On the Practical Semantics of Mathematical Diagrams

Dave Barker-Plummer
Center for the Study of Language and Information,
Ventura Hall, Stanford University,
Stanford, California, 94305, USA

dbp@csli.stanford.edu

Sidney C. Bailin
Knowledge Evolution, Inc.,
1050 17th Street NW, #520,
Washington, DC, 20036, USA

sbailin@kevol.com

Abstract

This paper describes our research into the way in which diagrams convey mathematical meaning. Through the development of an automated reasoning system, called &/GROVER, we have tried to discover how a diagram can convey the meaning of a proof. &/GROVER is a theorem proving system that interprets diagrams as proof strategies. The diagrams are similar to those that a mathematician would draw informally when communicating the ideas of a proof. We have applied &/GROVER to obtain automatic proofs of three theorems that are beyond the reach of existing theorem proving systems operating without such guidance. In the process, we have discovered some patterns in the way diagrams are used to convey mathematical reasoning strategies. Those patterns, and the ways in which &/GROVER takes advantage of them to prove theorems, are the focus of this paper.

Key words: Mathematical diagrams, reasoning strategies, visualization, proof, automated reasoning.

Introduction

Diagrams and visual images play an essential role in both the comprehension and communication of mathematical proofs. We contend that this role is to make the content of the proof "real" rather than formal. Diagrams are used to represent the objects and relations to which a proof refers. When successfully used, the validity of a proof can be "seen" in the diagram rather than justified as a step-by-step application of formal rules. We suggest that visualization distinguishes "following" a proof from "seeing" it to be true. In the former case, the proof is not fully assimilated, and thus, we might argue, not fully understood.

What distinguishes the full comprehension of a proof from just following the individual steps? The difference concerns the interpretation of the mathematical language: whether it is understood as a purely formal system of formulae, rules, and inferences, or whether it points to something that, however abstract, is real in the world of the mathematician.

Visualization, then, is a means by which mathematics sheds its purely formal character and takes on meaning. As such, it is a key aspect not just of mathematical learning but also of mathematical discovery. Diagrams, in turn, are a vehicle for communicating the visualized images. Far from being an expendable aid, diagrams play an essential role in the communication of mathematical meaning.

This paper summarizes our research into the way in which diagrams convey mathematical meaning. Through the development of an automated reasoning system, called &/GROVER, we have tried to discover how a diagram can convey the meaning of a proof. &/GROVER is a theorem proving system that takes, as its input, not only a theorem to be proved but also a diagram intended to represent the essence of the proof. &/GROVER interprets the diagram as a strategy for performing a detailed formal proof. The diagram focuses &/GROVER's attention on the relevant facts at each stage of the proof.

&/GROVER consists of two parts: GROVER the diagram processor which is the subject of this paper, and an underlying theorem prover, called &. The diagram processor constructs a strategy on the basis of information extracted from the diagram; & is then called upon to prove the subgoals in this strategy.

GROVER is a prototype system. Our eventual aim is to build a system capable of extracting from a diagram the same information that a human can. Development of GROVER has, conversely, yielded insights into the kinds of reasoning involved when humans infer meaning from a diagram.

Three non-trivial theorems which we have proved fully automatically using the &/GROVER system are:

1. The Diamond Lemma, a theorem from the theory of well-founded relations, described in (Barker-Plummer & Bailin 1992), and which we briefly recap here,
2. The Multiple Peaks Theorem, a generalization of the Diamond Lemma, which we describe in this paper,
3. The Schröder-Bernstein Theorem, a theorem from the theory of functions, whose proof we also describe here.

We chose to study these theorems for the following important reasons:

- Each of these theorems is non-trivial for automated reasoning systems. Indeed, in each case we know of no other automated reasoning system which is capable of producing fully automated proofs of any of them.
- Despite the power of the underlying & theorem prover, that system alone is not able to prove the theorem without the guidance that it obtains from GROVER's interpretation of the diagram. This indicates that the diagram is playing a crucial role in the derivation of the proof.
- Finally, when presented in tutorial mode, either in a textbook or in a classroom setting, the theorems are often explained using diagrams to motivate the proof. In our experience, the diagrams which accompany such presentations are canonical — they vary little between independent presentations — and furthermore, when called upon to do so, we ourselves remember the diagrams and then reconstruct the proofs from them, rather than remembering the proofs directly. We take this as indication that the diagram is playing a key psychological role in the proof.

In working with these theorems we have discovered a number of heuristics that appear to play a significant role in the interpretation of a mathematical diagram. The heuristics concern the identification and ordering of steps in the proof strategy, and the determination of relevant facts to be used at each stage of the proof.[1]

How can a Proof be Seen?

We have used GROVER to test some hypotheses about proof visualization, which we describe in this section. The basic hypothesis is that visualizations are partial models of the world to which a proof refers. They are partial because mathematical worlds are typically infinite (for example, the integers, the real numbers, and the universe of sets) and mathematical theorems typically quantify over all objects in such a world.

A visualization of such a theorem consists of exemplars of the patterns asserted to hold. When we prove a universal statement of the form

$$\forall x.A(x)$$

for example, we typically say something like "let c be an arbitrary x," and then proceed to demonstrate $A(c)$. If A is an existential formula of the form

$$\exists y.B(x,y)$$

then we might construct a y for which $B(c,y)$ holds, or we might prove $B(c,y)$ by assuming its negation, $\forall y.\neg B(c,y)$ and deriving a contradiction. This too will typically involve instantiating y, at some point, to one or more specific objects, from which a contradiction is derived.

We hypothesize that the diagram illustrating such a proof is a trail of the instantiations performed along the way: the objects themselves, together with a representation of the relevant facts about them. These facts are mathematical assertions composed of primitive or defined relations between the objects, and logical operators such as conjunction, disjunction, negation, and implication. The repertoire of relations depends on the particular "world" or field of mathematics in which the theorem is being proved.

In general, the logical operators are not explicitly represented in a visualization. They may serve to interpret the relationships between several images that arise in the course of a proof. For example, a proof by cases, which involves deriving the theorem from a disjunction

$$A \vee B \cdots \vee C$$

might involve separate visualizations of A, B, and C. Implication is somewhat more complicated: the proof of

$$A \rightarrow B$$

might involve starting with a visualization of A and elaborating it so that it becomes a visualization of B. This is, in fact, our understanding of the relationship between the hypotheses of a theorem and its conclusion as they appear in a visualization of a proof.

