_1) + a_2 = b - a_3 & \\
& \Downarrow \textbf{\textit{apply}} \quad (U + V = W \Rightarrow U = W - V) \\
x + a_1 = (b - a_3) - a_2 & \\
& \Downarrow \textbf{\textit{apply}} \quad (U + V = W \Rightarrow U = W - V) \\
x = ((b - a_3) - a_2) - a_1 &
\end{array}
$$

Finally, the schematic proof needs to be shown to be correct, i.e., that *proof*(n) outputs a proof of the theorem for case n. This is discussed in §2. In §3 we give more examples of the use of schematic proofs in mathematics.

There are three particular aspects of schematic proofs that we investigate in some detail. First, we examine how schematic proofs can be constructed from examples of proofs. The mathematical foundation for the construction of schematic proofs provides a justification for the step from examples to general proofs to theorem-hood. So, in §4, our first conjecture is that:

> *Schematic proofs explain how examples can be used for constructing general proofs.*

Second, we examine how schematic proofs have been used in the past to represent claimed proofs of theorems. However, upon closer examination, it turned out in some cases that what was thought to be a proof, was actually faulty and not a proof at all. We argue that this may be due to the omission of the verification of the schematic proof. Hence, in §5, our second conjecture is that:

> *Schematic proofs account for some erroneous proofs in mathematics.*

We give some historical examples which support our conjecture.

Finally, schematic proofs of some theorems can be very different from their standard non-schematic inductive counterparts. They often seem to be more easily understood than inductive proofs. A number of examples are given to support our claim. Therefore, in §6, our third and final conjecture is that:

> *Schematic proofs are more intuitive than inductive proofs.*

1.1 Technical Terminology

Here we give some definitions of technical terms used in this paper that might prove useful. Notice that in the literature, the terms induction, abstraction and generalisation are often used interchangeably for the same concept. We have three different notions for these terms, and hence define them here precisely.

A Recursive function is a function whose definition appeals to itself without an infinite regression. For example, Hex is a recursive function which for each input natural number n gives the n^{th} hexagonal number:

$$\begin{aligned} Hex(0) &= 0 \\ Hex(1) &= 1 \\ Hex(n+1) &= Hex(n) + 6 \times n \end{aligned}$$

The Successor function is a function that adds one to its argument. For example, $s(s(0)) = s(1) = 2$.

Instantiation is a process of replacing a variable with some value. Instantiation of a function is a process of assigning values to the arguments of the function and evaluating the function for these values. For example, instantiating the above function Hex for 3 gives $Hex(3) = Hex(2+1) = Hex(2) + (6 \times 2) = (Hex(1) + (6 \times 1)) + 12 = 1 + 6 + 12 = 19$.

Abstraction is a process of extracting a general argument from its examples. In this paper it refers to constructing a schematic proof from example proofs. For example, the process of constructing $proof(n)$ for the *multiple addition* theorem given above from the examples of its proof for $n = 2$ and $n = 3$ is referred to as abstraction.

Another meaning of abstraction in this paper is to refer to an *abstraction device*, such as ellipsis (i.e., the "..." notation), to represent general diagrams. Abstraction is sometimes referred to as inductive inference, or "philosophical induction", or generalisation.

Generalisation replaces a formula by a more general one. For example, constants, functions or predicates can be replaced by variables (e.g., $x + 3 = y$ is generalised to $x + a = y$ where a constant 3 is replaced by a variable a), or universally quantified variables are decoupled (e.g., $\forall x.(x+x)+x = x+(x+x)$ is generalised to $\forall x \forall y \forall z.(x+y)+z = x+(y+z)$).

Object-level statement is a well-formed term, proof or inference step of the logic in use (cf. meta-level statement). For example, the proof of *multiple addition* theorem given above in §1 is an object-level statement.

Meta-level statement is a statement *about* an object-level statement, in some logical theory (cf. object-level statement). For example, a claim that the proof of *multiple addition* theorem given above in §1, is a correct proof of this theorem, is a meta-level statement about the proof of the *multiple addition* theorem.

Mathematical induction or standard induction is a rule of inference in some logical theory which is used to prove the statement that some proposition $P(n)$ is true for all values of $n > n_0$, where n_0 is some base value. This rule of inference makes an assertion about object-level statements (cf. meta-induction). For example, in Peano arthmetic, the rule of induction is:

$$\frac{P(0) \quad P(n) \rightarrow P(s(n))}{\forall n.P(n)}$$

Meta-induction is a rule of inference in some logical theory which is used to prove the meta-statement that some proposition $MP(n)$ about the object-level statement $P(m)$ is true for all values of $n > n_0$, where n_0 is some base value. This rule of inference makes an assertion about proofs rather than object-level statements (cf. mathematical induction). For example, in Peano arithmetic, the rule of meta-induction is (where *proof* is a recursive function, and ":" stands for "is a proof of"):

$$\frac{\mathit{proof}(0) : P(0) \quad \mathit{proof}(n) : P(n) \rightarrow \mathit{proof}(s(n)) : P(s(n))}{\forall n. \mathit{proof}(n) : P(n)}$$

Schematic is an adjective that refers to some general way of describing a class of objects. We use this adjective when describing a program that generates a proof for all instances of some corresponding theorem. We refer to these programs as *schematic proofs*. A formal definition of a schematic proof is given in §2 in Definition 3.

2 Schematic proofs

Our interest in schematic proofs comes from the perspective of automated reasoning, where the aim is to implement a system which constructs schematic proofs. The automation of proof extraction requires some suitable mechanism to capture a general proof. Schematic proofs provide such a mechanism. General schematic proofs can be constructed from a sequence of instances. A mathematical basis which justifies the step from specific examples to a general schematic proof is the constructive ω-rule [1]. ω is the name given to the infinite set $\{0, 1, 2, 3, \ldots\}$, or equivalently, using the successor function s (see §1.1), the set $\{0, s(0), s(s(0)), s(s(s(0))), \ldots\}$. Typically, a schematic proof is formalised as a recursive program. This recursive program allows us to conclude a general schematic proof for the universally quantified theorem. In this section, we formally define what a schematic proof is, and what is the mathematical basis for its formalisation.

