The Role of Diagrams in Mathematical Proofs

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

citation, so here they are in order.

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.

1. Introduction

Open almost any mathematics text book and you will find, along with the familiar symbolism of mathematics and motivational text, many diagrams which are included to help the reader *visualize* the particular point being made. One might be tempted to conclude that mathematicians think in terms of pictures as much as they do in terms of the familiar symbolism. Indeed, the association of visual perception and argument is inherent even in the language that we use: we speak of "seeing" a proof, "transparent" arguments, and "clarity" of thought, for example. The argument as visual object is perhaps one of the *Metaphors we Live By* [2].

Diagrams and visual images play an essential role in both the comprehension and communication of mathematical proofs. 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, therefore, 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. The significance of this distinction for mathematical education, and more generally mathematical communication, can be inferred from the following passage from an old textbook on analysis:

> It is a basic principle in the study of mathematics, and one too seldom emphasized, that a proof is not really understood until the stage is reached at which one can grasp it as a whole and see it as a single idea. In achieving this end, much more is necessary than merely following the individual steps in the reasoning. This is only the beginning. A proof should be chewed, swallowed, and digested, and this process of assimilation should not be abandoned until it yields a full comprehension of the overall pattern of thought. [1, p. xi]

What distinguishes the full comprehension of a proof from just following the individual steps? The verbs "grasp" and "see" in this passage suggest that 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 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 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: 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 section 3, and,
2. The Multiple Peaks Theorem, a generalization of the Diamond Lemma which is described in section 5.1.
3. The Schröder-Bernstein Theorem, a theorem from the theory of functions, whose proof we describe in section 6.4

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. Although we did not view them in this way when we began, we found that the diagrams implicitly represent a series of existence proofs. Relations between objects in the diagram contain hints about the order in which to "solve for" these objects, the properties of each object that characterize a successful existence proof, and the facts to be invoked in proving these properties. We do not claim that all mathematical diagrams are interpreted in this way: we can easily cite counterexamples. Nevertheless, the identification of these principles in diagrams of very different styles, representing theorems from different branches of mathematics, suggests that they have some generality.

2. 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, however, 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, as in figure 1.

Other relations may not warrant preferred idioms because they are too specialized to the context of a particular proof, or they may themselves be arbitrary (that is, universally quantified) and thus do not have a specific identity. In such cases, generic visual idioms are used, such as connecting the objects by lines, locating them in close proximity, or similar techniques.

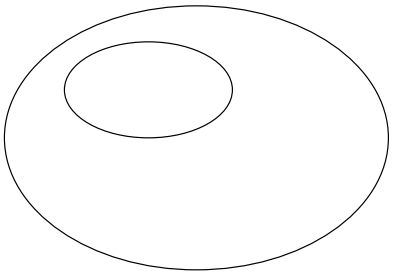

Fig. 1. Certain relations have preferred visual idioms.

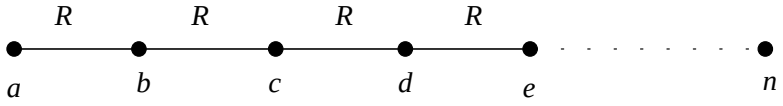

Fig. 2. If the relation is well-founded, all such chains eventually terminate.

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 Section 4. First, however, we present an example of a proof that illustrates the concept.[1]

3. 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 $R(b, a)$.
- 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 $R(a, b)$, $R(b, c)$, $\ldots$ as shown in Figure 2.
- 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 $R(c, d)$. This property can be represented graphically as shown in Figure 3.

[1] Throughout this paper we supress some technical details for clarity of exposition.

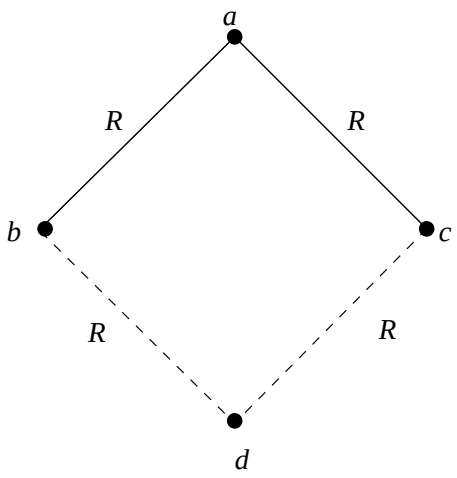

Fig. 3. In a locally confluent relation, the diamond can always be completed.

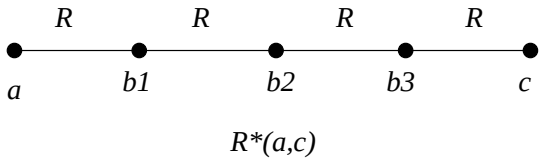

Fig. 4. Transitive closure of a relation R

- 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 $R(a, b_1)$, $R(b_1, b_2)$, $\ldots R(b_{n-1}, b_n), R(b_n, c)$, as shown in Figure 4.
- The relation R is *globally confluent* (GC_R) if and only if its transitive closure R^* is locally confluent.

Thus, the Diamond Lemma can be expressed graphically by stating that, if R is a locally confluent well-founded relation, then the diamond in Figure 5 can always be completed by some element h.

The standard proof of this theorem uses a diagram that begins as the upper half of the diamond in Figure 5 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 such that the top half of the diamond in Figure 5 holds. 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. This is shown in Figure 6.