We conjecture that the primary role of visualizations is to represent the relations between exemplar objects. Depending on the particular field of mathematics, there may be preferred representations of certain relations. For example, in set theory we typically illustrate the *subset* relation by containment of one circle within another.

These observations lead us to the first major decision in the design of GROVER:

> A diagram represents a set of facts concerning the properties of, and relations between, exemplar objects that are identified in the course of a proof.

The interpretation of a diagram as a trail of the exemplars invoked in the course of a proof is one of our basic ideas, which we have validated against several (very different) theorems. We develop this idea in the next section.

Example of a Diagram-Based Proof: The Diamond Lemma

The Diamond Lemma, a theorem in the theory of well-founded relations, states that a *well-founded* relation that is *locally confluent* is also *globally confluent.*

The definitions of these terms are as follows:

- The *domain* of a relation R is the set of all elements that are related by R to some other element, that is, all a such that for some b, either $R(a,b)$ or $R(b,a)$.

[1] A more detailed description of the work presented here, including more detail on all of the three proofs mentioned above, can be found in (Barker-Plummer & Bailin 1997).

- A relation R is *well founded* (WF_R) if there are no infinite R-chains, that is, no infinite series of elements $a, b, c \ldots$ such that $R(a,b)$, $R(b,c)$, ...
- A relation R is *locally confluent* (LC_R) if and only if for any three elements a, b, and c in the domain of R, if $R(a,b)$ and $R(a,c)$, then there is an element d such that $R(b,d)$ and $R(c,d)$.
- The *transitive closure* of R is the relation R^* such that $R^*(a,c)$ if and only if there is an R-chain from a to c, that is, a series $b_1, b_2, \ldots b_n$ such that $R(a,b_1)$, $R(b_1,b_2)$, $\ldots R(b_{n-1},b_n)$, $R(b_n,c)$.
- The relation R is *globally confluent* (GC_R) if and only if its transitive closure R^* is locally confluent.

The standard proof of this theorem uses a diagram that begins as the upper half of the diamond in Figure 1 and is elaborated in steps, eventually yielding the element h. The proof begins by assuming that arbitrary elements a, b, and c have been selected with $R^*(a,b)$ and $R^*(a,c)$. Since $R^*(a,b)$, there is an R-chain from a to b and therefore there is an element d that is the first element of this chain. Similarly, there is an e one step along an R-chain from a to c.

Now the local confluence of R is invoked to deduce that there is an element f which completes the small diamond shown in Figure 1.

The next step of the proof uses *transfinite induction*, which is a technique for proving properties about the domain of a well-founded relation. Transfinite induction states that, in order to prove a property $P(a)$ for all elements a in the domain of a well-founded relation R, it suffices to show that the property "climbs up" R. That is, it suffices to show:

> For every x in the domain of R, $P(x)$ holds if it holds for every y that is "lower" than x

where y is "lower" than x if $R(x,y)$.

Transfinite induction is applied by observing that e is "lower" than a: we can, therefore, assume the theorem to hold when e is the upper vertex of an R^* diamond. We now have the upper half of an R^* diamond with vertex e, the other elements being f and c. Although e and f are illustrated as related by R, not R^*, we can see that there is an R-path from e to f with no intermediate elements (the degenerate case), and therefore $R^*(e,f)$ holds.

Applying the theorem to the half-diamond with vertices e, f and c, we obtain an element g such that $R^*(f,g)$ and $R^*(c,g)$.

The next step is to observe that there is an R-path from d to g, passing through f. Thus, $R^*(d,g)$ holds even though it is not explicitly noted in the Figure 1.

The R-path from d to g provides another opportunity to apply transfinite induction. This time we observe that d is "lower" than a and therefore that the half-diamond with vertices d, b, and g can be completed with an element h as shown in Figure 1.

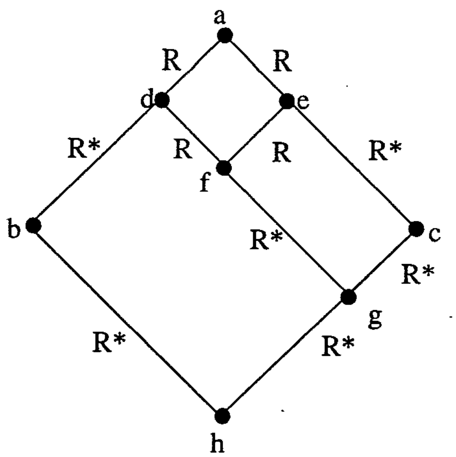

Figure 1: Completion of the proof of the Diamond Lemma

Finally, observing in Figure 1 that there is an R-path from c to h (through g), we see that the theorem has been successfully proven.

Diagrams as Staged Observations: The Existential Solve Heuristic

Existential solve is a heuristic procedure that we use to the infer the trail of existence proofs implicit in a diagram. We first developed *existential solve* as a means to prove the Diamond Lemma automatically. We then discovered that it plays an essential role in the proofs of two other difficult theorems, which are described in this paper.

Existential solve implements the reasoning described in the proof of the Diamond Lemma above. The goal of the heuristic is to construct a sequence of lemmas, each of which proves the existence of (or "solves" for) one existential object in the diagram.[2] The key point is that the objects are solved for successively, one at a time. The heuristic is used to determine which object to solve for first, which next, and so on.

Solving for an existential object means proving the existence of an object that has the properties asserted in the diagram. This is not as obvious a process as it might seem, however, because some properties may involve other objects which may not have been solved for yet. The procedure must, therefore, not only determine a succession of existential objects, but for each such object it must decide which properties of the object are to be considered *defining* properties.

The key idea in *existential solve* is to use the availability of defining properties as the principal criterion for ordering the existential objects. A *defining property* is a formula whose variables consist only of the

[2]Throughout this paper we indicate that an object is existential by writing it in bold face.

following:

- One and only one existential object that has not already been solved for,
- Universal objects,
- Existential objects that have already been solved for.

At the beginning of the proof of the Diamond Lemma there are two existential objects with defining properties, **d** and **e**. When there is more than one candidate, *existential solve* chooses the existential object whose defining properties, taken together, contain the most other objects (universals and previously solved for existentials). The rationale for this criterion is that a greater number of objects in the properties means, in some sense, more information, or greater constraint, and thus a stronger definition. If there are ties when this criterion is applied, *existential solve* proves the existence of the remaining candidates in logical parallel. That is, a random order is used, but since the defining properties of each object do not reference the competitor objects, the selected order has no effect on the resulting proofs.