The mathematical basis for extraction of schematic proofs is the constructive ω-rule. This rule is a version of the ω-rule [3]:

Definition 1 (ω-Rule).
The ω-rule allows inference of the sentence $\forall x.\ P(x)$ from an infinite sequence $P(n)$ for $n \in \omega$ of sentences

$$\frac{P(0), P(1), P(2), \ldots}{\forall n. P(n)}$$

Using the ω-rule, an infinite number of premisses needs to be proved in order to conclude a universal statement. This makes the ω-rule unusable for automation. Hence, we consider the constructive version of this rule [1]:

Definition 2 (Constructive ω-Rule).
The constructive ω-rule allows inference of the sentence $\forall x.\ P(x)$ from an infinite sequence $P(n)$ for $n \in \omega$ of sentences

$$\frac{P(0), P(1), P(2), \ldots}{\forall n. P(n)}$$

such that each premiss $P(n)$ is proved **uniformly** *(from parameter n).*

Note that the ω-rule and the constructive ω-rule are stronger alternatives for mathematical induction.

The uniformity criterion is taken to be the provision of a computable procedure describing the proof of $P(n)$, e.g., *proof*(n). The requirement for a computable procedure is equivalent to the notion that the proofs for all premisses are captured in a recursive function. We refer to such a recursive function as a *schematic proof.*

Definition 3 (Schematic Proof).
A schematic proof is a recursive function,[2] *e.g., $proof_P(n)$,*[3] *which outputs a proof of some proposition $P(n)$ given some n as input.*

Suppose the recursive function, *proof*, is a schematic proof. The function *proof* takes one argument, namely a parameter n. In general, this function can be defined to take any number of arguments. By instantiation, i.e., by assigning a particular value to n and passing it as an argument to the function *proof*, and by application of this instantiated function to the theorem, $proof_P(n)$ generates a proof for a particular premiss $P(n)$. More precisely, $proof_P(n)$ describes the inference steps (i.e., rules) made in proofs for each $P(n)$. Now, *proof*(n) is schematic in n, because we may apply some rule R a function of n (or a constant) number of times. That is, the number of times that a rule R is applied in the proof might depend on the parameter n. This recursive definition of a proof is used as a basis for implementation of the schematic proofs [2, 1].

From a practical point of view, the constructive ω-rule and schematic proofs eliminate the need for an infinite number of proofs, or in other words, they enable us to specify an infinite number of proofs in a finite way. Moreover, they provide a technique which enables an automation of search for proofs of universally quantified theorems from instances of proofs.

We now show how schematic proofs of universally quantified theorems can be found using several heuristics.

2.1 Finding a Schematic Proof

A schematic proof can be constructed by considering individual examples of proofs for instances of a theorem, and then extracting a general pattern from

[2] Technical terminology is explained in §1.1

[3] Note that we omit the use of subscript P in $proof_P(n)$ where it is clear which theorem *proof* proves.

these instances. The idea is that in order to extract a general structure common to all instances of a proof, the particular examples of proofs of a theorem which are considered, need to be general representatives of all instances, and not special cases. These are normally taken to be some intermediate values, e.g., 5 and 6, or 7 and 9, rather than the initial values, e.g., 0 and 1, since the proofs for initial values of a parameter n are almost always special cases. Therefore, we use such intermediate values, e.g., $P(7)$ and $P(9)$ and correspondingly $\mathit{proof}(7)$ and $\mathit{proof}(9)$, to extract the pattern, which we hope is general. A structure which is common to the considered examples is extracted by an abstraction. The result is the construction of a general schematic proof. If the instances for the intermediate values that were considered are not representative of all instances, so that the abstraction was carried out on incomplete information, then the constructed recursive function *proof* could be wrong. Therefore, the function *proof* needs to be verified as correct. This involves reasoning about the proof (using meta-level reasoning), and showing that *proof* indeed generates a correct proof of each $P(n)$.

The following procedure summarises the essence of using the constructive ω-rule in schematic proofs:

1. Prove a few particular cases (e.g., $P(7)$, $P(9)$, ... and thereby discover $\mathit{proof}(7)$, $\mathit{proof}(9)$, ...).
2. Abstract $\mathit{proof}(n)$ from these proofs (e.g., from $\mathit{proof}(7)$, $\mathit{proof}(9)$, ...).
3. Verify that $\mathit{proof}(n)$ proves $P(n)$ by meta-induction[4] on n.

The general pattern is abstracted from the individual proof instances by learning induction or abstraction. By meta-induction we mean that we introduce a theory META such that for all n the base case of the meta-induction is:

$$\text{META} \vdash \mathit{proof}(0) : P(0)$$

and the step case is:

$$\text{META} \quad \vdash \quad \mathit{proof}(n) : P(n) \longrightarrow \mathit{proof}(n+1) : P(n+1)$$

By meta-induction we need to show in the meta-theory that given a proposition $P(n)$, $\mathit{proof}(n)$ indeed proves it, i.e., it gives a correct proof with $P(n)$ as its conclusion, and axioms of some object logic as its premisses. This ensures that the constructed general schematic proof is indeed a correct proof for all instances of a proposition.

[4] The meta-induction is often much simpler than the mathematical induction that is alternative to the schematic proof. For example, whereas generalisation is required in some object-level inductive proofs, no generalisation is required in the meta-induction at the verification stage of the corresponding schematic proof. See §4 and §6 for more discussion and some examples.

