Argument Mapping for Mathematics in Proofscape

Steve Kieffer

Caulfield School of Information Technology,
Monash University, Caulfield, Victoria 3145, Australia,
`Steve.Kieffer@monash.edu`

Abstract. The Proofscape argument mapping system for mathematical proofs is introduced. Proofscape supports argument mapping for informal proofs of the kind used by working mathematicians, and its purpose is to aid in the comprehension of existing proofs in the mathematical literature. It supports the provision of further clarification for large inference steps, which is available on demand when a proof is explored interactively through the Proofscape browser, and theory-wide exploration is possible by expanding and collapsing cited lemmas and theorems interactively. We examine how an argument map makes the structure of a proof immediately clear, and facilitates switching attention between the detailed level and the big picture. Proofscape is at `http://proofscape.org`.

Keywords: argument mapping, informal proofs, theory exploration

1 Introduction

Proofs in the mathematical literature use words like "therefore" and "hence" to introduce assertions, but such language fails to indicate from precisely which prior statements the new one is meant to follow. In other words, the *inferential structure* of the proof is not made explicit. Moreover, proofs may explain too little, leaving the reader to fill in difficult gaps, or explain too much, dissipating attention and obscuring the overview. For these reasons prose proofs can be difficult to understand.

In this paper I present Proofscape, an *argument mapping system* designed specifically for mathematics. Each assertion, assumption, and definition made in a literature proof is placed on a node in a Proofscape diagram, and the nodes are linked together with arrows indicating the inferential structure of the proof unambiguously. In fact the diagrams are not mere graphs but *DASHgraphs*, which I define in Sec. 3, and examples of which are seen in Fig. 2.

The overview provided by the diagram counteracts the problem of too much detail, or obscuration of the general plan, in prose proofs. As for the opposite problem of not enough detail, or difficult inferences, this is counteracted by supplementary clarifications, written by and for users of the system. These clarifications can be interactively opened and closed, meaning that extra nodes and edges are added to or taken away from the diagram. Theory-wide interactive exploration is also supported by allowing users to open and close cited lemmas and theorems.

Proofs are represented internally using a simple system I have designed for transcribing what happens in informal proofs, but proper discussion of that system requires a separate paper contrasting it to related proposals e.g. Wiedijk's [1]. Nor is there room here to discuss the layout algorithms that are used. On the contrary, the focus in the present paper is on *the design of the visual encoding* (Sec. 3), and on the demonstration of *how Proofscape diagrams support study of proofs* (Sec. 4). I review related work in Sec. 2. The main contributions of this paper are the visual notation, and the working Proofscape system at `http://proofscape.org`.

2 Related Work

The problems of obscure inferential structure and lack of modularity – i.e. the ability to expand and collapse levels of detail – in prose proofs were confronted by Lamport [2], who proposed presenting proofs in the form of indented outlines whose levels could be opened and closed, and in which claims cited previous lines by number. Lamport envisioned implementation in hypertext, which was later realised by Cairns and Gow [3], but this system did not use the diagrammatic nodes-and-arrows style of argument mapping systems. An earlier related proposal by Leron [4] used nodes and arrows to diagram not the steps of a proof but the relations between the modular levels into which it was to be decomposed.

The mapping of "what follows from what" is essentially the aim of all argument mapping systems (see e.g. Harrell's survey in [5]), but the structure and visual encoding of the diagram may depend on the intended subject matter. In particular, most existing argument mapping systems seem to be intended for use with subjects like law, philosophy, business, or politics (e.g. Gordon and Walton [6] or van Gelder [7]), and accordingly rely on the Toulmin model [8] or variants thereof. The Toulmin model is adapted for the questionable nature of claims regarding the real world, accommodating things like rebuttals, and citations of factual evidence, but none of this is appropriate for mathematics, where claims are either true or false, never debatable, and factual information about the real world is irrelevant. Moreover, systems like these omit many features that *are* needed for mathematical arguments, which I discuss in Sec. 3.

Some existing software does generate diagrams of mathematical proofs, but these seem almost universally to be proofs that have been encoded in some formal logical system. Here existing software is of two kinds: (1) interfaces for semi-automated theorem provers or formal proof assistants (e.g. Siekmann et. al. [9]); or (2) educational software intended to teach the inference rules of basic formal logics (e.g. Watabe and Miyazaki [10]). Systems of the former kind may present proofs at a high level, but still in a formal system, and their visual representations often represent progress in the construction of a proof, not a complete literature proof as it stands. Meanwhile the latter sort of system is concerned with the low-level atomic steps sanctioned by a rudimentary formal logic.