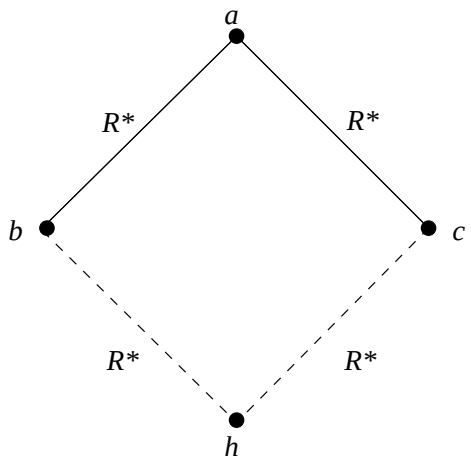

Fig. 5. In a globally confluent relation, the diamond can always be completed.

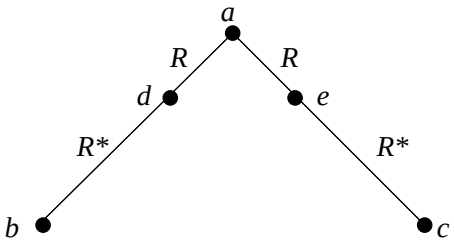

Fig. 6. Interpolation of elements in the Diamond Lemma

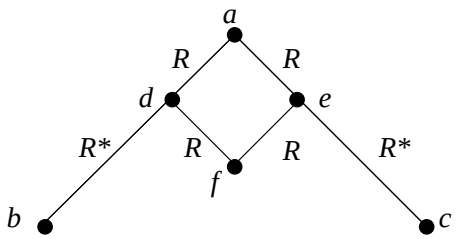

Fig. 7. Application of local confluence in the Diamond Lemma

Now the local confluence of R is invoked to deduce that there is an element f which completes the small diamond shown in Figure 7.

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 to the situation described in Figure 7 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. In Figure 7 we have the upper half of an R^* diamond with vertex e, the other elements being f and c. Although Figure 7 shows e and f to be 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 vertex e, we obtain an element g such that $R^*(f, g)$ and $R^*(c, g)$, as shown in Figure 8.

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 8.

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 9.

Finally, observing in Figure 9 that there is an R-path from c to h (through g), we see that the theorem has been successfully proven.

A few observations about this proof are crucial in trying to automate it. First, the diagram in Figure 9 was constructed in stages, but we present GROVER only with this single diagram, in its complete form. If we regard this diagram as an expression of the proof, then there must be some way to derive its temporal unfolding, or history, from

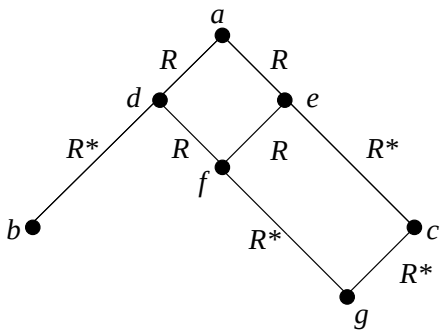

Fig. 8. First application of well-foundedness in the Diamond Lemma

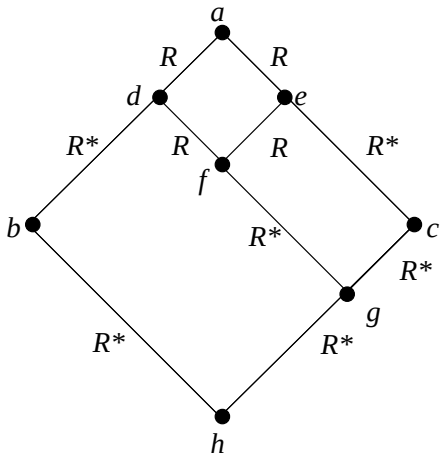

Fig. 9. Completion of the proof of the Diamond Lemma

its final form which is what we present to GROVER. This corresponds to interpreting the diagram as a *trail* of objects, which is the focus of Section 4.

Second, the proof uses transfinite induction twice. The justification for applying transfinite induction is embedded in the assumption that R is well-founded.

Third, the proof twice discovers R-paths through concatenation. This is a good example of something that is quite obvious when rendered visually, but is very hard for an automated reasoning system to prove. The theorem proving system &, which GROVER uses to construct detailed proofs, is able to derive the transitivity of R^* completely automatically. This success is partly dependent on GROVER pruning the set of available hypotheses before & is invoked. The criteria that GROVER applies to prune the hypothesis set at each stage of a proof are discussed in Section 5.2.

4. 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. 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. For example, in the Diamond Lemma, solving for $\mathbf{d}$ means finding a particular object d' such that $R(a, d')$ and $R^*(d', b)$. 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. For example, Figure 9 also contains the assertion $R(\mathbf{d}, \mathbf{f})$, but in the proof we solved for $\mathbf{d}$ before solving for $\mathbf{f}$. Thus, $R(\mathbf{d}, \mathbf{f})$ cannot be considered part of the "definition" of $\mathbf{d}$ (though it later forms part of the definition of $\mathbf{f}$). 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 following:

- One and only one existential object that has not already been solved for,
- Universal objects (such as a, b, and c in Figure 9),
- Existential objects that have already been solved for.

Thus, at the beginning of the proof of the Diamond Lemma, the object $\mathbf{f}$ has no defining properties, since all the formulae containing it also contain $\mathbf{d}$, $\mathbf{e}$, or $\mathbf{g}$, none of which have yet been solved for. From this fact, *existential solve* infers that solving for $\mathbf{f}$

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

is *not* the first thing to be done in the proof of the Diamond Lemma.

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.

4.1. Existential Solve in the Diamond Lemma

To see how *existential solve* works in the Diamond Lemma, we apply it to the diagram in Figure 9. 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.

5. Diagrams as Elisions of Infinitely Many Observations

In Section 2 we presented 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, such as that of the Diamond Lemma, 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 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.

[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.

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 a more general process of associating the diagram terms with the conjecture terms which we have not described here, see [4, 3, 6] for details. Thus, 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.