3 Application of schematic proofs

To illustrate the use of the constructive ω-rule in schematic proofs, we give here five examples of schematic proofs for the following theorems: an arithmetic schematic proof of *associativity of addition* implemented by Baker [1], a schematic proof of *rotate-length theorem*, two diagrammatic schematic proofs, the first of the theorem regarding the *sum of odd naturals* implemented by Jamnik *et al* [2], and the second regarding the *sum of hexagonal numbers* presented by Penrose [4], and a faulty schematic proof of *Euler's* theorem presented by Lakatos in [5].

3.1 Associativity of Addition

Consider a theorem about the *associativity of addition*, stated as

$$(x + y) + z = x + (y + z)$$

Baker studied schematic proofs of such theorems in [1]. The recursive definition of "+" is given as follows:

$$0 + Y = Y \tag{1}$$
$$s(X) + Y = s(X + Y) \tag{2}$$

We also need a reflexive law $\forall n.\ n = n$.

The constructive ω-rule is used on x in the statement of the *associativity of addition*. We write any instance of x as $s^n(0)$. By $s^n(0)$ is meant the n-th numeral, i.e., the term formed by applying the successor function to 0 n times. Next, the axioms are used as rewrite rules from left to right, and substitution is carried out in the ω-proof, under the appropriate instantiation of variables. Hence, the following encoding:

$$\frac{\forall n.(s^n(0) + y) + z = s^n(0) + (y + z)}{\forall x. \quad (x + y) + z = x + (y + z)}$$

where n is the parameter, represents any instance of the constructive ω-rule in our example (note the use of ellipsis):

$$\frac{(0 + y) + z = 0 + (y + z),\ (s(0) + y) + z = s(0) + (y + z),\ (s(s(0)) + y) + z = s(s(0)) + (y + z),\ \ldots}{\forall x. \quad (x + y) + z = x + (y + z)}$$

We construct a schematic proof in terms of this parameter, where n in the antecedent captures the infinity of premisses actually present, one for each value of n. This removes the need to present an infinite number of proofs. The aim is to reduce both sides of the equation to the same term. The schematic proof of this theorem is the following program:

proof(n) = Apply rule (2) n times on each side of equality,
Apply rule (1) once on each side of equality,
Apply rule (2) n times on left side of equality,
Apply Reflexive Law

Running this program on the associativity theorem proves it. For example:

$$
\begin{array}{rcl}
(s^n(0) + y) + z & = & s^n(0) + (y + z) \\
& \Downarrow & \textbf{Apply rule (2) } n \textbf{ times on each side} \\
& \vdots & \\
s^n(0 + y) + z & = & s^n(0 + (y + z)) \\
& \Downarrow & \textbf{Apply rule (1) on each side} \\
s^n(y) + z & = & s^n(y + z) \\
& \Downarrow & \textbf{Apply rule (2) } n \textbf{ times on left} \\
& \vdots & \\
s^n(y + z) & = & s^n(y + z) \\
& \Downarrow & \textbf{Apply Reflexive Law} \\
& true &
\end{array}
$$

Note that the number of proof steps depends on n, which is the instance of x we are considering. We see that the proof is schematic in n — certain steps are carried out a number of times depending on n.

3.2 Rotate-Length Theorem

The *rotate-length* theorem is about rotating a list its length number of times, and can be stated as:

$$rotate(length(l), l) = l$$

where $length(l)$ gives the length of a list l, and $rotate(x, l)$ takes the first x elements of a list l and puts them at its end (e.g., $rotate(3, [a, b, c, d, e]) = [d, e, a, b, c]$), and can be defined as:

$$
\begin{array}{c}
rotate(0, l) = l \\
rotate(x, [\,]) = [\,] \\
rotate(n + 1, l :: ls) = rotate(n, ls@[l])
\end{array}
$$

Note that :: is infix cons (it takes an element and a list and puts the element at the front of the list, e.g., $1 :: [2, 3, 4] = [1, 2, 3, 4]$) and @ is infix append (it takes two lists and puts them together, e.g., $[1, 2, 3]@[4, 5] = [1, 2, 3, 4, 5]$). Consider a schematic proof of this theorem. First we give an example proof for some instance of a theorem. An example proof for the instance of a list of any five elements $l = [a, b, c, d, e]$, i.e., $length(l) = 5$ goes as follows. Let the list l consist of five elements. We take the first element of the list and put it to the back of the list. Now, we do the same for the remaining four elements.

$$\begin{aligned} rotate(length([a,b,c,d,e]),[a,b,c,d,e]) = \\ rotate(5,[a,b,c,d,e]) = \\ rotate(4,[b,c,d,e,a]) = \\ rotate(3,[c,d,e,a,b]) = \\ rotate(2,[d,e,a,b,c]) = \\ rotate(1,[e,a,b,c,d]) = [a,b,c,d,e] \end{aligned}$$

It is very easy to see that this process gives us back the original list. Moreover, it is clear that if we follow the same procedure, i.e., schematic proof, for a list of any length, we always get back the original list. Hence, the number of inference steps in the proof depends on n, so a proof is schematic in n:

$$\begin{aligned} rotate(length([a_1,a_2,a_3,\ldots,a_n]),[a_1,a_2,a_3,\ldots,a_n]) = \\ rotate(n,[a_1,a_2,a_3,\ldots,a_n]) = \\ rotate(n-1,[a_2,a_3,\ldots,a_n,a_1]) = \\ rotate(n-2,[a_3,\ldots,a_n,a_1,a_2]) = \\ \vdots \\ rotate(1,[a_n,a_1,a_2,a_3,\ldots]) = [a_1,a_2,a_3,\ldots,a_n] \end{aligned}$$

