From: AAAI Technical Report SS-92-02. Compilation copyright © 1992, AAAI (www.aaai.org). All rights reserved.

Proofs and Pictures
Proving the Diamond Lemma with the GROVER Theorem Proving System*

Dave Barker-Plummer
Computer Science Program,
Swarthmore College,
Swarthmore, Pennsylvania, 19081

plummer@cs.swarthmore.edu

Sidney C. Bailin
CTA INCORPORATED,
6116 Executive Blvd, Suite 800
Rockville, Maryland, 20852

sbailin@cta.com

January 14, 1992

Key words. Automated reasoning, graphical theorem proving, proof strategies.

1 Introduction

In this paper we describe an automated reasoning system, called GROVER, which has a graphical component. In addition to the theorem to be proved, GROVER is presented with a diagram which would provide to the human reader the essence of the proof. GROVER consists of two parts: the graphical interface which is the subject of this paper, and an underlying theorem prover, called &. The graphical subsystem constructs a strategy on the basis of the information extracted from the diagram; & is then called upon to prove the subgoals in this strategy.

Diagrams provide two types of information: *how* to conduct the proof, and intermediate steps on the way which indicate *why* the theorem is true. The former type of information may suggest proof strategies to the theorem prover, for example a proof by induction, which is often indicated by the appearance of ellipses in a diagram. The latter type of information may suggest intermediate lemmas for the theorem prover to prove. In this way the diagram indicates "stepping stones" for the search of the main proof. The problem is reduced to finding a proof of the main theorem by combining proofs of the intermediate lemmas.

2 Key Result

We have used GROVER to prove the Diamond Lemma, a non-trivial theorem in the theory of well-founded relations. This theorem is stated:

$$LCR_R \wedge WF_R \rightarrow GCR_R$$

where LCR_R says that the relation R has the local Church-Rosser property, and GCR_R says that the relation R has the global Church-Rosser property[1]. The local Church-Rosser property says that for all a, b, and c, if aRb and aRc then there is a d such that bRd and cRd. In other words, whenever some diagram contains a construction matching the non-dashed lines in the diagram (figure 1), then the dashed lines may be added to that diagram.

The global Church-Rosser property is similar to the local property except that R^*, the transitive closure of R, is used instead of R.

*This material is based upon work supported by the National Science Foundation under award number ISI-8701133.

[1]For technical reasons related to the nature of the underlying prover, we actually consider the relation $R|x$ — R restricted to a given set x. For details the interested reader is referred to a longer version of this paper available from the authors on request.

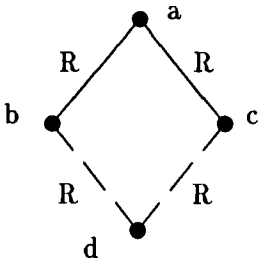

Figure 1: Local Church-Rosser Property

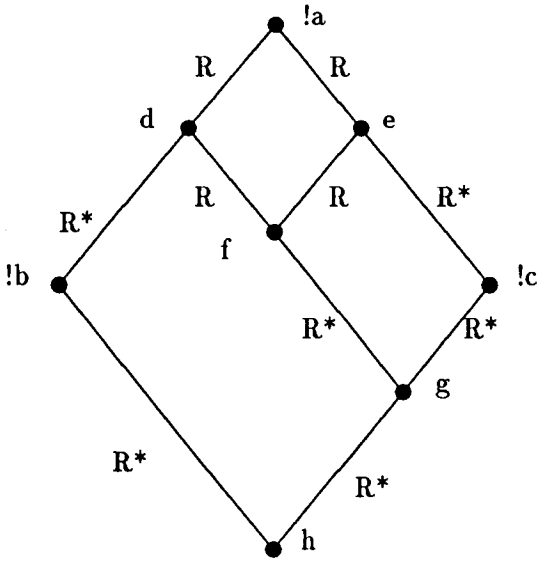

Figure 2: Diagram for the Diamond Lemma

WF_R asserts that the relation R is well-founded, i.e., there is no infinite sequence $a_1 R a_2 R \cdots$.

We use this theorem and the "canonical" diagram for the Diamond Lemma as a running example. This diagram is shown in figure 2. The underlying theorem prover, &, is not able to prove this theorem without the information provided by the graphical interface.

3 Interpreting The Diagram

For the current implementation of GROVER we have concentrated on the problem of extracting intermediate steps from the diagram. In this view, diagrams consist of a set of "facts" which collectively justify the theorem to be proved. Some of these facts may immediately imply the desired conclusion, others may represent hypotheses and still more may represent intermediate assertions, provable from the hypotheses and useful in the proof of the conclusion. The interpretation of the diagram must therefore consist of two parts, first the extraction of these facts from their graphical representation, and second the organization of these facts into a sequence of lemmas to prove.

The current version of GROVER does not read diagrams directly. Instead we have devised a language for expressing geometric relationships between objects. A specification in this language serves as an ASCII representation of the diagram.

In our language we can define objects of various categories, for example, the statement `arc(arc1)` asserts the existence of 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).