Existential solve organizes the existential objects in the diagram into a partial order by repeatedly applying the criteria just described. With each selection of the next object to be solved for, that object becomes available to appear in the defining properties of other objects. Eventually, every existential object will have at least one defining property, and the ordering process is then complete.

Existential Solve in the Diamond Lemma

To see how *existential solve* works in the Diamond Lemma, we apply it to the diagram in Figure 1. The following formulae are explicitly represented in the diagram:

$$\begin{array}{llll} R(a,b) & R(a,c) & R(a,\mathbf{d}) & R(a,\mathbf{e}) \\ R^*(\mathbf{d},b) & R^*(\mathbf{e},c) & R(\mathbf{d},\mathbf{f}) & R(\mathbf{e},\mathbf{f}) \\ R^*(c,\mathbf{g}) & R^*(\mathbf{f},\mathbf{g}) & R^*(b,\mathbf{h}) & R^*(\mathbf{g},\mathbf{h}) \end{array}$$

All of the objects are existential except a, b, and c, which are identified as universal in the hypothesis of the theorem.

In the first pass of *existential solve*, the existential objects with potentially defining properties are **d**, **e**, **g**, and **h**. The defining properties of **d** are

$R(a,\mathbf{d})$ and $R^*(\mathbf{d},b)$

with universals a and b. The defining properties of **e** are

$R(a,\mathbf{e})$ and $R^*(\mathbf{e},c)$

with universals a and c. The only defining property of **g** at this stage is $R(c,\mathbf{g})$, and the only one for **h** is $R(b,\mathbf{h})$. Since each of these contains only one universal, **g** and **h** are ruled out at this stage. There is no way to break the tie between **d** and **e**, so the order in which they are solved for is randomly chosen.

In the next pass, d and e may appear in the defining properties of other objects, so the object **f** has the defining properties

$R(d,\mathbf{f})$ and $R(e,\mathbf{f})$

The presence of the two previously solved for objects, d and e, means that **f** now wins out over **g** and **h**, each of which still has only one defining property containing only one other object.

In the next pass, **g** has the defining properties

$R(f,\mathbf{g})$ and $R(c,\mathbf{g})$

Now **g** wins over **h** because its defining properties contain two other objects, f and c, while **h** still has only one defining property, containing one other object.

In the final pass, **h** has the defining properties

$R(b,\mathbf{h})$ and $R(g,\mathbf{h})$

and this marks the end of the trail.

Diagrams as Elisions of Infinitely Many Observations

Recall our view of diagrams as partial models of the world to which a proof refers. Diagrams are finite, while mathematical worlds are typically infinite. While a theorem may quantify over an infinite range of objects (as in "for every integer $i \ldots$"), a diagram expressing the theorem will focus on an arbitrary example in that range (as in "let i_0 be an arbitrary integer").

In some proofs, the diagram consists of a finite number of such exemplars plus a finite number of existential objects that are "defined" (more precisely, proven to exist) in terms of these exemplars — and relations between these objects. Frequently, however, this does not suffice to convey a proof. In many cases it is necessary to represent an infinite range of objects through *elision*. The ellipsis notation (...) is the most common means of expressing an infinite range through elision.

When a mathematical argument relies on the implicit performance of an arbitrary number of calculations or operations, rigorous presentation of the argument must be based on inference rules that permit such reasoning. The most common of such rules are various forms of *induction*.[3] When the "arbitrary" number is finite (but arbitrarily large), the appropriate rule is some form of *mathematical induction* — that is, induction over the natural numbers — as opposed to *transfinite induction* which operates over an arbitrary (possibly infinite) well-founded tree.[4]

When GROVER detects the presence of ellipses in a diagram, it tries to determine whether a finite (but arbitrarily long) series is being represented, and hence whether mathematical induction should be applied. If the objects connected by the ellipses are labeled similarly except for numerical (integer) subscripts, GROVER

[3]These are not the only rules that permit such reasoning: others include the Axiom of Choice and its many equivalents.

[4]A well-founded tree is one that may have infinite branching but no infinite paths.

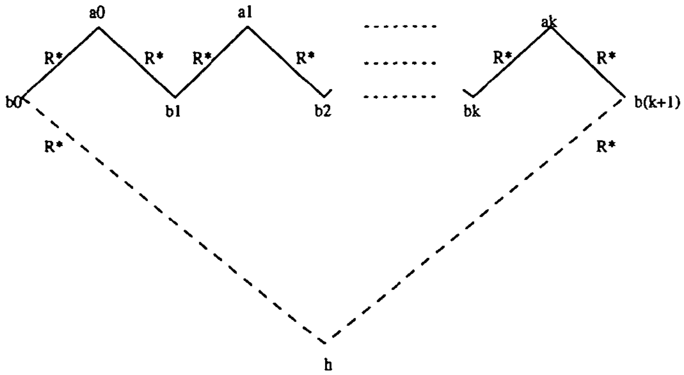

Figure 2: Graphical Statement of The Multiple Peaks Theorem

interprets this to indicate a situation requiring mathematical induction.

A proof by mathematical induction consists of two parts, the *base case* and the *step case.* The base case proves the theorem for the first element in the series of objects. The step case proves the theorem for an arbitrary element in the remainder of the series, on the assumption that it holds for the preceding element. Accordingly, when GROVER recognizes a diagram calling for mathematical induction, it decomposes the diagram into two simpler diagrams, one for the base case and one for the step case, using the ellipses to determine where the separation should occur.

GROVER's assumption in performing this decomposition is that each of the resulting diagrams will contain enough information to prove its part of the theorem. In particular, the diagram for the step case must not only express the desired conclusion (e.g., that the property $P(x)$ holds for $x = n+1$), but it must also express the inductive hypothesis (i.e., that $P(n)$ holds) which will be used to derive the conclusion $P(n+1)$. GROVER verifies this as part of the more general process of matching the diagram with the corresponding conjecture. This process was described in (Barker-Plummer & Bailin 1992). If the step case diagram does not represent the induction hypothesis, the process will fail even before a proof is attempted. In this sense, GROVER has a built-in safeguard against improperly interpreting the ellipses.

The Multiple Peaks Theorem

To see how the interpretation of ellipses works, we present the proof of the Multiple Peaks Theorem. In Figure 2, R^* refers to the *transitive closure* of a relation R. The theorem states that, if R is globally confluent then an object h can be found so that the figure can be completed along the dotted lines, i.e., $R^*(b_0, h)$ and $R^*(b_{k+1}, h)$.