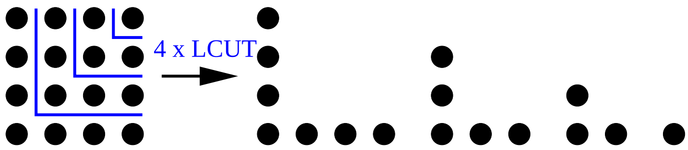
3.3 Sum of odd natural numbers

We now consider a theorem about the *sum of odd naturals* and its schematic proof as studied by Jamnik *et al* in [2] and [6]. Jamnik *et al* studied the notion of diagrammatic proofs and formalisation of diagrammatic reasoning. A diagrammatic proof is captured by a schematic proof that is constructed from examples of graphical manipulations of instances of a theorem. This diagrammatic schematic proof has to be checked for correctness. A diagrammatic proof consists of diagrammatic inference steps, rather than logical inference rules. Diagrammatic inference steps are the geometric operations applied to a diagram. The operations on diagrams produce new diagrams. Chains of diagrammatic inference rules, specified by the schematic proof, form the diagrammatic proof of a theorem. In Jamnik *et al*'s formalisation of diagrammatic reasoning, diagrams are used as an abstract representation of natural numbers, and are represented as collections of dots. Some examples of diagrams are a square, a triangle, an ell (two adjacent sides of a square). Some examples of geometric operations are *lcut* (split an ell from a square), *remove_row*, *remove_column*.

We demonstrate here a diagrammatic proof of the theorem about the *sum of odd natural numbers*. The theorem can be stated as

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

We consider an instance of the theorem $4^2 = 1+3+5+7$ and its diagrammatic proof where $n = 4$. Let us choose that n^2 is represented by a square of magnitude n, $(2n-1)$ is represented as an ell whose two sides are both n long, i.e., odd natural numbers are represented by ells, and a natural number 1 is represented as a dot. The proof of this instance of the theorem consists of cutting a square 4 times into ells.



Notice, that a similar procedure holds for a square of any size, i.e., for any instance of the theorem. Therefore, these steps are sufficient to transform a square of magnitude n representing the LHS of the theorem to n ells of increasing magnitudes representing the RHS of the theorem.

Note that the number of proof steps (i.e., diagrammatic inference steps) depends on n – for a square of size n the proof consists of n lcuts. Hence the proof is schematic in n. Here is a definition of this schematic proof:

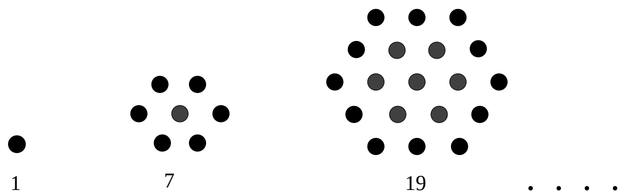
$$\begin{aligned} proof(n+1) &= apply\ lcut,\ then\ proof(n) \\ proof(0) &= empty \end{aligned}$$

3.4 Sum of hexagonal numbers

Let us now examine a theorem about the *sum of hexagonal numbers* and its (diagrammatic) schematic proof as presented by Penrose in [4]. We repeat here the formal recursive definition of hexagonal numbers from §1.1:

$$\begin{aligned} Hex(0) &= 0 \\ Hex(1) &= 1 \\ Hex(n+1) &= Hex(n) + 6 \times n \end{aligned}$$

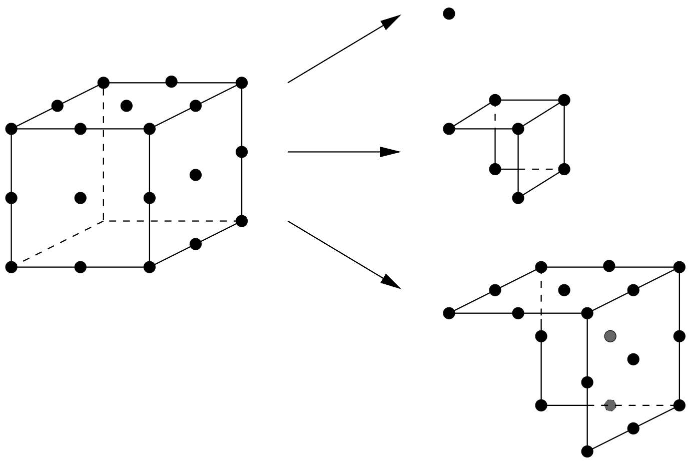
Informally, hexagonal numbers could be presented as hexagons where the hexagonal number is the number of dots in a hexagon:



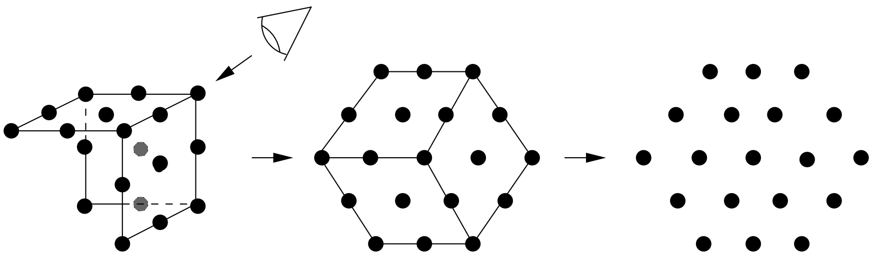
The theorem is stated as follows:

$$n^3 = Hex(1) + Hex(2) + \cdots + Hex(n)$$

Let n^3 be represented by a cube of magnitude n and $Hex(n)$ by an n^{th} hexagon. The instance of the proof that we consider here is for $n = 3$. The diagrammatic proof of the *sum of hexagonal numbers* consists of breaking a cube into a series of half-shells. A half-shell consists of three adjacent faces of a cube.



If each half-shell is projected onto a plane, that is, if we look at the top-right-back corner of each half-shell down the main diagonal of the cube from far enough, then a hexagon can be seen. So the cube is then presented as the sum of all half-shells, i.e., hexagonal numbers.