5.1. The Multiple Peaks Theorem

To see how the interpretation of ellipses works, we present the proof of a generalization of the Diamond Lemma, which we call the Multiple Peaks Theorem. Referring to the diagram in Figure 10, 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. The diagram that expresses the proof of the theorem is shown in Figure 11.

The proof is a straightforward application of mathematical induction. The base case ($n = 0$) follows immediately from GC_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 apply GC_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

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

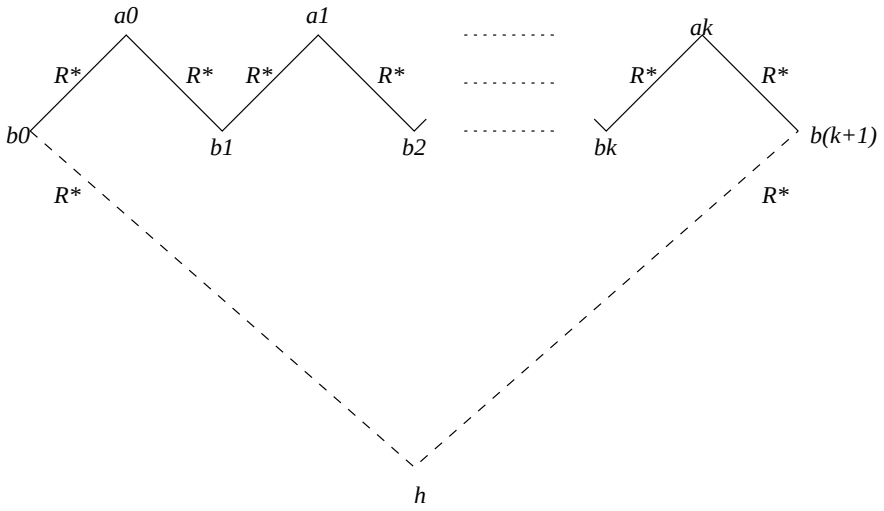

Fig. 10. Graphical Statement of The Multiple Peaks Theorem

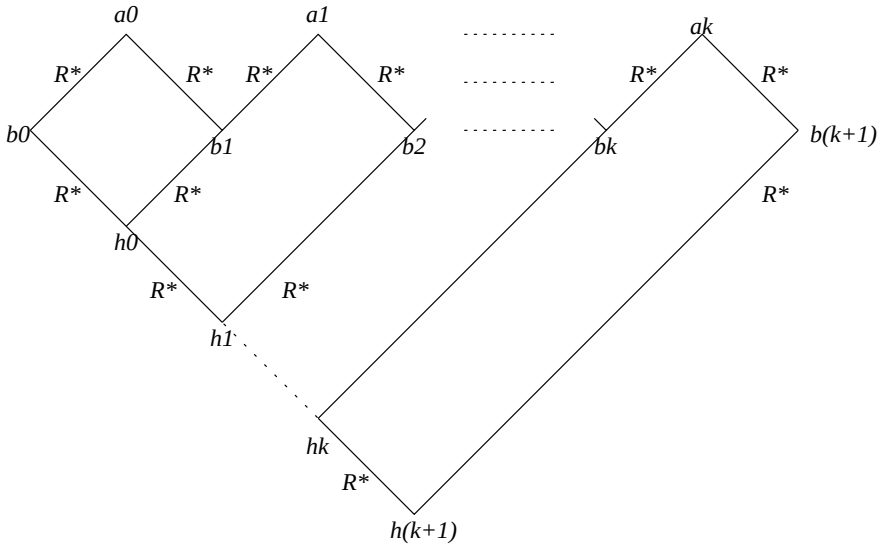

Fig. 11. The Diagram for The Multiple Peaks Theorem

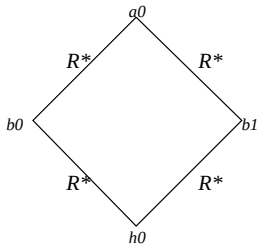

Fig. 12. The Diagram for the Base Case of The Multiple Peaks Theorem

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 11 as representing a *sequence* $t_1 \ldots t_m$ 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 11 with Figures 12 and 13, 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. The universal variables a and b, however, do not correspond to any one of the objects a_i and b_i, but rather to all of them. In order to complete the association, GROVER must have some criterion to establish this correspondence.

One way that we might expect GROVER to do this is to generalize the diagram facts

$$\begin{array}{ll} R^*(a_k, b_k) & R^*(a_k, b_{k+1}) \\ R^*(a_{k+1}, b_{k+1}) & R^*(a_{k+1}, b_{k+2}) \end{array}$$

into a universal formula matching the corresponding hypothesis in the theorem. Such an approach would have to rely on heuristic criteria for generalizing a set of diagram facts into a universal assertion, the goal being to do so when there are no counterexamples.

The notion of counterexample, however, is difficult to formalize. At a minimum, there must be no *explicit* counterexamples, i.e., if the diagram contains the facts $P(0)$, $P(1)$, $P(k)$ and $\neg P(j)$, then we would not want to generalize the first three facts to $\forall x.P(x)$. But what about a situation when, for some object j, the diagram asserts neither $P(j)$ nor $\neg P(j)$? In such a case, we expect that "negation by failure" is the appropriate procedure, which would imply that j should be considered a counterexample. If we could find an appropriate constraint Q that held for 0 and 1 but not for j, then we might generalize the diagram facts to $\forall x.Q(x) \rightarrow P(x)$. In general, however, this approach seems fraught