The *e-Proofs* system [11] guides students through informal textbook-style proofs, but does so using pre-generated instructional videos. In one of its three modalities the system shows a proof written out as a paragraph of text, and in successive frames draws boxes around various parts of the proof and arrows linking these, but these disappear in the next frame, and a graph is not built up (nor would the graph be laid out well if it did appear, its arrangement dictated by the line-breaks in the prose). The system is not meant for theory-wide exploration, each video being dedicated to a single proof, nor can the user interactively explore and hide supplementary clarifications for inference steps.

Proofscape thus fills a need for an argument mapping system designed specifically for informal mathematical proofs, with interactive access to further clarification, and theory-wide exploration.

3 Visual Encoding

In Proofscape, a proof is represented by a directed, acyclic, styled, hierarchical graph, or *DASHgraph*. The directed edges of the graph may represent both *deduction* and *flow* (see below). The graph is acyclic because deduction never involves circular logic. It is said to be styled because additional information is carried by the shape and stroke style (plain, bold, or dashed) of node boundaries, as well as the stroke style of edges. Finally, a DASHgraph is hierarchical in that some of its nodes may be nested inside of others, in order to represent subproofs and to contain cited theorems and their proofs.

The nodes carry the content of a proof, such as definitions, assumptions, and conclusions. Edges drawn with solid stroke are called *deduction arrows*, and indicate which assertions follow from which others; specifically, when the transcriber believes that $A_1, A_2, ..., A_n$ are the nodes which are, in an informal sense, "used" in inferring node B, then there is to be a deduction arrow from each A_i to B. Meanwhile edges drawn with dashed stroke are called *flow arrows*, and indicate the order in which the author of the proof would like you to explore the graph. This helps to ensure that the argument is built up in a sensible sequence, e.g. so that definitions are encountered before they are used.

There are five *basic* node types, two *compound* types, and several "special nodes" of which just two are considered here (see Fig. 1). The bulk of the content of a proof is carried by the first three basic node types: *Intro* nodes, *Assumption* nodes, and *Assertion* nodes.

Intro nodes are for the introduction of objects and symbols in the proof. For example, the label could state "Let K be a number field," or "Let G be the Galois group of E over K." Their boundary is rectangular and drawn with a bold stroke.

Assumption nodes are for the introduction of the premises of a theorem, and for additional assumptions made during the course of a proof, such as at the beginning of a proof by contradiction, or proof by cases. Thus the labels state suppositions like "Suppose G is Abelian," or "Suppose φ is surjective." The boundary of an assumption node is rectangular with a dashed stroke.

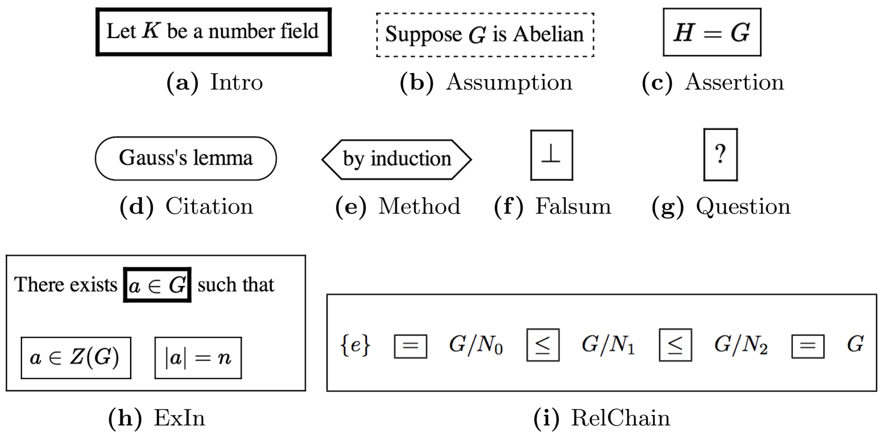


Fig. 1: Basic node types: (a),(b),(c),(d),(e); Special nodes: (f),(g); Compound node types: (h),(i)

There is a difference between the premises of a theorem and any additional assumptions introduced during the course of a proof, in that the latter will be discharged by the end of the proof whereas the former will not; in order to make this difference obvious to the reader, Assumption nodes belonging to top-level deductions are drawn both dashed and bold.

Assertion nodes are the most common type of node, carrying the assertions of a proof. Since it is up to the deduction arrows in the DASHgraph to indicate which assertions follow from which others, all logical language such as "therefore," "hence," "it follows that," etc. is omitted from the node labels. For example a label would state simply "$H = G$," never "therefore $H = G$." Assertion nodes have a rectangular boundary with plain stroke.