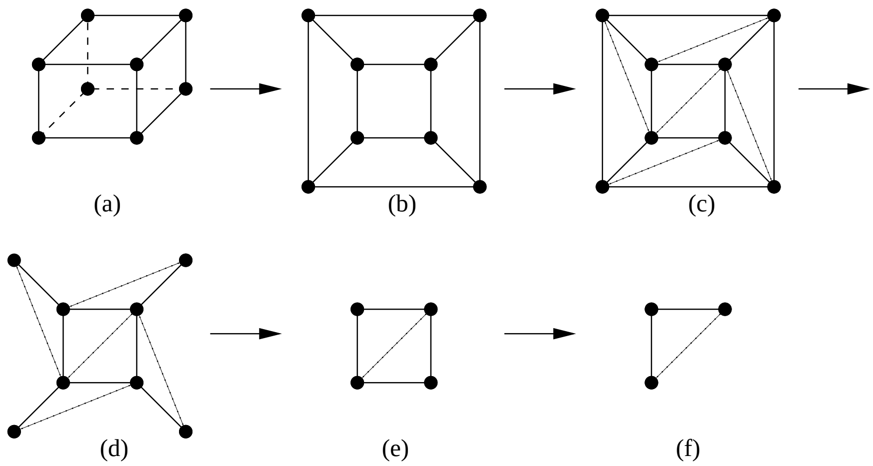
Again, notice that the general proof holds for any instance n. That is, these steps are sufficient to transform a cube of magnitude n representing the LHS of the theorem to n increasing hexagons representing the RHS of the theorem. Note that the number of diagrammatic inference steps depends on the value of n, so the proof is schematic in n.

3.5 Euler's Theorem

Let us consider a famous example of an erroneous schematic "proof", namely, the history of *Euler's theorem* [5]. *Euler's theorem* states that for any polyhedron $V - E + F = 2$ holds, where V is the number of vertices, E is the number of edges, and F is the number of faces. Lakatos[5] initially gives a proof, historically due to Cauchy, of the theorem, which is a uniform method for proving instances of *Euler's theorem*. Thus, the method is a schematic proof. However parts of the method are not explicitly stated, but seem very convincing when applied

[5] The proof of *Euler's theorem* is also discussed in [7, pages 47-48].

to simple polyhedra. Here is a summary of the proof method taken from [5, pages 7-8].[6]



Take any polyhedron (note that in our case, we take a cube, but the result is the same for any polyhedron). Imagine that it is hollow, and that its faces are made out of rubber (see (a) of the diagram above). Now, remove one face from the polyhedron, and stretch the rest of the polyhedron onto the plane (see (b) of the diagram). Note that since we have taken one face off, our formula should be $V - E + F = 1$. Note also that the relations between the vertices, edges and faces are preserved in this way. Triangulate all of the faces of this plane network (i.e., we are adding the same number of edges and faces to the network, so the formula remains the same — see (c) of the diagram). Now, start removing the boundary edges (see (d) of the diagram). This will have the effect of removing an edge and a face from the network at the same time, or two edges, one vertex and one face, so our formula is still preserved. We continue removing edges in appropriate order (see (e)), thus preserving the formula, until we are left with one triangle only. Clearly, for this triangle $V - E + F = 1$ holds, since there are three vertices, three edges and one face. Here is an informal diagrammatic schematic proof:

1. remove one face from any given polyhedron,
2. stretch the rest of the polyhedron onto the plane,
3. triangulate all of the faces that are not triangles already,
4. remove the boundary edges one after another, until you are left with a single triangle.

However, this schematic “proof” is faulty, and we will discuss the reasons for this in §5.

[6] The diagram demonstrating the proof of *Euler’s theorem* is also taken from [5, page 8].

4 Learning from examples

Schematic proofs and the constructive ω-rule explain why one or more examples can represent a general proof. Therefore, our first conjecture is that *schematic proofs explain the use of examples for construction of proofs.* Furthermore, we propose that reasoning with concrete cases, i.e., instances or examples, is often more easily understood than reasoning with abstract notions.

As described in §2, the constructive ω-rule enables us to capture infinitary concepts in a finite way. It enables us to use schematic proofs in order to prove universal statements. The constructive ω-rule gives us a mathematical basis which justifies how and why the examples or instances of problems can be used in order to conclude a general statement, in our case a general proof of a universally quantified theorem. We describe two systems which use schematic proofs, and hence reason with instances of theorems in order to prove universally quantified theorems, namely Baker's system CORE which reasons about theorems of arithmetic [1], and Jamnik's system DIAMOND which formalises diagrammatic reasoning [2].

Baker used schematic proofs in order to prove theorems of arithmetic, especially the ones which could not be proved by automated systems without the use of generalisation (for definition, see §1.1). One of Baker's example theorems is a special version of the theorem about *associativity of addition.* In §3.1 we gave a general version of this theorem. Baker's special version of the theorem can be stated as:

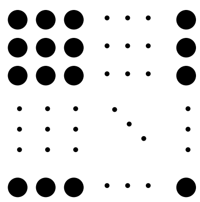
$$(x + x) + x = x + (x + x)$$

The CORE system automatically proves this theorem by enumerating instances of a proof, then constructing a general schematic proof, and finally, verifying that the schematic proof is correct. Instances of the theorem can be encoded as:

$$(s^n(0) + s^n(0)) + s^n(0) = s^n(0) + (s^n(0) + s^n(0))$$

for each parameter n. The schematic proof of this theorem is identical to the one in §3.1. In a theorem prover that cannot construct schematic proofs, this theorem would normally be proved by mathematical induction. But induction in this case is blocked, as $P(s(n))$ cannot be given in terms of $P(n)$ (for more details see [1]). Hence, generalisation to full associativity $(x+y)+z = x+(y+z)$ is necessary. Rather than using generalisation, as in other automated reasoning systems, CORE was able to prove this theorem using concrete instances of a theorem and its proof.