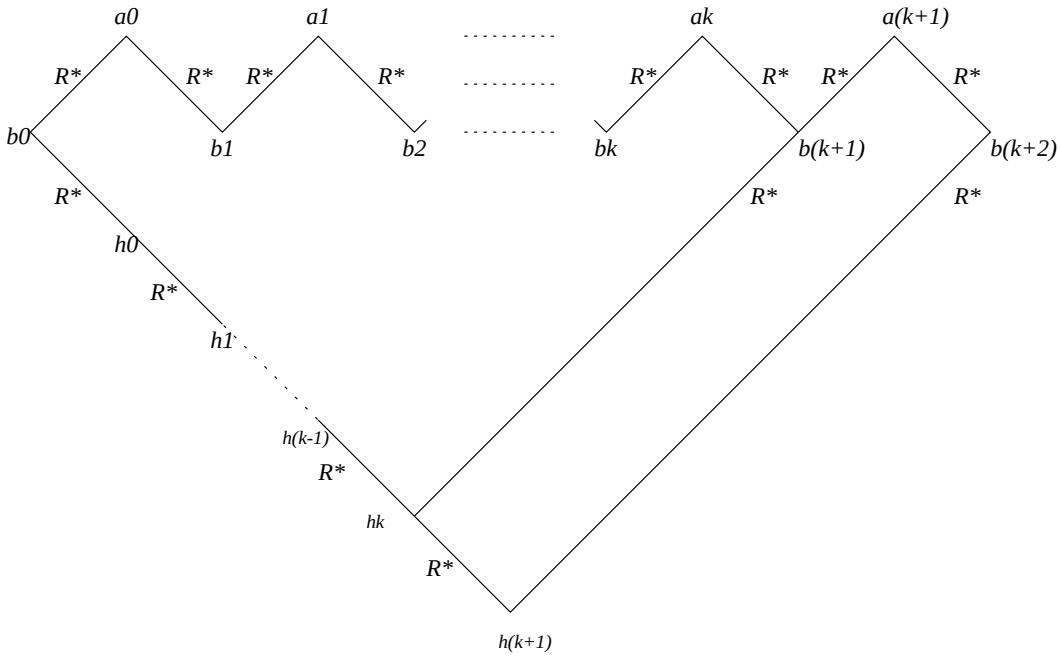

Fig. 13. The Diagram for the Step Case of The Multiple Peaks Theorem

with difficulty since it leads to the thorny problem of analogy matching.

Therefore, instead of generalizing the diagram facts, GROVER takes the opposite approach and tries to instantiate universal variables in the theorem hypotheses so that they match facts in the diagram. GROVER searches the theorem for *spanning hypotheses*, which are hypotheses of the form

$$\forall x.(x \leq n+1 \rightarrow A) \tag{1}$$

where $n+1$ is associated with a diagram *spanning limit*.[5] A spanning limit, by definition, is the subscript of the final term of a sequence in the diagram.

Following this procedure in the Multiple Peaks Theorem, GROVER replaces each spanning hypothesis (1) with the instantiated formulae $A[t/x]$ for all *spanning instances* t. A spanning instance, by definition, is a diagram object participating in one of the diagram

[5]More precisely, n has been associated with a diagram object k, and $k+1$ is a diagram spanning limit.

sequences. In the case of the Multiple Peaks Theorem, these formulae are

$$R^*(a_0, b_0)$$
$$R^*(a_0, b_1)$$
$$R^*(a_1, b_1)$$
$$R^*(a_1, b_2)$$
$$R^*(a_{k+1}, b_{k+1})$$
$$R^*(a_{k+1}, b_{k+2})$$

GROVER is then able to match the hypotheses of both the base case and step case theorems with facts in their respective diagrams.

The next step for GROVER is to develop a strategy leading from theses hypotheses to the conclusions of the respective theorems. There are two major steps to strategy formation:

1. Partially order the diagram facts to be proven as lemmas
2. Determine the hypotheses for each lemma

The first step is performed by the *existential solve* heuristic. In the base case, there is only the existential h_0 to solve for, with defining facts

$R^*(b_0, \mathbf{h_0})$ and $R^*(b_1, \mathbf{h_0})$

In the step case, h_k appears as a universal object since its existence is stated by the induction hypothesis, so it remains only to solve for h_{k+1} with defining facts

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

The second step in creating a proof strategy is to determine the hypotheses for each lemma. The next section describes GROVER's approach to hypothesis selection, using the Multiple Peaks Theorem as an example.

5.2. 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.

Example: Choosing Relevant Hypotheses in the Multiple Peaks Theorem

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.

6. Diagram Idioms: Visualization and Abstraction

In section 4.1 we asserted that the collection of formulae on page 11 were represented explicitly in the diagram of figure 9. 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, and somewhat more subtle than the example of the Diamond Lemma might suggest.

One of the key components of &/GROVER is a graphical editor called DEGAS[6]. DEGAS is a rather conventional graphical editor, with tools allowing the drawing of lines, ellipses, 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).[7] The G-file is a generic textual representation of the diagram structure, irrespective of any semantics that we associate with the diagram. The G-file identifies the objects, object types (e.g., circle, triangle, etc.), object attributes (e.g., shading, etc.), arcs, arc types (e.g., directed, undirected etc.), and arc attributes (e.g., dotted, smooth, etc.) found in the diagram. This notation allows us to describe the key features of the diagram in a textual notation and at a high-level of abstraction.

The G-file representation of the Diamond Lemma diagram of figure 9 is shown in figure 14.

The G-file language has statements which assert the existence of graphical structures in the diagram. For example, the statement `arc(arc1)` says that there is an arc in the diagram with the symbolic name `arc1`. Other types of objects that our system can currently handle include arrows (directed arcs), dots and closed figures (circles).

Objects may be related together in various ways. For example the assertion `in(dot2,circle1)` asserts that `dot2` appears within the closed figure `circle1`. Other relationships that may hold between objects include *within* (one closed figure completely enclosing another) and *overlap*.