Most notable about this theorem is the fact that the number of "peaks," represented by the variable n, is arbitrary.

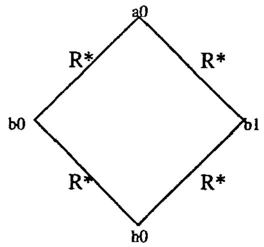

Figure 3: The Diagram for the Base Case of The Multiple Peaks Theorem

The proof is a straightforward application of mathematical induction. The base case ($n = 0$) follows immediately from the global confluence of R. The step case ($n = k+1$) follows from the inductive hypothesis, which gives us the existence of an h_k such that

$$R^*(b_0, h_k) \wedge R^*(b_{k+1}, h_k)$$

Since we also have, from the assumptions of the theorem, that

$$R^*(a_{k+1}, b_{k+1}) \wedge R^*(a_{k+1}, b_{k+2})$$

we infer, from the transitivity of R^*, that

$$R^*(a_{k+1}, h_k)$$

We therefore use the global confluence of R to get the existence of an h_{k+1} such that

$$R^*(h_k, h_{k+1}) \wedge R^*(b_{k+2}, h_{k+1})$$

and then, from the transitivity of R^* again, infer that

$$R^*(b_0, h_{k+1})$$

As in the proof of the Diamond Lemma, the transitivity of R^* is automatically inferred by & and applied where needed.

GROVER interprets each ellipsis in Figure 2 as representing a *sequence* $t_1 \dots t_m$ of objects. Since the objects in one of the sequences are existential, GROVER infers that their existence is to be proven by mathematical induction. GROVER therefore replaces Figure 2 with Figures 3 and 4, and the theorem itself is decomposed into a base case and a step case by applying &'s mathematical induction tactic.

Having decomposed both the diagram and the theorem into two parts, GROVER must now match the terms in each theorem with objects in the corresponding diagram so that the theorem's hypotheses are recognized as facts in the diagram.

Through its analysis of the ellipses as a shorthand for mathematical induction, GROVER is able to associate the diagram subscript k with the induction variable n in the theorem. Completing the association process is complicated, however, by a discrepancy in representation between the theorem and the diagram. The theorem (in its original form as well as

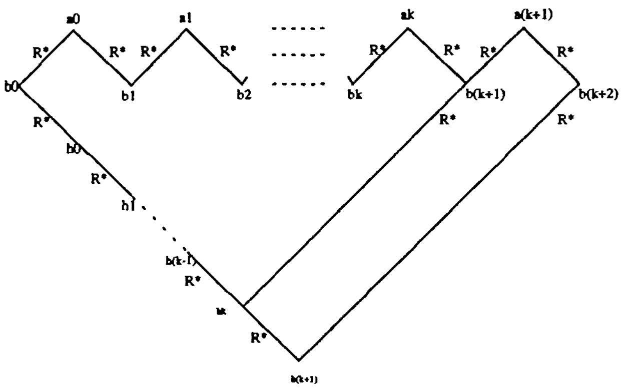

Figure 4: The Diagram for the Step Case of The Multiple Peaks Theorem

in the step-case) does not contain the universal variables $a_0, a_1 \ldots a_k, a_{k+1}$ and $b_0, b_1, b_2 \ldots b_k, b_{k+1}, b_{k+2}$ but rather two universal variables a and b, which are applied as functions to an index variable i. In order to complete the association, therefore, GROVER must establish the correspondence between the variables a and b in the theorem and the instantiated terms in the diagram.

GROVER solves this problem using the idea of *spanning hypotheses* — hypotheses of the form

$$\forall x.(x \leq n + 1 \rightarrow A) \qquad (1)$$

where $n+1$ is associated with a diagram *spanning limit*, which is the subscript of the final term of a sequence in the diagram. GROVER replaces each spanning hypothesis (1) with the instantiated formulae $A[t/x]$ for all *spanning instances* t, which are the diagram objects participating in one of the diagram sequences. GROVER is then able to match the hypotheses of both the base case and step case theorems with facts in their respective diagrams.

Focus of Attention: Choosing Relevant Hypotheses

A diagram fact that has been proven from the conjecture's hypotheses is available as a hypothesis during any individual step of the proof strategy. Furthermore, the conclusion of any previous step in the strategy is available as a hypothesis in subsequent steps. Not all of these potential hypotheses are necessarily useful, however, and in order to facilitate &'s search for a proof, GROVER tries to keep the hypotheses to a minimum. Underlying GROVER's approach is the idea that some previously proven facts are relevant to the current lemma, and some are not. If a hypothesis is explicitly cited as a hint for a given object's existence, GROVER assumes that it is relevant. GROVER determines the relevancy of other facts by comparing the terms found in the current lemma to those found in the potential hypotheses. The objective is to find hypotheses that, taken together, mention all of the terms found in the lemma's conclusion. We call this a process of *covering* all of the lemma's terms.

To determine the hypotheses for a given lemma, a heuristic algorithm examines the preceding lemmas to see whether any of them can contribute to "covering" the current lemma's terms. The algorithm proceeds backwards, examining the most recent lemmas first and then, if necessary, moving on to the earlier lemmas. As this process continues, the set of terms that still need to be covered shrinks.

The obvious selection criterion would be to add a previous lemma to the hypotheses of the current lemma if the previous lemma contains any of the terms remaining to be covered. We found that a more restrictive criterion of relevancy is necessary, however. A measure of relevancy is provided by defining two classes of terms in the current lemma:

1. Terms from the lemma's conclusion that still need to be covered — we call these the *required* terms
2. Terms that appear in the lemma's conclusion or in the hypotheses thus far selected — we call these the *desired* terms

GROVER sorts parallel lemmas by 1) the number of required terms they contain, and 2) within that, the number of desired terms they contain. If none of the parallel lemmas contains any required or desired terms, the algorithm proceeds to the next latest set of parallel lemmas to consider as candidate hypotheses. Otherwise, the parallel lemmas that come out best in the sort — i.e., the highest number of required terms, and within that the highest number of desired terms — are selected as hypotheses.

When the process described above is complete — either because all of the required terms have been covered, or because there are no more earlier lemmas to provide potential hypotheses — GROVER considers the hypotheses of the theorem, and the diagram facts that represent them. GROVER again applies a relevancy criterion to determine which of these might be suitable hypotheses for the current lemma.