The node styles described so far have been chosen deliberately. It was felt that a dashed boundary is a natural metaphor for the contingent nature of assumptions, so is suitable for Assumption nodes in general, while the addition of a bold stroke suggests the somewhat more permanent nature of the premises of top-level deductions. The simple bold boundary of Intro nodes reflects the introduction of objects and symbols by fiat. Meanwhile Assertion nodes are the most common, and so should have a plain style, while the special styles of Assumption and Intro nodes help the reader to locate them quickly.

The final two basic node types are *Citation* nodes, which can be used *internally* to cite other results in the Proofscape library, and *externally* to cite any result in the literature; and *Method* nodes, carrying phrases like, "by induction," or, "substituting c for x," which clarify *how* an inference is made. Citation and Method nodes are given "stadium" and hexagonal shapes, respectively, which were chosen because they can easily stretch horizontally. The boundaries of Citation nodes are drawn dashed and solid to indicate internal and external citations respectively.

The first compound node type is the *Existential Introduction* or *ExIn* node, which handles formulas like, "There exists $a \in G$ such that $a \in Z(G)$ and $|a| = n$," in which an object is both stated to exist and introduced for further use. It features internal Intro and Assertion nodes for the components of the statement. The other compound node type is the *Relation Chain* or *RelChain* node which handles expressions of the form, "$A_1 = A_2 = \cdots = A_n$," by putting each infix relation $=$, $\leq$, $\geq$, etc. on its own internal Assertion node. Arrows to and from internal nodes may cut across hierarchy levels in the DASHgraph.

Among special nodes is the *Falsum* node, which is a special Assertion node featuring only the falsum symbol $\perp$, and which is to be used at the end of any proof by contradiction; and the *Question* node, featuring only a question mark, for use when the transcriber is not sure how an inference is to be made, thus to mark a point to return for further study.

4 Use Case and Conclusions

We conclude with an illustration of the use of Proofscape. Suppose you wanted to understand Mihăilescu's proof of the Catalan Conjecture [12]. You begin by locating the work in the Proofscape library and opening the main result, "Theorem 5," in the Proofscape browser. Initially only the theorem statement is shown. You click to open the proof. See Fig. 2a. The first thing you notice is that it is a proof by contradiction, with a Falsum node pointing to the final conclusion. You see the contradiction assumption in the node with dashed boundary in the upper left. Tracing backward along the deduction arrows that point to the Falsum node you see that the contradiction is between an assertion that $p > q$ which appears to follow relatively easily from the assumption, and another, $q > p$, which seems to require the bulk of the work in the proof.

From the external Citation node at the top right you see exactly which step in the argument relies on Baker's methods for linear forms in logarithms. Meanwhile the internal Citation nodes refer to three results from the present work, on which the main line of reasoning leading to $q > p$ depends. Suppose you decide to delve into Proposition 1 first, by selecting it in the sidebar. The Citation node is then replaced by the full statement of Prop. 1. After reading the statement you click again to open its proof. See Fig. 2b.

You might begin by tracing backward from the conclusion to see where it came from, and see that we get $q > p$ from $q - 2 \geq p - 1$. Expressions like $q - 2$ or $p - 1$ tend to arise because they represent some meaningful quantity, and you continue tracing backward to see what these might have been. On the far left you see that $q - 2$ is an upper bound on the number of zeroes of a certain polynomial F, whose definition is easily located in the Intro node above this. On the other hand there is a longer line of reasoning that shows that this polynomial F has at least $p - 1$ zeroes. The polynomial F is thus a *pivot* in the sense of Leron [4], a constructed object on whose properties the proof hinges.

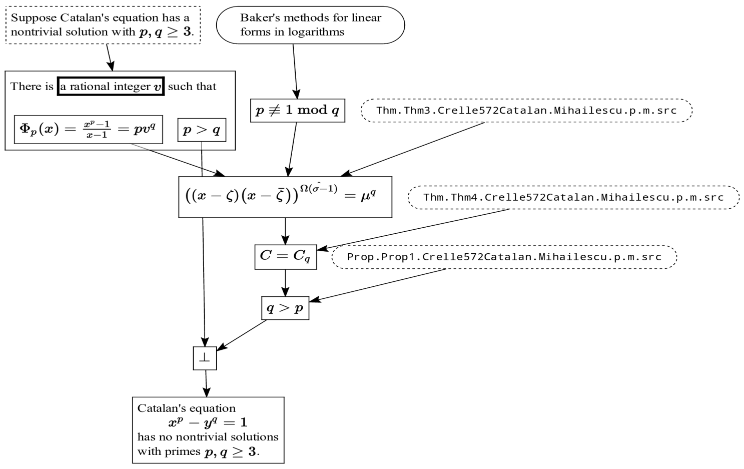
Next you may wish to see where the one premise of the Proposition is used. You easily locate it at the top right, in the node with the bold and dashed boundary, and can see that it contributes to a long line of reasoning which finally helps to get the $p-1$ zeroes of F.