Objects may also have attributes. Arcs and arrows, for example, have `end-points` which are usually (but not necessarily) dots. A special type of attribute is a `label`, which indicates text associated with a given object. For example `label(arc1,"R")` indicates that `arc1` has the textual label `"R"` associated with it. The presence of quotation marks indicates that the text is to be parsed as a formula of the language of the theorem prover (currently all labels must be so treated). The diagram itself is an object which may be labeled, typically with the conjecture that it supports.

Notice that the G-file contains much less information that the corresponding diagram. For example, no information regarding the relative placements of points, and the placement of the labels relative to the objects in the diagram is present in the G-file.

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

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

```
diagram(diamond_lemma).
label(diamond_lemma, "WF_R & LC_R -> GC_R").

dot(dot1).
label(dot1, "!a").

dot(dot3).                          dot(dot2).
label(dot3, "e").                   label(dot2, "d").

dot(dot4).
label(dot4, "f").
hint(dot4, "LC_R").

dot(dot5).                          dot(dot8).
label(dot5, "!c:R*(!a,!c)").        label(dot8, "!b:R*(!a,!b)").

dot(dot6).                          dot(dot7).
label(dot6, "g").                   label(dot7, "h").
hint(dot6, induction, "!e").        hint(dot7, induction, "!d").

arc(arc1).                          arc(arc2).
label(arc1, "R").                   label(arc2, "R").
end_points(arc1, dot1, dot2).       end_points(arc1, dot1, dot3).

arc(arc3).                          arc(arc4).
label(arc3, "R").                   label(arc4, "R").
end_points(arc3, dot3, dot4).       end_points(arc4, dot2, dot4).

arc(arc5).                          arc(arc6).
label(arc5, "R*").                  label(arc6, "R*").
end_points(arc5, dot2, dot8).       end_points(arc6, dot4, dot6).

arc(arc7).                          arc(arc8).
label(arc7, "R*").                  label(arc8, "R*").
end_points(arc7, dot3, dot5).       end_points(arc8, dot6, dot7).

arc(arc9).                          arc(arc10).
label(arc9, "R*").                  label(arc10, "R*").
end_points(arc9, dot8, dot7).       end_points(arc10, dot5, dot6).
```

Fig. 14. The G-file representation of the Diamond Lemma Diagram

Nor for example, is the fact that the arc from a to d and the arc from d to b are collinear. The G-file notation is sufficient only to describe diagrams up to topological equivalence. We believe that, in general, mathematical diagrams do not rely on detailed positional information of objects within the diagram, and that this abstract notation is a useful intermediate step in the "translation" between diagram and logical formulae.

The use of a graphical editor which is able to produce a representation of the diagram at this very high 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 points in a line a really 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.

6.1. 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. We call this phase *geometry to logic*, and describe it section 6.2. The global analysis phase, which we call *verify logic*, detects larger constructions in the diagram. This process is described in section 6.3.

6.2. 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 15), 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 15 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 enclosed 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.

6.3. 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 diagram that supports the Diamond Lemma (figure 9) contains no larger scale structures, and this phase is a no-op in that proof. In the next section we describe the proof of the Schröder-Bernstein Theorem whose diagram does contain such structures.

The *verify logic* phase of the diagram 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.

6.3.1. Definition by Cases

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 [5]), 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.

This example explains what otherwise might be considered an odd choice of name: *verify logic*. We were led to look for these higher-level structures when we began our analysis of the diagram for the Schröder-Bernstein Theorem (described in section 6.4). We were faced with an L-file representation that contained an apparent contradiction. We assume that diagrams are not intended to be contradictory, and therefore that some further processing is needed to explain and remove the apparent contradiction. Initially we saw this task as verifying and repairing the L-file, which has flaws because it is produced by local, bottom-up analysis. However, we have come to see this as the more general task of discovering how the different parts of the initial L-file fit together to form larger-scale structures.

In section 6.4 we describe the proof of the Schröder-Bernstein Theorem in detail, and describe the verify logic critics that participate in this proof (including *definition by cases*) in detail.

6.4. Example: The Schröder-Bernstein Theorem

The Schröder-Bernstein Theorem is a theorem from the theory of functions which concerns the way in which the "size" of sets can be measured. Measuring the size of a finite set is quite straightforward, you count the elements, but it is harder when the sets are infinite. The solution to the comparison problem is to think in terms of functions between sets. Two sets are the same size if there is an association of each element of one set with a unique element of the other and vice versa. Such an association is called a $Bijection$. Similarly, set B is at least as big as set A if there is an association of every member of A with some unique member of B (but not necessarily the reverse); such an association is called an $Injection$. The Schröder-Bernstein Theorem demonstrates that if set A is at least as big as set B and set B is at least as big as set A (in this sense), then sets a and b are the same size (again, in this sense). Formalized, the theorem is stated:

Theorem 6.1 (Schröder-Bernstein.)
$\forall f, g, A, B.Injection(f, A, B) \wedge Injection(g, B, A) \rightarrow \exists h.Bijection(h, A, B)$

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 15 illustrates this strategy. We believe that mathematicians will readily understand this diagram as representing the intended strategy, and also that the diagram is a natural expression of the strategy outlined above. That is, we believe that a mathematician asked to represent the intended construction will draw a diagram very much like this one (with one likely difference, see section 6.2).