To understand how the procedure we have just described helps to prune hypotheses, we consider the final lemma step of the Multiple Peaks Theorem, which is the theorem's conclusion:

$$\exists h.(R^*(b_0, h) \wedge R^*(b_{k+2}, h))$$

We back up to the preceding lemma, which is

$$R^*(h_k, \mathbf{h_{k+1}}) \wedge R^*(b_{k+2}, \mathbf{h_{k+1}})$$

This lemma contains the required term b_{k+2}, but the required term b_0 still needs to be covered, so we back up to the parallel lemmas

$$R^*(b_0, \mathbf{h_0}) \wedge R^*(b_1, \mathbf{h_0})$$
$$R^*(b_0, \mathbf{h_k}) \wedge R^*(b_{k+1}, \mathbf{h_k})$$

Both lemmas contain the required term b_0, so we must look to the desired terms in order to break the tie. The $\mathbf{h_k}$ goal wins because it contains the desired term $\mathbf{h_k}$ while the $\mathbf{h_0}$ goal contains no other desired term.

Diagram Idioms: Visualization and Abstraction

In this section we will describe the process by which we move from the diagram to a collection of formulae which it represents. This is a crucial step in GROVER's automatic processing of the diagram.

One of the key components of &/GROVER is a graphical editor called DEGAS[5]. DEGAS is a rather conventional graphical editor, with tools allowing the drawing of lines, ellipses, and rectangles, and for attaching labels to these objects. The most important feature of DEGAS for GROVER is that it is able to save the diagram in the form of a *geometry facts file* (G-file).[6] The G-file is a generic textual representation of the diagram structure, irrespective of any semantics that we associate with the diagram.

The use of a graphical editor which is able to produce a representation of the diagram at this level of abstraction allows us to avoid some potentially difficult problems in understanding the diagram. We do not, for example, have to be involved in line-finding, recognizing collections of lines as rectangles, worrying about whether a collection of points are collinear, and so on. DEGAS provides us with a representation of the structure of the diagram which is based on the tools used to draw the diagram.

Interpreting the Diagram

When presented with a diagram, GROVER must interpret it as representing facts that are expected to follow from the hypotheses of the current theorem. We have developed a small expert system for carrying out this task. An important point in understanding the development of this system is that we are not attempting to develop a new language for drawing diagrams, rather we are trying to ensure that the system properly interprets the "natural" diagram for proving a given theorem. The rules of the expert system are intended, therefore, to capture the usual practice of mathematical diagrams rather than to define a new language. While we do not believe that a complete and correct set of rules for achieving this goal necessarily exists, we do believe that we can devise a generally useful set of rules that approximate this desire.

We also observe that the rules used in interpreting diagrams will depend on the mathematical context in which the diagram is drawn. For example, a circle in a diagram represents an abstract mathematical circle if the diagram is offered in the context of a geometry proof, while it probably represents a set when offered in a set theory proof like the Schröder-Bernstein Theorem. In addition, the specific diagrammatic idioms used may differ from author to author in an idiosyncratic manner. Both of these factors indicate the existence of a number of diagrammatic idioms used in mathematics, rather than a single unified language of mathematical diagrams.

The interpretation of the diagram is divided into two parts: a local analysis, and a global analysis. The local analysis phase produces atomic formulae from the spatial and explicit relationships in the diagram, and writes them to a *logic file* (L-file). The global analysis phase detects larger constructions in the diagram.

Local Analysis: Geometry To Logic

The analysis of the diagram proceeds in a bottom-up fashion. First the individual objects in the diagram are examined. The labels that are associated with some objects are symbolic representations of the objects. Various types of labels are allowed in our system, corresponding to the practices that we have encountered. The simplest label attaches a name to an object, but more structured labels are possible, for example, the label `"c : R*(a,c)"` indicates that the labeled object is called c and that it has the property $R^*(a,c)$. Other label forms that are allowed include equalities such as `a = f(b)`.

The analysis of the labels, as we have just seen, can lead to some formulae being discovered, but the system may obtain further facts from the geometric relationships between the objects in the diagram. For example, our expert system interprets a dot within a closed figure as the $\in$ relation and a closed figure completely within another as the $\subseteq$ relation.

In addition to arbitrary geometric relationships, relationships may be stated explicitly. For example, given an arc labeled with the formula `"R"` whose end points are dots labeled a and b respectively, we infer that the meaning of the arc is $R(a,b)$. This is because, in the language of our prover, `"R"` has the right form to be a predicate symbol. An alternative reading is possible, namely $\langle a,b\rangle \in R$. This possible interpretation is not eliminated completely by our system, but it is deemed to be less likely (since R is not a legal *term* in the syntax of our logic), and the preferred interpretation is returned by the system. The rules for interpreting arrows are similar to those for interpreting arcs, except that the preferred interpretation of an arrow is as representing a function. For example, in the Schröder-Bernstein Theorem diagram (figure 5), an arrow labeled by the term `"f"` and end-points labeled `"a"` and `"b"` will be interpreted as $\langle a,b\rangle \in f$, since f is a constant term, rather than a predicate symbol, in the & logic.

As another example, in the diagram of figure 5 we use the device of dividing a circle into two parts by a straight line. This indicates a partition of the set represented by the circle into two disjoint subsets. The natural diagram might instead divide the enclosing set by indicating a subset of that set using a second en-

[5] DEGAS is the Diagram Editor for the GROVER Automated System.

[6] DEGAS can also save the diagram as a postscript file, or in a representation suitable for saving and restoring diagrams within DEGAS.

closed circle. GROVER would correctly interpret this diagram as indicating that the enclosed circle represents a subset of the set represented by the enclosing circle, but this would not cause the system to focus on the remainder of the enclosing circle as an object in its own right, which is what we need in the proof of the Schröder-Bernstein Theorem. We can imagine other devices for dealing with this problem, for example the use of shading to indicate the salience of the remainder of the circle as an object.

Global Analysis: Verify Logic

The result of the local analysis of the G-file is a collection of atomic formulae, which are implicitly conjoined. We call this representation a *Logic File* (L-file). Diagrams can represent more complex structures than a flat collection of atomic formulae however. These structures are detected in an analysis of the L-file which we call *verify logic*. *Verify logic* is only activated once the G-file representation has been completely interpreted as an L-file, so it is an operation on logical formulae. In principle, the same processing could be performed on the G-file representation, or interleaved with the *geometry to logic* phase. From an implementors point of view, however, it is simpler to wait until the L-file representation is complete before looking for higher-level structures.