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. The rules of the expert system are intended to capture the usual practice of mathematical diagrams. The analysis of the diagram proceeds in a bottom-up fashion. First the individual objects in the diagram are examined. Various types of labels may be associated with objects, corresponding to the practices that we have encountered. For example, the label "`c : aR*c`" indicates that the la-

beled object is called c and that it has the property that aR^*c. Other label forms that are allowed are simple names, and 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, 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 aRb. This is because, in the language of our prover, `"R"` has the right form to be a predicate symbol. Similarly, our expert system interprets a closed figure completely within another as the $\subseteq$ relation, and a dot within a closed figure as the $\in$ relation. 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. So an arrow labeled f with end points labeled a and b is interpreted as representing the equation $f(a) = b$.

Individual objects in the diagram may be universal or existential, corresponding to quantifiers in the formal logic. In figure 2, the objects a, b, and c are universal, indicated by the ! prefix, which is the symbol for universal quantification in our system. Existential objects may have hints associated with them, indicating which formulae are to be used in proving the existence of this object. For example, in our diagram for the diamond lemma we associate the hint `use(induction)` with the objects g and h in the diagram, and the hint `use("LCR_R")` with the object labeled f. These hints are instructions to the underlying theorem prover about which hypotheses to consider in the proof (note, however, that the induction hypothesis is not part of the original theorem; & preprocesses the theorem to derive an induction hypothesis.). We view hints as undesirable — we would much rather have the prover complete the proof without them — but we do view them as defensible. In addition to the diagram, the proof of a theorem is often accompanied by a narrative in the form of either accompanying text (in a book), or spoken narrative (in a lecture context). In either case, repeated reference to the diagram might be made, and we believe that in almost all plausible cases, the hints that we have given would be mentioned.[2]

Repeated references to the diagram also indicate a temporal component to the reader/listener. We know from the "standard" proof of the Diamond Lemma that the existence of d and e are proved first, and then the existence of f follows, for example. We must recover this temporal component in the creation of a strategy for carrying out the proof. This is described in the next section.

4 Using The Diagram

When we are asked to prove a conjecture of the form $\Gamma \vdash \Sigma$ with the aid of a collection of diagram assertions, Δ, we proceed in two phases. The first is to verify that the diagram does in fact entail the conclusion of the theorem by proving $\Delta \vdash \Sigma$. In general we expect that the diagram contains more information than the hypotheses of the main theorem, so this proof should be easier to produce than the original goal.[3] However, we have assumed more in the proof of the conclusion than we are permitted by the theorem, so we must discharge the assumption of the diagram formulae. To do this we must prove each of the members of Δ from the set of hypotheses, Γ. Once a given member of Δ has been proved from Γ, it may be added as a new hypothesis for the proofs of the subsequent members of Δ. We attempt to find a partial ordering of the members of Δ so that we can *prove*, and then *use*, each formula.

Prove Conclusion. We begin by attempting to prove the conclusion of the conjecture from the collection of diagram assertions, $\Delta \vdash \Sigma$ as above. A problem immediately arises here, namely that the theorem and the diagram may be expressed in different vocabularies. For example, the Diamond Lemma is (after preprocessing) a conjecture concerning three universal objects $\forall x, y, z....$ (see section 5). These universal objects have counterparts in the diagram, namely a, b and c. By the time the diagram is called into play, the leading universal quantifier may have been eliminated in a $\forall$-Introduction step. This inference rule replaces the bound variables with ar-

[2]The conjectures that the underlying theorem prover is called upon to prove are simplified as a result of these hints. However, those theorems remain highly non-trivial and require a very powerful underlying prover to prove them.

[3]It is not always the case that adding extra hypotheses makes a conjecture easier to prove. Additional formulae can often be the cause of a combinatorial explosion.

bitrarily chosen constant terms. Therefore, before beginning any proof GROVER must determine an association between universal objects in the diagram and the universal constants in the sequent.

The correct association is determined by generating associations and then determining whether, given this association, the hypotheses of the conjecture are represented in the diagram.

Developing a Strategy. Having proved the conclusion theorem, the next stage is to discharge the assumption of the diagram formulae. Some of these formulae have already been justified as hypotheses when the universal objects in the diagram were associated with those in the conjecture. The remaining formulae must be proved from this set. As observed above, once a formula has been proved it is available as a hypothesis in the proof of subsequent members of Δ.

We have developed a strategy, called *existential solve* which orders the formulae in Δ according to the following intuition: we have to prove the existence of each existential object in the diagram, and this is effectively to solve for unknowns. We begin by collecting together formulae on the basis of the number of existential objects that they contain. We seek to solve for the "most constrained" existential first, so we consider first those formulae that contain only one existential object. We collect together all those formulae that share the same existential, and we plan to prove the conjunction of these formulae in a single proof — thereby taking into account all of the constraints on the existential object in the proof.

Once we have proved the collection of formulae involving a given existential object, we know that this object exists. So, for the purposes of ordering the remaining formulae, the existential object is considered as a universal one. Any formula in the diagram that contained more than one existential, including the one that was just solved for, now contains fewer existentials and is therefore more constrained. In this way we proceed inductively to address each existential in the diagram.