The diagram contains objects **h**, $\mathbf{A_1}$, $\mathbf{A_2}$ and **C** whose existence must be proved, but in addition the diagram contains the graphical representation of the *definition* of these objects. GROVER uses these definitions to simplify the proofs that & must carry out. Rather than synthesize objects with the desired properties, &'s task is to verify that particular objects have desired properties. Not only does this simplify the work that the system must do, but we also believe that this is the intuitive use to which the

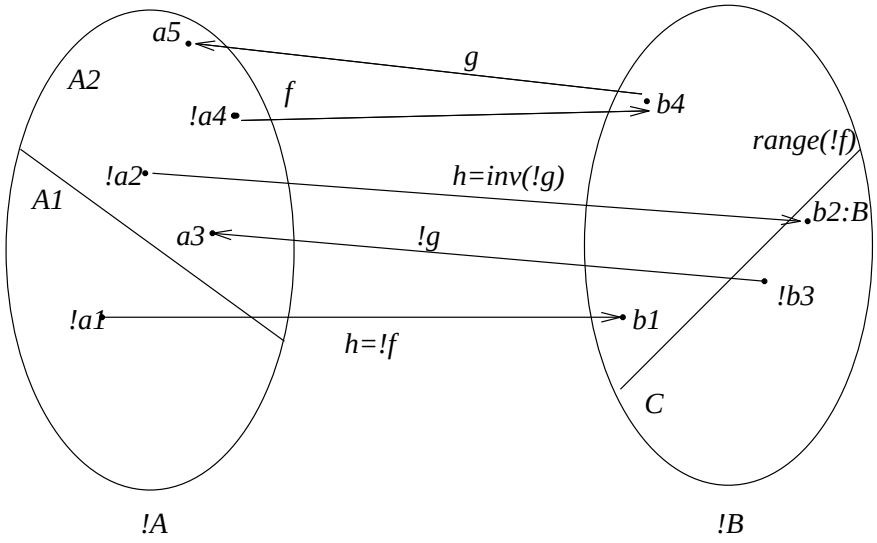

Fig. 15. The Diagram for the Schröder-Bernstein Theorem

diagram is being put. We see the diagram as explicitly stating the definitions of (some of) the existential objects that it contains.

Consider, for example, the treatment of the existential **h**. The desired strategy suggests that we recognize that it is being defined by cases, and that membership in $\mathbf{A_1}$ or $\mathbf{A_2}$ is the crucial feature which determines whether **h** acts like f or like g^{-1}. The information recognized in the diagram is sufficient to produce a set term that which represents **h** completely. Similarly, the recognition of the fact that $\mathbf{A_2}$ is formulated by a closure operation is sufficient for GROVER to guess an appropriate set term that defines $\mathbf{A_2}$, namely the intersection of all sets that are closed under the operation.

GROVER's ability to infer the sets defining **h** and $\mathbf{A_2}$ from the diagram greatly simplifies the proofs of the conjectures that & is asked to construct. However, the proofs that **h** and $\mathbf{A_2}$ have the required properties must still be carried out. This involves verifying that **h** (as defined by cases on $\mathbf{A_1}$ and $\mathbf{A_2}$) is a function, which follows from the fact that $\mathbf{A_1}$ and $\mathbf{A_2}$ are disjoint and also depends on the fact that f and g^{-1} are themselves functions. It is also necessary to prove that $\mathbf{A_2}$ (as defined) has the properties of containing $g^{-1}(B - range(f))$ and being closed under $g \circ f$. This fact does not immediately follow from the fact that $\mathbf{A_2}$ is the intersection of all sets which have these properties (that is, for an arbitrary property P it does not follow that the intersection of all sets with the property P is itself a set which enjoys the property P).

After local analysis of the diagram of figure 15 using *geometry to logic*, the following

formulae are produced:

1. $Disjoint(range(f), \mathbf{C})$
2. $Disjoint(\mathbf{A1}, \mathbf{A2})$
3. $A = (\mathbf{A1} \cup \mathbf{A2})$
4. $B = (range(f) \cup \mathbf{C})$
5. $\mathbf{h} = f$
6. $\mathbf{h} = g^{-1}$
7. $Function(f, \mathbf{A1}, range(f))$
8. $Function(f, \mathbf{A2}, range(f))$
9. $Function(g, range(f), \mathbf{A2})$
10. $Function(g, \mathbf{C}, \mathbf{A2})$
11. $Function(\mathbf{h}, \mathbf{A1}, range(f))$
12. $Function(\mathbf{h}, \mathbf{A2}, \mathbf{C})$
13. $a1 \in \mathbf{A1}$
14. $a2 \in \mathbf{A2}$
15. $a4 \in \mathbf{A2}$
16. $b3 \in \mathbf{C}$
17. $\langle a1, \mathbf{b1} \rangle \in f$
18. $\langle a1, \mathbf{b1} \rangle \in \mathbf{h}$
19. $\langle a2, \mathbf{b2} \rangle \in g^{-1}$
20. $\langle a2, \mathbf{b2} \rangle \in \mathbf{h}$
21. $\langle a4, \mathbf{b4} \rangle \in f$
22. $\langle b3, \mathbf{a3} \rangle \in g$
23. $\langle \mathbf{b4}, \mathbf{a5} \rangle \in g$
24. $\mathbf{a3} \in \mathbf{A2}$
25. $\mathbf{a5} \in \mathbf{A2}$
26. $\mathbf{b1} \in range(f)$
27. $\mathbf{b2} \in B$
28. $\mathbf{b2} \in \mathbf{C}$
29. $\mathbf{b4} \in range(f)$

Notice that the formulae within this L-file representation of the diagram are inconsistent, for example asserting different domains and ranges for the defined function **h** in formulae 11 and 12. The intention is that the formulae from the diagram are to be interpreted conjunctively, and therefore this mismatch between the different formulae must be dealt with. This situation arises because each formula is constructed in isolation. In practice the diagrammatic structures interact to form larger patterns. In the next phase of the processing of the diagram, a global analysis of the L-file is performed in order to create a coherent L-file which consolidates the information produced by the local analysis

of the diagram. In the next sections we describe the verify logic critics which detect the different high-level structures in the L-file. We begin with the *definition by cases* critic.

6.4.1. Definition by Cases in the Schröder-Bernstein Theorem

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. The critical sub-diagram is displayed in figure 16.