The global analysis is implemented as a collection of "critics", each of which looks for specific conditions that might hold within the diagram, and modifies the logical representation appropriately. For example, one of the critics implemented in GROVER is the `definition by cases` critic.

The `definition by cases` critic is triggered by the presence of two equalities in the L-file of the form $x = t_1, x = t_2$, where x is an existential object, and t_1, t_2 are arbitrary terms involving only universal objects. It is a general feature of diagrams that distinct tokens represent distinct objects (token referentiality, see (Barwise 1993)), and therefore such a pair of equalities present a puzzle on the face of it. One explanation is that the diagrammer is attempting to assert $t_1 = t_2$, but the role of x is then unexplained. The `definition by cases` critic attempts to gather evidence that the existential object x is being defined by cases, as under some circumstances being equal to t_1 and under other disjoint circumstances being equal to t_2. If such evidence can be found, the equalities $x = t_1$ and $x = t_2$ are replaced by the critic with the more complex formulae: $P \rightarrow x = t_1 \wedge Q \rightarrow x = t_2$, where P and Q are possibly complex formulae representing the two alternative conditions.

Example: The Schröder-Bernstein Theorem

The `definition by cases` critic plays a crucial role in The Schröder-Bernstein Theorem, a theorem from the theory of functions which concerns the way in which the "size" of sets can be measured. The Schröder-Bernstein Theorem states that if there is a one-one function (an *injection*) from the set A *into* the B, and a one-one function from B into A, then there is a *bijection* between the two sets, i.e., a one-one function from A *onto* B.

$$\forall f, g, A, B. Injection(f, A, B) \wedge Injection(g, B, A) \rightarrow \exists h. Bijection(h, A, B)$$

An intuitive proof of the Schröder-Bernstein Theorem would proceed as follows: The bijection h must be some combination of of f and g^{-1}, i.e., for each $a \in A$, $h(a)$ will be either $f(a)$ or $g^{-1}(a)$. The problem is therefore to define a partition of A into sets A_1 and A_2 so that h behaves like f for members of A_1 and g^{-1} on members of A_2. Since h is to be a bijection, every $b \in B$ will have to be in $range(h)$. Therefore, if b is not in $range(f)$, then $h^{-1}(b)$ must be in A_2. So A_2 contains $g^{-1}(B - range(f))$. Moreover, A_2 must be closed under $g \circ f$, because if $a \in A_2$ then $h(a) = g^{-1}(a)$, so $h(a)$ cannot be $f(a)$ unless $f(a) = g^{-1}(a)$. Therefore, unless $f(a) = g^{-1}(a)$, $f(a)$ must be "hit" under h by some other element of A, which can only be $g(f(a))$. So let A_2 be the smallest set containing $g^{-1}(B - range(f))$ and closed under $g \circ f$, and let A_1 be $A - A_2$.

The diagram of figure 5 illustrates this strategy.

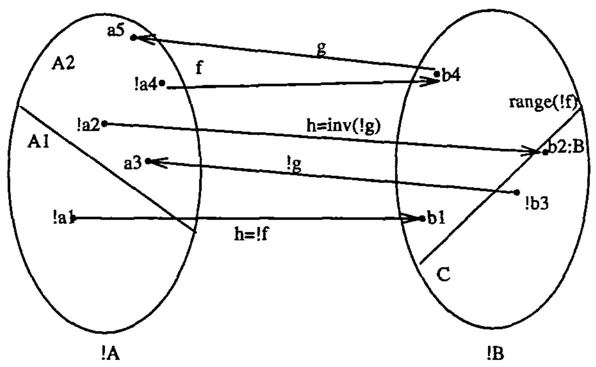

Figure 5: The Diagram for the Schröder-Bernstein Theorem

The diagram contains objects **h**, $\mathbf{A_1}$, $\mathbf{A_2}$ and **C** whose existence must be proved, and in addition it represents the *definition* of these objects.

The two arrows defining the function **h** in the diagram, one arrow labeled $\mathbf{h} = f$ and the other labeled $\mathbf{h} = g^{-1}$ are recognized by the `definition by cases` critic as indicating a definition of the function **h** by cases. The critic looks at the source points of the respective arrows, to determine whether they indicate that the function **h** is defined to be f on some subset of its domain, and g^{-1} on the other subset of the domain.

Three other critics are needed in the diagrammatic proof of The Schröder-Bernstein Theorem. The

function chains critic looks for information concerning items at the end points of arrows, in order to construct appropriate assertions concerning the relationships between objects at the ends of these arrows. The diagram of figure 5 contains universal objects a_1, a_2, a_4 and b_3, whose role in the diagram is to serve as starting points for function arrows. Since these are universal objects, they are examplars for arbitrary objects with the same properties that they themselves exhibit. The function chains critic generalizes the formulae containing these objects to universal formulae.

b_3, for example, is a member of $\mathbf{C}$ which is mapped by the function g onto some member of A_2. Rather than view this structure as three distinct formulae, $b_3 \in \mathbf{C}$, $\langle b_3, \mathbf{a_3} \rangle \in f$ and $\mathbf{a_3} \in \mathbf{A_2}$, we recognise that the geometric structure is intended to represent that every member of $\mathbf{C}$ is mapped by g to some member of $\mathbf{A_2}$.

The function chains critic examines the L-file for formulae which match this pattern, constructing the appropriate generalizations of the specific formulae. The part of the diagram which is significant for this step is shown in figure 6.

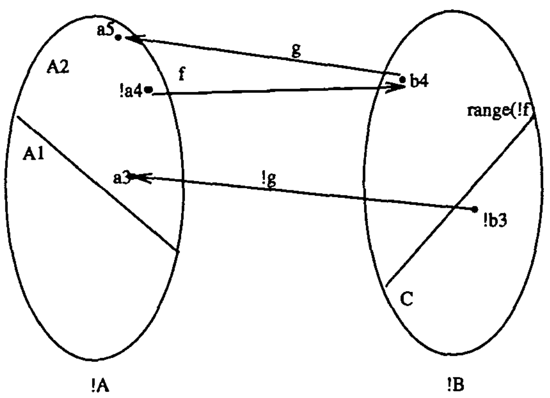

Figure 6: Function Chains

The result of applying the function chains critic to the formulae just mentioned is the new formula:

$$\forall b3.(b3 \in \mathbf{C} \rightarrow \forall x.(\langle b3, x \rangle \in g \rightarrow x \in \mathbf{A_2}))$$

The same critic notes that formulae $a_4 \in A_2$, $\langle a_4, b_4 \rangle \in f$, $\langle b_4, a_5 \rangle \in g$ and $a_5 \in A_2$ indicate that an arbitrarily chosen element in $\mathbf{A_2}$ maps under $g \circ f$ back into $\mathbf{A_2}$.