You have learned all this about the structure of the argument without yet having spent any time working through difficult inferences. At this point you have a choice: If you want to understand the proof of Proposition 1 thoroughly you should begin to study each of the inferences one by one. As you go you may find user-supplied clarifications. On the other hand, if you are satisfied with this overview then you may close the proof by clicking again in the sidebar, and return to your study of the main theorem.

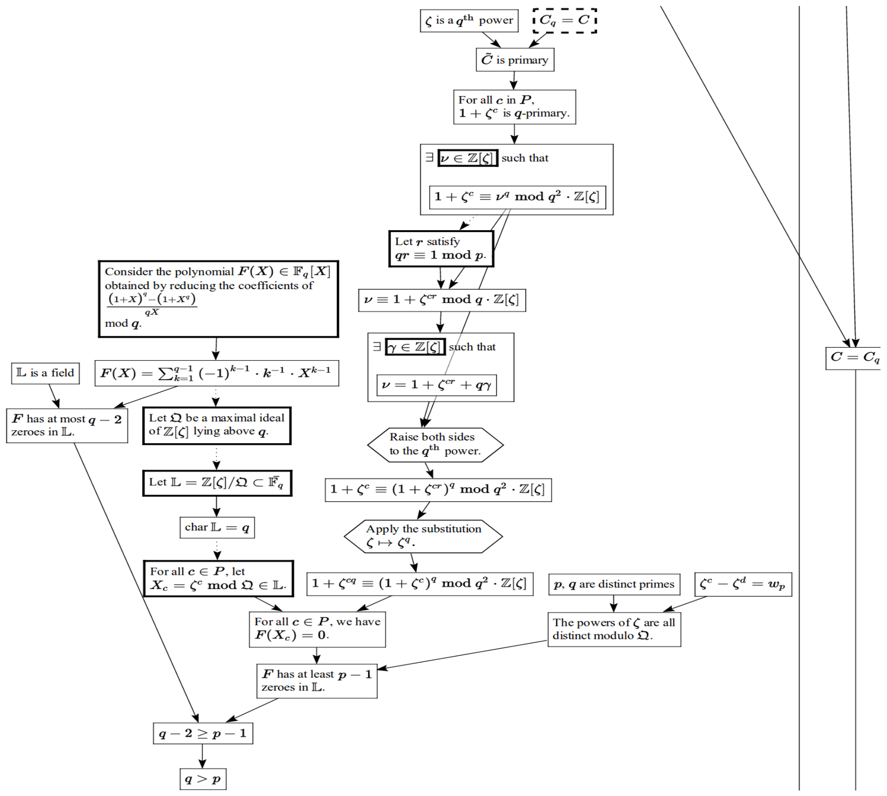
These are some of the ways in which Proofscape can help to show the structure of a proof, and to organise the work involved in studying it. The user quickly learns how the parts of the proof fit together, where the assumptions are used, and where the bulk of the work is apt to lie. The forked lines of reasoning leading off from any pivot objects are easily found, and the purpose of those objects thus made clear. The user can open cited results and clarifications, and close them when finished. The diagram serves as a map to chart the user's progress in confirming each individual inference, and then allows the user to quickly return from the details to the big picture and the overall strategy of the proof.