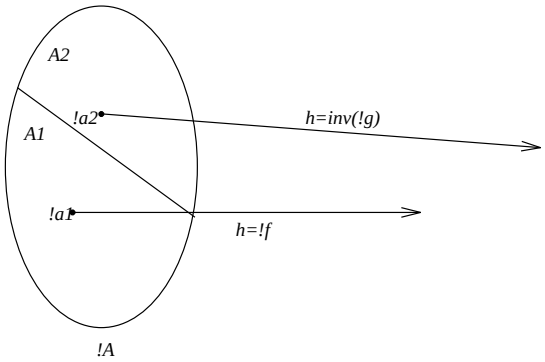

Fig. 16. Definition by Cases

The `definition by cases` critic removes the two equalities from the L-file, along with every formula involving the endpoints of the arrows labeled by these equalities (formulae 5, 6, 13, 14, 17– 20 and 26– 28). The formulae:

$$\forall x, y.x \in \mathbf{A_1} \rightarrow (\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in f)$$
$$\forall x, y.x \in \mathbf{A_2} \rightarrow (\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in g^{-1})$$

are added in their place.

Since $\mathbf{A_1}$ and $\mathbf{A_2}$ are known to be disjoint (by the fact that they are distinct regions of a partition), the definition:

$$\begin{aligned}\mathbf{h} \quad = \quad & \{x | x \in f \wedge \exists y, z.x = \langle y, z\rangle \wedge y \in \mathbf{A_1} \wedge \langle y, z\rangle \in f\} \cup \\ & \{x | x \in g^{-1} \wedge \exists y, z.x = \langle y, z\rangle \wedge y \in \mathbf{A_2} \wedge \langle z, y\rangle \in g\}\end{aligned}$$

is inferred. A hint that **h** is defined to be this set term is added to the collection of hints associated with the diagram. This hint is used when the individual goals of the strategy are constructed.

Finally, this critic uses formula 4 to consolidate $Function(g, range(f), \mathbf{A_2})$ and $Function(g, C, \mathbf{A_2})$ into the single formula $Function(g, B, \mathbf{A_2})$, and similarly uses formula 3 to modify the domains of $\mathbf{h}$ and f.

The resultant L-file is:

1. $\forall x, y.(x \in \mathbf{A_1} \rightarrow (\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in f))$
2. $\forall x, y.(x \in \mathbf{A_2} \rightarrow (\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in g^{-1}))$
3. $Disjoint(range(f), \mathbf{C})$
4. $Disjoint(\mathbf{A_1}, \mathbf{A_2})$
5. $A = (\mathbf{A_1} \cup \mathbf{A_2})$
6. $B = (range(f) \cup \mathbf{C})$
7. $Function(f, A, range(f))$
8. $Function(g, B, \mathbf{A_2})$
9. $Function(\mathbf{h}, A, B)$
10. $a4 \in \mathbf{A_2}$
11. $b3 \in \mathbf{C}$
12. $\langle a4, \mathbf{b4}\rangle \in f$
13. $\langle b3, \mathbf{a3}\rangle \in g$
14. $\langle \mathbf{b4}, \mathbf{a5}\rangle \in g$
15. $\mathbf{a3} \in \mathbf{A_2}$
16. $\mathbf{a5} \in \mathbf{A_2}$
17. $\mathbf{b4} \in range(!f)$

6.4.2. Function Chains

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 15 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 specific formulae concerning 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 17.

The result of applying the function chains critic to the formulae just mentioned 11, 13 and 15, is the new formula:

$$\forall b3.(b3 \in \mathbf{C} \rightarrow \forall x.(\langle b3, x\rangle \in g \rightarrow x \in \mathbf{A2}))$$

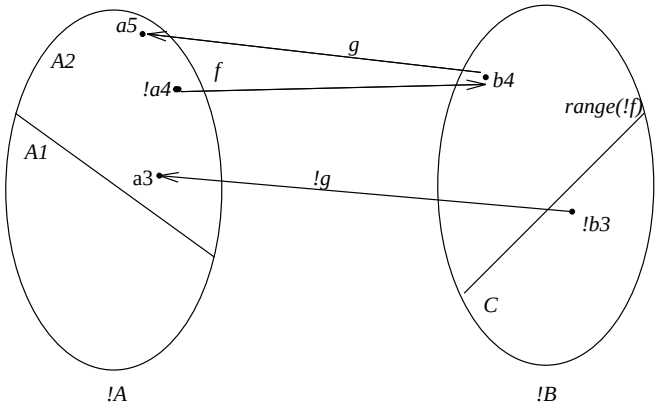

Fig. 17. Function Chains

The same critic notes that formulae 10, 12 14 and 16 (in the revised formula list) 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.

6.4.3. Closure Recognition

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 17.

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:

$$\forall s.((\forall a4.(a4 \in s \rightarrow (\forall x, y.((\langle x, y\rangle \in g \wedge \langle a4, x\rangle \in f) \rightarrow y \in s))))\wedge$$
$$\forall b3.(b3 \in \mathbf{C} \rightarrow \forall z.(\langle b3, z\rangle \in g \rightarrow z \in s))) \rightarrow$$
$$\mathbf{A_2} \subseteq s$$

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 term that is produced as the definition of $\mathbf{A_2}$ is:

$$\{x|(x \in A\wedge$$
$$(\forall a.(((\forall b_1.((b_1 \in C) \rightarrow (\forall a_1.((\langle b_1, a_1\rangle \in g) \rightarrow (a_1 \in a)))))\wedge$$
$$(\forall a_2.((a_2 \in a) \rightarrow (\forall b_2.(\forall a_3.((((\langle a_3, b_2\rangle \in g) \wedge (\langle a_2, a_3\rangle \in f)) \rightarrow (b_2 \in a)))))))) \rightarrow$$
$$(x \in a))))\}$$

6.4.4. Function Domains

The final critic used in this 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 15, 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 $Function(f, A, B)$.