The formulae involving a_4, b_4 and $\mathbf{a_5}$ have the same structure, except that this represents a chain of function applications. Again, the chain beginning with the universal object is traversed, and the properties of the beginning and end points of the chain examined. The result is a universal formula which asserts that all start points with the same properties as the exemplar are mapped by the same chain, to end points with the same properties as *its* exemplar.

The result of applying this critic to the chain is:

$$\forall a4.(a4 \in \mathbf{A_2} \rightarrow \\ \forall x, y.((\langle a4, x \rangle \in f \wedge \langle x, y \rangle \in g) \rightarrow y \in \mathbf{A_2})))$$

These formulae capture the intent of the larger structure in the diagram, by aggregating facts recognized as forming a pattern into an appropriate compound formula.

On the basis of the formulae derived by the function chains critic, the **Closure** critic recognizes that $\mathbf{A_2}$

1. contains the image under g of $B - range(f)$,
2. and is closed under the composition $g \circ f$.

and therefore that $\mathbf{A_2}$ is (probably) intended to be the closure of the given base set under the composition of g and f. The crucial part of the diagram for this critic is coincidentally identical to the part relevant to the function chains critic, so consult figure 6.

The choice to consider $\mathbf{A_2}$ as the closure rather than some superset of the closure is heuristic, but we believe that this is generally likely to be the intention, particularly when no additional information about the set is available, as in this case. The choice of $\mathbf{A_2}$ as the closure means that we will add a formula to the L-file indicating that $\mathbf{A_2}$ is a subset of all non-empty sets with the properties 1 and 2. Note that this formula does not imply that the set A_2 itself enjoys properties 1 and 2 above. A proof of this fact must be constructed by the theorem prover later in the processing.

The closure critic adds the hint that $\mathbf{A_2}$ is defined to be this intersection. This hint is used when the individual goals of the strategy are constructed.

The final critic used in the proof is the **generalize domain and range** critic, which is responsible for inferring the intended domains and ranges of *Function* assertions. In the diagram of figure 5, the only arrows labeled by g have target points in $\mathbf{A_2}$, but we do not know that g's range is just $\mathbf{A_2}$. Indeed in the intended proof, g is an injection into A. The **generalize domain and range** critic examines the diagram looking at all of the target points of arrows sharing the same label. Having identified these end points the critic identifies the largest graphical object containing all of these end points, and asserts this as the set into which the function maps. This results in the L-file formula $Function(g, B, \mathbf{A_2})$ being replaced by $Function(g, B, A)$, and $Function(f, A, range(f))$ by $Function(f, A, B)$.

Like the **closure** critic, the action of the **generalize domain and range** critic can be undesirable. It may over-generalize, since for example the intended range of g may indeed have been $\mathbf{A_2}$, or undergeneralize, since the intended range of the function may in fact not appear as a object in the diagram, but may contain the inferred range. Experience with other diagrams will determine which of these cases is the most likely to occur, and the diagram cues that we may use to determine the likely intended values for the domains and ranges of sets.

Other Approaches to Graphical Theorem Proving

We are aware of work in graphical theorem proving by Gelernter, Barwise and Etchemendy, and Pastre. We try to identify both the similarities and differences of our work with these other approaches.

Gelernter's Geometry Machine

The concept of graphical theorem proving was introduced by Gelernter in his Geometry Machine ((Gelernter 1963; Gelernter & others 1963; Gilmore 1970)). GROVER resembles the Geometry Machine (GM) in the following respects:

- The diagram is used as a model of the goal to be proven
- The diagram suggests constructions of terms that are needed in the intended proof

Although GM is oriented specifically towards proving theorems in Euclidean Geometry (the prover uses a set of axioms for Euclidean Geometry, which provide the basis for proving the existence of needed terms), Bundy has shown how the approach can be applied to other domains if one replaces the concept of "diagram" with the concept of "model" ((Bundy 1983), pp. 142-149). The principal difference between the Gelernter approach and GROVER concerns the way in which the graphical information is used. In GM, the diagram is consulted in order to guess bindings that will prove the current subgoal. It is assumed that if the instantiated subgoal is true in the diagram, it may be provable. GROVER offers a different form of guidance: the advice takes the form of specifying the subgoals themselves. Thus with GROVER the high-level structure of the proof is determined by the diagram.

Bundy's generalization of GM provides a method of pruning irrelevant formulae from the proof. In terms of &, this would mean discarding a conclusion C_i of a goal sequent $A_1 \cdots A_m \vdash C_1 \cdots C_n$ if the model can be used to refute C_i, that is, if variables of C_i can be bound to elements in the model in such a way that the instantiated form of C_i is false in the model (it must be remembered here that the variables in & play the role conventionally played by constants in other provers, and that schematic terms in & play the role conventionally played by variables). There is no explicit analog to Bundy's method in GROVER, but we believe that it is implicitly present because the diagram itself is what suggests the subgoals; thus the C_i are already known to be true in the diagram.

Barwise and Etchemendy's Hyperproof

The work of Jon Barwise and John Etchemendy in graphical reasoning stems from an underlying belief that visual information is frequently a more effective medium for reasoning than is text ((Barwise & Etchemendy 1994; 1990b)). They present several examples to illustrate this point. They have been developing a theoretical foundation for this work in the form of an algebraic theory of *infons*, that is, information that can be manipulated independently of the specific representations it may take. One of its key results is the identification of five basic rules for manipulating information, which provide the basis for the applied work ((Barwise & Etchemendy 1990a), Section 3).

On the applied side, they have developed a system called Hyperproof which allows the user to reason about the blocks world both graphically and with formulas. A subsequent version of Hyperproof, which will support a more general class of problems, is now being developed.

Hyperproof is a purely interactive tool, i.e., a proof checker, rather than as an automatic prover such as GROVER. The user can invoke either standard logical inferences on formulas, which are displayed at the bottom of the screen, or graphical inferences on the diagrams. These two forms of reasoning can be interleaved. For example, the user can perform an operation that splits a diagram into two alternative diagrams, each with more information than the original one (proof by cases), and an operation that merges two diagrams into a single diagram containing the information common to both. The graphical inference rules supported by Hyperproof are formulated at a fairly low level, comparable to those of first order logic. Nevertheless, the graphical inferences serve to elide what would otherwise be large blocks of logical inferences. The reason for this power is that the diagrams compress a lot of information into a concise representation.