References

1. Wiedijk, F.: Formal proof sketches. In Berardi, S., Coppo, M., Damiani, F., eds.: TYPES 2003. Volume 3085 of LNCS., Heidelberg, Springer (2004) 378–393
2. Lamport, L.: How to write a proof. The American Mathematical Monthly **102**(7) (1995) 600–608
3. Cairns, P., Gow, J.: A theoretical analysis of hierarchical proofs. In Asperti, A., Buchberger, B., Davenport, J.H., eds.: Mathematical Knowledge Management. Volume 2594 of LNCS., Heidelberg, Springer (2003) 175–187
4. Leron, U.: Structuring mathematical proofs. The American Mathematical Monthly **90**(3) (1983) 174–185
5. Harrell, M.: Using argument diagramming software in the classroom. Teaching Philosophy **28**(2) (2005) 163–177
6. Gordon, T.F., Walton, D.: The Carneades argumentation framework. Frontiers in Artificial Intelligence and Applications **144** (2006) 195
7. Van Gelder, T.: Argument mapping with reason!able. The American Philosophical Association Newsletter on Philosophy and Computers **2**(1) (2002) 85–90
8. Toulmin, S.: The Uses of Argument. Cambridge University Press, Cambridge (1969)
9. Siekmann, J., et al.: LΩUI: Lovely Ωmega User Interface. Formal Aspects of Computing **11** (1999) 326–342
10. Watabe, T., Miyazaki, Y.: Visualization of logical structure in mathematical proofs for learners. In Lee, R., ed.: Computer and Information Science 2012. Volume 429 of Studies in Computational Intelligence., Heidelberg, Springer (2012) 197–208
11. Alcock, L.: e-proofs: Online resources to aid understanding of mathematical proofs. MSOR Connections **9**(4) (2009) 7–10
12. Mihăilescu, P.: Primary cyclotomic units and a proof of Catalan's conjecture. J. Reine Angew. Math. **572** (2004) 167–195


Suppose Catalan's equation has a nontrivial solution with $p, q \geq 3$.
Baker's methods for linear forms in logarithms
There is a rational integer v such that
$\Phi_p(x) = \frac{x^p-1}{x-1} = pv^q$
$p > q$
$p \not\equiv 1 \bmod q$
Thm.Thm3.Crelle572Catalan.Mihailescu.p.m.src
$((x-\zeta)(x-\bar{\zeta}))^{\Omega(\hat{\sigma}-1)} = \mu^q$
Thm.Thm4.Crelle572Catalan.Mihailescu.p.m.src
$C = C_q$
Prop.Prop1.Crelle572Catalan.Mihailescu.p.m.src
$q > p$
$\perp$
Catalan's equation $x^p - y^q = 1$ has no nontrivial solutions with primes $p, q \geq 3$.

(a) Mihăilescu's Thm. 5 and its proof are open.


ζ is a q^{th} power
$C_q = C$
$\tilde{C}$ is primary
For all c in P, $1 + \zeta^c$ is q-primary.
$\exists$ $\nu \in \mathbb{Z}[\zeta]$ such that
$1 + \zeta^c \equiv \nu^q \bmod q^2 \cdot \mathbb{Z}[\zeta]$
Let r satisfy $qr \equiv 1 \bmod p$.
Consider the polynomial $F(X) \in \mathbb{F}_q[X]$ obtained by reducing the coefficients of $\frac{(1+X)^q - (1+X^q)}{qX}$ mod q.
$\nu \equiv 1 + \zeta^{cr} \bmod q \cdot \mathbb{Z}[\zeta]$
$\exists$ $\gamma \in \mathbb{Z}[\zeta]$ such that
$\nu = 1 + \zeta^{cr} + q\gamma$
$C = C_q$
$\mathbb{L}$ is a field
$F(X) = \sum_{k=1}^{q-1} (-1)^{k-1} \cdot k^{-1} \cdot X^{k-1}$
F has at most $q-2$ zeroes in $\mathbb{L}$.
Let Ω be a maximal ideal of $\mathbb{Z}[\zeta]$ lying above q.
Raise both sides to the q^{th} power.
Let $\mathbb{L} = \mathbb{Z}[\zeta]/\Omega \subset \bar{\mathbb{F}}_q$
$1 + \zeta^c \equiv (1 + \zeta^{cr})^q \bmod q^2 \cdot \mathbb{Z}[\zeta]$
char $\mathbb{L} = q$
Apply the substitution $\zeta \mapsto \zeta^q$.
For all $c \in P$, let $X_c = \zeta^c \bmod \Omega \in \mathbb{L}$.
$1 + \zeta^{cq} \equiv (1 + \zeta^c)^q \bmod q^2 \cdot \mathbb{Z}[\zeta]$
p, q are distinct primes
$\zeta^c - \zeta^d = w_p$
For all $c \in P$, we have $F(X_c) = 0$.
The powers of ζ are all distinct modulo Ω.
F has at least $p-1$ zeroes in $\mathbb{L}$.
$q - 2 \geq p - 1$
$q > p$

(b) Prop. 1 and its proof have been opened. Only one node of Thm. 5 is still visible on the far right, while the others are above and below the large box for Prop. 1.

Fig. 2: Using Proofscape