We note that, 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 under-generalize, 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. We observe here that a subsequent step in the construction of a strategy based on the diagram for the Schröder-Bernstein Theorem (associate with diagram) would fail if the domain and range were not correctly inferred by this critic. This failure could then be used to prompt the critic to suggest alternative

domain and range assignments, and therefore to arrive at the intended result by trial-and-error.

6.4.5. The L-file

The L-file which results from the completed process of the diagram by GROVER is:

1. $Function(f, A, B)$
2. $Function(g, B, A)$
3. $Disjoint(\mathbf{C}, range(f))$
4. $B = \mathbf{C} \cup range(f)$
5. $\forall x.x \in \mathbf{C} \rightarrow \forall y.(\langle x, y\rangle \in g \rightarrow y \in \mathbf{A_2})$
6. $\forall x.x \in \mathbf{A_2} \rightarrow \forall y, z.((\langle x, y\rangle \in f \wedge \langle y, z\rangle \in g) \rightarrow z \in \mathbf{A_2})$
7. $\forall a.(\forall x.x \in \mathbf{C} \rightarrow \forall y.(\langle x, y\rangle \in g \rightarrow y \in a) \wedge$
   $\forall x.x \in a \rightarrow \forall y, z.((\langle x, y\rangle \in f \wedge \langle y, z\rangle \in g) \rightarrow z \in a)$
   $\rightarrow \mathbf{A_2} \subseteq a)$
8. $Disjoint(\mathbf{A_1}, \mathbf{A_2})$
9. $A = \mathbf{A_1} \cup \mathbf{A_2}$
10. $Function(\mathbf{h}, A, B)$
11. $\forall x, y.x \in \mathbf{A_1} \rightarrow (\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in f)$
12. $\forall x, y.x \in \mathbf{A_2} \rightarrow (\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in g^{-1})$

We also have hints establishing the identity of $\mathbf{h}$ and $\mathbf{A_2}$ as determined by the appropriate critics.

6.5. Existential Solve in the Schröder-Bernstein Theorem

Once no *verify logic* critics can be used, the L-file is complete and the next phase of the processing begins. The *existential solve* heuristic, described in section 4.1, is used to create the strategy to complete the proof.

One additional heuristic is used within *existential solve* in our proof of the Schröder-Bernstein Theorem. Applying the *existential solve* strategy as previously described to the formulae above leads to an impasse, since there is no way to order the formulae containing $\mathbf{h}$ and $\mathbf{A_1}$. This is not in general a problem, since if there is no reason to impose an ordering on two classes of formulae, they are left unordered in the partition; but in this case a special situation arises. The formulae that remain are:

8. $Disjoint(\mathbf{A_1}, A_2)$
9. $A = \mathbf{A_1} \cup A_2$
10. $Function(\mathbf{h}, A, B)$
11. $\forall x.x \in \mathbf{A_1} \rightarrow (\forall y.\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in f)$
12. $\forall x.x \in A_2 \rightarrow (\forall y.\langle x, y\rangle \in \mathbf{h} \leftrightarrow \langle x, y\rangle \in g^{-1})$

Formulae 11 and 12 are constructed by the definition by cases critic as the two halves of the definition of existential **h**. GROVER's analysis of the defining properties of the exsitential objects must take this fact into account in addition to the usual heuristics concerning the number of existential and universal objects in the formulae. The definition by cases formulae are considered by GROVER to form the defining properties of **h**, long with the formula 10. This overriding consideration leads to these formulae being grouped into a single lemma by GROVER.[8] The lemma formed by these formulae contain both **h** and $\mathbf{A_1}$ as existential objects, while the remaining formulae contain just $\mathbf{A_1}$. The result is the following partition of the L-file (using the formulae numbered as above):

$\Delta_0 = (1,2)$
$\Delta_1 = (3,4)$
$\Delta_2 = (5,6,7)$
$\Delta_3 = (8,9)$
$\Delta_4 = (10,11,12)$

7. 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.

Although the examples we have investigated provide evidence of these uses of diagrams in mathematical proof, we consider it essential to study a larger number of diagrams from a variety of mathematical areas. In addition, it is important to consider the way in which diagrams can be treated as dynamic objects. When drawn by humans in the process of communicating a proof, diagrams are elaborated and marked up as the proof proceeds. This dynamic unfolding conveys meaning, and we would like to explore it through the interaction of the user with the theorem proving system.

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

[8] The same effect would be obtained if GROVER had produced the definition by cases as a single formula consisting of the conjunction of 11 and 12.

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

1963

[1] G.F. Simmons. *An Introduction to Topology and Modern Analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill, 1963.

1980

[2] G. Lakoff and M. Johnson. *Metaphors We Live By*. University of Chicago Press, 1980.

1992

[3] D. Barker-Plummer and S.C. Bailin. Graphical theorem proving: An approach to reasoning with the help of diagrams. In *Proceedings of the European Conference on Artificial Intelligence (ECAI-92), Vienna, Austria*, pages 55–59. John Wiley and Sons, August 1992.

[4] D. Barker-Plummer and S.C. Bailin. 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*, March 1992.

1993

[5] J. Barwise. Heterogeneous reasoning. In G. Allwein and J. Barwise, editors, *Working Papers on Diagrams and Logic*, pages 1–13. Indiana University Logic Group, 1993.

1996

[6] D. Barker-Plummer, S.C. Bailin, and S.M.T Ehrlichman. Diagrams and mathematics. In B. Selman and H Kautz, editors, *Proceedings of the 4th International Conference on Artificial Intelligence and Mathematics*, 1996.