Jamnik uses schematic proofs for diagrammatic proofs of theorems of natural number arithmetic, like the theorem about the *sum of odd natural numbers* given in §3.3. To devise a general diagrammatic proof of this theorem, one would need to use abstract diagrams, i.e., diagrams of a general size. Therefore, diagrams would have to be represented using abstraction devices, such as ellipsis. Abstraction devices in diagrams are problematic as they are inherently ambiguous. The pattern on either end of the ellipsis needs to be induced by the system. For instance, it is ambiguous whether an abstract collection of rows or columns of dots with ellipsis, like this:



is a square or a rectangle, or if it is of odd or even magnitude. The problem becomes more acute when dealing with more complex structures. To recognise the pattern that the ellipsis represents, the system needs to carry out some sort of pattern recognition technique which deduces the most likely pattern and stores it in an exact internal representation. This guessed pattern might still be wrong. Because of the ambiguity of ellipsis it is difficult to keep track of it during manipulations of diagrams. Schematic proofs are a good way of avoiding this problem, as they allow us to use concrete instances of a theorem and its proof, and yet prove a general theorem. A procedure to construct a schematic proof in DIAMOND and CORE is to first prove instances of a theorem, e.g., a diagram, then construct a schematic proof, and finally prove that this schematic proof is correct. Using instances of a theorem enables us to use concrete diagrams in order to extract formal general proofs.

Besides the ability to extract general proofs from examples, it also appears that reasoning with examples seems easier for humans to understand than reasoning with abstract notions. The usual way in logic to prove Baker's theorem by a mechanised provers is to use mathematical induction and a generalisation, which is difficult to find for both, a human and an artificial mathematician – a mechanised mathematical reasoning system. Furthermore, another way of diagrammatically proving Jamnik's theorem is to reason with abstract diagrams which contain problematic ellipses. Using schematic proofs and instances of theorems seems an easier way to prove these theorems, and seems to convey better why the theorems hold.

5 Erroneous proofs

A generally accepted definition of a proof of a theorem in mathematical logic is the one given by Hilbert. Here is a translation of a quote from Hilbert's article [8].

> *"Let me still explain briefly just how a* **mathematical proof** *is formalized. As I said, certain formulas, which serve as building blocks for the formal edifice of mathematics, are called axioms. A mathematical proof is an array that must be given as such to our perceptual intuition; it consists of inferences according to the schema*

$$\frac{\mathcal{S} \quad \mathcal{S} \to \mathcal{T}}{\mathcal{T}}$$

> *where each of the premisses, that is, the formulas $\mathcal{S}$ and $\mathcal{S} \to \mathcal{T}$ in the array, either is an axiom or results from an axiom by substitution, or else coincides with the end formula of a previous inference or results from it by substitution. A formula is said to be provable if it is the end formula of a proof."* [9, pages 381-382]

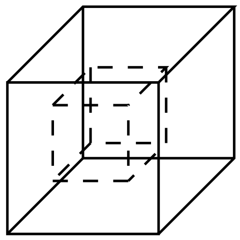
What Hilbert is talking about is sometimes referred to as Hilbert's Programme and is about the axiomatisation of mathematical systems. The definition of a proof in such a system can be summarised as follows. A proof of a theorem is a sequence of inference steps which are valid in some logical theory that has a complete axiomatisation, and which reduces a theorem that also belongs to this logical theory to a set of axioms, i.e., known true facts of the same logical theory.

However, this definition is questionable as it implies that the only explanation for errors in proofs is that they must be syntactic ones. Namely, Hilbert's argument suggests that all proofs boil down to a mechanical exercise of decomposing a theorem into a set of axioms of the theory to which they all belong. We suggest that syntactic errors could be automatically detected during this decomposition, and so erroneous proofs would not survive for years. In mathematics, people do not always formalise all axioms and inferences, yet their justifications for the truthfulness of theorems are generally accepted as correct proofs of theorems. For instance, consider Euclid's proofs of theorems of geometry long before a complete axiomatisation of geometry was given by Hilbert [10].

Mathematical proofs of theorems sometimes turn out to be faulty. The history of mathematics has taught us that there are plenty of faulty proofs of theorems which were for a long time considered to be correct, but later it turned out that the "proofs" were not proofs at all, that is, they were incorrect. Amongst famous examples is Cauchy's proof of the conjecture which says that the limit of any convergent series of continuous functions is itself continuous. Cauchy's "proof" persisted for almost forty years until the faulty conjecture was modified [5]. Another example is the *4-colour conjecture* which had faulty proofs [11]. An interesting discussion of this conjecture and its "proofs" is given in [12], and a correct proof of this theorem can be found in [13]. If Hilbert's definition of a proof was an accurate description of mathematical practice, then these erroneous "proofs" would not arise – any fault in the "proof" would be detected quickly as syntactic error. So what is going on, why do erroneous "proofs" persist?

Clearly, in mathematics in general Hilbert's definition of a proof holds only for a small part of mathematics, namely conjectures in logical theories which have complete axiomatisations. However, not all mathematical conjectures are part of known axiomatised logical theories.

Let us consider the famous example of an erroneous proof of *Euler's theorem*, given in §3.5. Analysing this proof, Lakatos [5] presents a number of counter examples in which the method of proof, i.e., the schematic proof, fails. It turns out that the initial theorem does not hold for *all* polyhedra. For example, it does not hold for hollow polyhedra, e.g., a solid cube with a cubical hole inside it, since $V - E + F = 4$. Note that the schematic proof fails at step 2.