The high level goals of GROVER and Hyperproof are quite compatible. The implementation approaches seem to reflect different priorities. The emphasis in GROVER has been on the heuristic elaboration of very high level inferences (that is, translating high level claims by the user, which are expressed graphically, into formal proofs), so that the tool becomes a vehicle for discovering, expressing, and verifying proofs at a high level. The emphasis in Hyperproof is on the use of diagrams as a concise representation of complex situations, so that the tool facilitates human reasoning about such situations. In addition Hyperproof embodies a *formal* model of reasoning with diagrams, in which inference rules may diagrams as hypotheses and/or conclusions. In GROVER the diagram is a heuristic, meta-level device which plays no formal role in the eventual proof.

More specifically, the salient differences between Hyperproof and GROVER can be summarized in terms of capabilities found in one but not the other. The key features of Hyperproof that are not now supported by GROVER are:

- Graphical inference operations
- Dynamic interaction of the user with the diagram

The graphical inference operations support the dynamic interaction by enabling the user to, in effect, "draw" individual proof steps. The diagram in Hyperproof is thus an intrinsic part of the interactive reasoning process, manifested in the graphical inferences.

The key features of GROVER that appear to be absent in Hyperproof are:

- Ability for the user to assert an existential claim graphically
- Automated construction of strategies to prove these claims

There is no provision in Hyperproof for the user to add to a diagram an object whose existence must then be proven as a subgoal. In Hyperproof the initial diagram represents the problem situation about which some claim is to be proven; the graphical inferences represent logical inferences on these situation descriptions. In contrast, GROVER provides the user with a graphical means of specifying subgoals of the form $\exists x.P(x)$ whose proofs will assist in the proof of the desired theorem. It is this form of argumentation that we have emphasized in GROVER.

The way in which GROVER processes such user input—the construction of a strategy—is another key difference. GROVER heuristically processes the graphical information to construct an existential assertion, which is then passed to the underlying theorem prover. The construction of these assertions is a non-trivial task since it involves selecting, from all of the information in the diagram, those facts that are relevant to the object x and that should be included in the assertion P. For example, in the proof of the Diamond Lemma, the *existential solve* heuristic collects all those atomic facts that refer to x and to no other object that has not yet been defined.

The two approach are complementary. In Hyperproof the diagram is used as a presentation device that makes it easier for the user to reason from a situation (the assumptions of the theorem) to the desired conclusion. In the current implementation of GROVER, we assume that the user already has a high level proof strategy in mind—in particular, a series of existential subgoals— and that this strategy is most easily expressed by means of diagrams. Thus the challenge in GROVER is for the machine to interpret this graphical expression and derive a best guess at what the user has in mind, and then to carry out the laborious and error-prone details of the proof.

DATTE – The Work of D. Pastre

Pastre has described a theorem prover which uses diagrams to aid the proof of theorems, (Pastre 1977), however this work is quite different from ours. The diagram in Pastre's theorem prover, DATTE, is an internal representation of the formulae that the theorem prover is currently manipulating rather than something that the users provides to guide the prover, as in our case.

The chief similarity between our proposal and Pastre's work is the use of graphical inference rules. In DATTE, various definitions are expressed as "statement rules", which cause the internal diagram to be modified. For example, if the diagram represents $a \subseteq b$ and $x \in a$ then the diagram is updated to represent $x \in b$. This is rather like our proposed graphical inference rules. The major difference is that, in our case, the user provides the diagram as guidance for the prover, and the system views it not only as a representation of formulae, but also as a proof strategy. DATTE's diagram represents only those formulae that are known to be true at a particular point in the proof, not hypothetical information or goals to be proved as in GROVER.

Conclusions

We have described various issues that arise when an automated system tries to interpret a diagram as a mathematical proof. In our investigation of three theorems whose proofs require different techniques — transfinite induction, mathematical induction, and set theory, respectively — we found that a common element is the decomposition of the proof into a series of existence proofs; the diagram suggests the conditions to be proven in "solving" for successive objects. The diagram also suggests the degree of relevance of each previously solved for object to the current existence proof, thus providing a tractable set of hypotheses to be used in each lemma. Finally, patterns in the diagram may suggest higher-order abstractions that are crucial in proving the theorem.

Eventually, our goal is to develop a system that will foster the development of proofs by students of mathematics and even by working mathematicians. By raising the level of the conversation to the types of abstractions contained in diagrams, a theorem proving system could serve as a kind of surrogate colleague with whom ideas are tested and the implications of different constructs explored. GROVER, in its prototype state, is a long way being such a system, but it is a start at uncovering the kinds of meaning embedded in a mathematical diagram.

References

Barker-Plummer, D., and Bailin, S. 1992. Proofs and pictures: Proving the diamond lemma with the GROVER theorem proving system. In *Working Notes of the AAAI Symposium on Reasoning with Diagrammatic Representations, March 25–27th 1992, Stanford, USA.*

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

Barwise, J., and Etchemendy, J. 1990a. Information, infons, and inference. In Cooper, R.; Mukai, K.; and

Perry, J., eds., *Situation Theory and Its Applications.* Center for the Study of Language and Information.

Barwise, J., and Etchemendy, J. 1990b. Visual information and valid reasoning. In Zimmerman, W., ed., *Visualization in Mathematics.* Mathematical Association of America.

Barwise, J., and Etchemendy, J. 1994. *Hyperproof.* CSLI Press. (ISBN: 1-881526-11-9).

Barwise, J. 1993. Heterogeneous reasoning. In Allwein, G., and Barwise, J., eds., *Working Papers on Diagrams and Logic.* Indiana University Logic Group. 1–13.

Bundy, A. 1983. *The Computer Modelling of Mathematical Reasoning.* Academic Press.

Gelernter, H., et al. 1963. Empirical explorations of the geometry theorem proving machine. In Feigenbaum, and Feldman., eds., *Computers and Thought.* McGraw Hill.

Gelernter, H. 1963. Realization of a geometry theorem proving machine. In Feigenbaum, and Feldman., eds., *Computers and Thought.* McGraw Hill.

Gilmore, P. 1970. An examination of the geometry theorem proving machine. *Artificial Intelligence* 1:171 - 187.

Pastre, D. 1977. Automatic theorem proving in set theory. Technical report, University of Paris (VI).