The reader is referred to [5] for a number of counter examples of this theorem. One of the problems with Cauchy's schematic proof is that the definition of a polyhedron is not clearly stated. Therefore, a refinement of a theorem is needed. Lakatos's suggestion for this is to define a polyhedron as a surface and not as a solid. Lakatos proceeds to discuss other counter examples to Cauchy's schematic proof, and finally refines the definition of a polyhedron in a way that *Euler's theorem* does hold. It turns out that the theorems holds for all *simple*[7] [5, page 34] polyhedra whose faces are *simply connected* [8] [5, page 85].

Cauchy used a procedure for construction of schematic proofs in order to convince us of his "proof" of *Euler's theorem*. However, he did not carry out the last step of the procedure for extraction of schematic proofs, namely, he did not verify that the schematic proof is indeed correct.[9] We argue that if he did use the complete procedure, then the fallacy of the procedure would be detected at the verification stage. Note that this would require a constructive definition of a polyhedron.

It seems plausible that humans use some sort of schematic procedure to find general proofs of theorems. In particular, humans often use examples of proofs for certain instances and then abstract them into a general schematic proof. If not all the cases are covered by the examples, then the schematic proof might be incorrect, as in the case of the proof of *Euler's theorem* mentioned above. If a

[7] Simple polyhedra are ones which can be stretched onto the plane, i.e., those that are topologically equivalent to a sphere.

[8] A surface S is defined to be *connected* if any pair of its points can be joined by a continuous curve lying entirely within the surface. Further, a surface is said to be *simply connected* if any *closed* curve C on the surface divides the surface into *two* distinct regions, each of which is internally connected in the sense just described, and such that any continuous curve which joins a point in one of those regions to a point in the other must cross the closed curve C.

[9] A modern formal proof of *Euler's theorem* was devised only much later and is according to Lakatos [5, page 118] due to Poincaré [14]. It works by representing polyhedra as sets of vertices, edges and faces together with incidence matrices to say which vertices are in each edge and which edges are in each face. A restricted class of polyhedra is then turned into a formulae of vector algebra and a calculation in this algebra gives the value 2 for $V - E + F$. The proof is not intuitively clear, and it is not easy to see why the theorem holds and why this formal proof is correct.

counter example is encountered, then the method needs to be revised to exclude such cases. It seems that humans sometimes omit this step all together. Human machinery for extracting a general schematic argument is usually convincing enough to reassure them that the schematic argument is correct, e.g., consider the "proof" of *Euler's theorem.* Humans are happy with intuitive understandings of definitions and steps in the proof – as long as they do not encounter a counter example, their general pattern of reasoning in the proof is acceptable. Lakatos refers to such mathematical proofs as "thought experiments". It is only recently, in the 20th century, that thought experiments were replaced by logical proofs.

In an automated reasoning system, formality is of crucial importance. The correctness of the induced schematic argument has to be formally shown. This confirms that a schematic proof is indeed a correct formal proof of a theorem. If all proofs of theorems that people find followed rules of some formal logic, then there would be no explanation for how erroneous proofs could arise. The errors would always be detected as syntactical errors, provided that the rules used to prove the theorem are correct.

So, our second conjecture is that *human mathematicians often use a procedure similar to the construction of schematic proofs in order to find proofs of theorems, but they often omit the verification step which ensures that the proof is correct.* We propose further, that omitting the verification step of such procedure accounts for numerous examples of faulty "proofs". For instance, if one has not considered all the representative examples, then the schematic proof may not prove all cases of the theorem. A counter example may be found.

6 Intuitiveness of schematic proofs

Here, we extend the point in §4 that reasoning with examples or instances of a problem is easier than reasoning with abstract notions. We propose that *schematic proofs seem to correspond better to human intuitive proofs.* It appears easier to see why the theorem holds by looking at the instances of a theorem and its proof and then constructing a schematic proof, than considering a logical proof. As evidence, we give four examples of theorems from §3, where their schematic proofs are easier to understand than formal logical proofs: Baker's proof of *associativity of addition* from §3.1, Jamnik's diagrammatic proof of the *sum of odd naturals* from §3.3, Penrose's *sum of hexagonal numbers* from §3.4, and *rotate-length* theorem from §3.2.

We now consider further the *rotate-length* theorem. The informal schematic proof of this theorem is very easy to understand and to generalise to all cases of any list.

In contrast to a schematic proof of the *rotate-length* theorem, this theorem is not easy to prove by a conventional (non-diagrammatic) theorem prover. The inductive proof of the *rotate-length* theorem usually requires generalisation: e.g., $rotate(length(l), l@k) = k@l$, where @ is the list append function as defined in §3.2. It is harder to see that this theorem is correct. Schematic proofs avoid such generalisations. Baker used schematic proofs to exploit this fact for theorems of arithmetic [1].

We propose that the schematic proof given in §3.2 is a common way that people think about the proof of this theorem. Anecdotal evidence from humans suggests that schematic proofs are psychologically plausible. This supports our conjecture that schematic proofs correspond better to human intuitive proofs.

7 A proposed study

In this paper we proposed a number of conjectures about schematic proofs.

1. *Schematic proofs explain the use of examples for inducing formal proofs.*
2. *Schematic proofs account for erroneous proofs.*
3. *Schematic proofs are more intuitive than standard inductive proofs.*

These conjectures are not yet supported by an empirical study, but by our intuition and some suggestive examples. Hence, we propose an experimental study which could support or refute our intuitions. The study would look at some or all of the aspects of schematic proofs addressed in the previous sections. In particular, it would attempt to answer the following questions:

1. Do humans prefer to reason with concrete rather than general cases of a problem? Do humans use a procedure similar to the construction of schematic proofs when solving problems? If so, in what way do they use it and when?
2. Are there other examples which support the conjecture that incomplete schematic proofs account for some erroneous proofs?
3. Is reasoning with examples easier than reasoning with abstract notions? Are schematic proofs more easily understood than formal inductive proofs? If so, why do they appeal to humans more than formal inductive proofs?

The study proposed here would explore human intuitive reasoning in a novel way. We think that humans find schematic proofs easier to understand and more compelling than their logical counterparts. This is also part of the reason why humans might find diagrammatic proofs more intuitive than standard inductive proofs. We have only anecdotal evidence to support our belief. However, a comparative psychological validity experimental study could be carried out to answer some of the questions posed above and to provide some empirical evidence for or against our claims.

The proposed study could take the following form. An experiment could be carried out on a class of students with a certain level of mathematical knowledge (probably final year of secondary school level – the students should be equipped with the notion of mathematical induction). The class should be sufficiently large that the results are statistically significant. The students would be given examples of inductive theorems and non-theorems, and asked if they think the theorem is true or not. If they think it is true, the students would be asked to give an argument why they think it is true. Some of the non-theorems could be those which hold for the majority of cases, but are not true for some special and non-obvious cases. The students would also be asked to provide details of their problem solving process, i.e., the arguments that helped them reach a proof of a theorem or a conclusion that the theorems does not hold.

The data collected from the students would be analysed. Here are a few aspects that could be addressed in the analysis:

- classification of problem solving strategies using some existing techniques,
- analysis of whether the arguments used in the proof are inductive, schematic (using something like the constructive ω-rule), or some other type,
- analysis of the responses for non-theorems which are true for most cases, but not true for some more obscure special cases:
  - If the students realise that the conjecture is a non-theorem, how did they discover this (especially in the case of a schematic argument)?
  - If the students do not realise that the conjecture is a non-theorem, what are the arguments that falsely reassure them that the conjecture is a theorem and that it is true?

Another test that the students could be given consists of theorems and non-theorems, and their proofs and faulty "proofs" respectively. Each (non-) theorem could be accompanied with, say, three different (faulty) proofs each following a different strategy, e.g., inductive, schematic or other. In the case of non-theorems, the inductive argument would contain some syntactic errors and the schematic argument would not be verified for correctness. The students would be asked to choose the proof that is most convincing and that they think they understand best, and to elaborate on the reasons for their choice.

The questions which should be studied in more detail before the experiment is conducted include how much mathematical knowledge and knowledge of logic should the students have. Should they be trained in mathematical induction, constructive ω-rule, and other problem solving techniques? The danger is that people who have some training in mathematics, but not in logic would solve problems differently from those trained in logic, or those with little knowledge of mathematics and logic. Hence, the results would say less about the nature of proofs than about the abilities of individual students. A possibility is to separate subjects into two or more groups according to their level of training, and study the data according to these groups.

Here, we gave some preliminary suggestions for the design of the proposed experimental study. However, these ideas should be investigated in much greater detail before an experiment is conducted.

8 Conclusion

In this paper we posed several conjectures about the use of schematic proofs in mathematics. These conjectures make claims about the psychological validity of schematic proofs. First, we suggested that humans often use examples in order to conclude a general mathematical statement. Second, we conjectured that incomplete schematic proofs account for some erroneous proofs. Our suggestion is that looking at faulty proofs that have survived for years might give us useful insights into human reasoning. Finally, we conjectured that often schematic proofs are more intuitive than their inductive counterparts. These three conjectures are only supported by anecdotal evidence, so there is a clear need for a scientific experimental study which would test them. The motivation for this work is to investigate the nature of human mathematical thought and the notion of mathematical proof. Schematic proofs provide a good case study for such an investigation. Hence, our aim was to demonstrate that schematic proofs are worthy of a further study by cognitive scientists, and to propose the sort of questions that such an experiment could aim to answer. We hope that we provided enough evidence and motivation that the study of psychological validity of schematic proofs will be seen as a profitable scientific investigation, and will ultimately lead to further research and useful results.

References

1. Baker, S., Ireland, A., Smaill, A.: On the use of the constructive omega rule within automated deduction. In Voronkov, A., ed.: International Conference on Logic Programming and Automated Reasoning — LPAR-92. Number 624 in Lecture Notes in Artificial Intelligence, Berlin, Germany, Springer Verlag (1992) 214–225
2. Jamnik, M., Bundy, A., Green, I.: On automating diagrammatic proofs of arithmetic arguments. Journal of Logic, Language and Information **8** (1999) 297–321
3. Sundholm, B.: A Survey of the Omega Rule. PhD thesis, University of Oxford, Oxford, UK (1983)
4. Penrose, R.: Mathematical intelligence. In Khalfa, J., ed.: What is Intelligence?, Cambridge, UK, The Darwin College Lectures, Cambridge University Press (1994) 107–136
5. Lakatos, I.: Proofs and Refutations: The Logic of Mathematical Discovery. Cambridge University Press, Cambridge, UK (1976)
6. Jamnik, M.: Mathematical Reasoning with Diagrams: From Intuition to Automation. CSLI Press, Stanford, CA (2001)
7. Gamow, G.: One two three ... infinity: Facts & Speculations of Science. Dover Publications, New York (1988)
8. Hilbert, D.: Über das Unendliche. Mathematische Annalen **95** (1926) 161–190
9. Van Heijenoort, J., ed.: From Frege to Gödel – A Source Book in Mathematical Logic, 1879-1931. Havard University Press, Harvard, MA (1967)
10. Hilbert, D.: Grundlagen der Geometrie. Teubner, Stuttgart, Germany (1899) English translation 'Foundations of Geometry' published in 1902 by Open Court, Chicago.
11. Kempe, A.: On the geographical problem of the four colours. American Journal of Mathematics **2** (1879) 193–200
12. Mackenzie, D.: Mechanizing Proof. MIT Press, Boston, MA (2001)
13. Appel, K., Haken, W.: Every planar map is four colorable. Bulletin of the American Mathematical Society **82** (1976) 711–712
14. Poincaré, H.: Complément à l'analysis situs. Rendiconti del Circolo Matematico di Palermo **13** (1899) 285–343