Prove Using Strategy. Once GROVER has constructed a partial ordering on subsets of the diagram formulae, we set out to prove these in the suggested order. It is possible that there is no total ordering on the existentials in the diagram, but this is not required, only that some partial ordering exists. If two members of Δ are ordered with respect to the partial ordering, then the intention is to use the earlier formula as a hypothesis in the proof of the later one. Initially we have as hypotheses only those formulae which appear in the theorem, i.e., Γ. The members of the first class of formulae to be proved are conjoined, and an existential quantifier is "wrapped around" this conjunction; the bound variable is substituted for the existential object in the diagram. We then pass this formula to the underlying theorem prover, &, for proof. If this proof succeeds, the formula is added to the hypotheses for proving all subsequent conjectures.

For example, in the Diamond Lemma, the first class in the ordering contains the formulae aRd and dR^*b. We conjoin these formulae, and bind the existential object to obtain the conjecture: $\exists x.(aRx \wedge xR^*b)$. Once this conjecture has been proved, we are allowed to assume aRd and dR^*b for subsequent proofs, where d is now a constant.

5 The Diamond Lemma

The statement of the diamond lemma is

$$LCR_R \wedge WF_R \rightarrow GCR_R$$

This statement is *not* the theorem that the diagram justifies. It is necessary to first preprocess the theorem. The diagram really justifies the theorem

$$\forall yz.((x \neq y \wedge x \neq z) \wedge (xR^*y \wedge xR^*z) \rightarrow \exists w.(yR^*w \wedge zR^*w)$$

under the assumptions of the local Church-Rosser property, and an induction hypothesis that the property holds for all x' that are "R-less" than x. The induction hypothesis is justified by the well-foundedness of R.

In the current implementation, GROVER must be explicitly told to apply the transfinite induction rule, and then to use the information in the diagram. This is not desirable, but we believe that a closer coupling of the graphical system and & would enable us to take these steps automatically. Once this preprocessing has been carried out, the graphical subsystem is invoked and the remaining proofs are performed automatically.

Conjecture	Time	Subgoals
conclusion	14.5mins	166
D-exists	18.5mins	515
E-exists	18.5mins	515
F-exists	37.5secs	20
G-exists	2hrs 6mins	672
H-exists	8hrs	1208

Figure 3: Statistics for the Diamond Lemma

1. The proof of the conclusion

$$\exists x.bR^*x \wedge cR^*x$$

follows from the diagram formulae, with h as the required term. The system must derive and use the transitivity of R^* to show this.

2. The proof of the existence of d, namely

$$\exists x.aRx \wedge xR^*b$$

follows since $a \neq b$. The proof of the existence of e is similar. These two subproofs are not ordered by the existential solve heuristic, and are thus proved effectively in parallel, i.e. the conclusion of the existence of d subproof is not added to the hypotheses for the existence of e subproof.

3. The existence of f follows from the existence of d and e and the local Church-Rosser property. For this proof, the formulae just proved concerning d and e are added to the hypothesis set Γ.

4. The existence of g and h follow from the induction hypothesis, applied to the diamonds below e and below d respectively. For the proof of g, the formulae just proved concerning f are added to the hypotheses, and for the proof of h the formulae concerning g are available.

Figure 3 shows the times required for the subtheorems that the system must prove. These times were obtained running the system on a SUN 3/50. The entire system is implemented in the Quintus dialect of Prolog (version 3.1).

6 On Hints

In a typical proof, not all hypotheses of the theorem are used in every lemma. One criterion for the elegance of a proof is that each hypothesis is used only once, so that its rôle in the proof is clear. GROVER allows the user to annotate the diagram with *hints* which specify the hypotheses needed to prove individual diagram assertions. The hints correspond to speech acts that are common in informal mathematics: when we walk through a proof, with or without the help of a diagram, we verbally invoke specific hypotheses at appropriate points. The rôle of hints in the proof of the Diamond Lemma is only to determine which of the hypotheses of the global proof to include in the hypotheses of the individual subproofs. Without such hints, & would be faced with a combinatorial explosion of decomposing irrelevant formulas and constructing irrelevant instantiations. In the current implementation, GROVER adds a global hypothesis to the hypothesis of the individual subproofs only if the hypothesis is specifically invoked by using a hint.

7 The Dynamic Nature of Diagrams

Our current approach requires the user to develop the diagram prior to invoking the prover. Following construction of the diagram, the user invokes &; GROVER is then invoked by & in response to the *use diagram* command, which the user enters. GROVER retrieves the geometry file, which (for the time being) serves as our graphical representation of the diagram.

In this approach we are using a static object—a diagram of elements and their relations—to express dynamic knowledge of the intended proof strategy. This method of invoking GROVER is not the best possible: it does not reflect the use of diagrams in informal mathematics. In practice, diagrams are accompanied by a narrative of informal inferences. Attention is drawn to the diagram at appropriate points within the narrative. Requiring GROVER to construct a proof strategy solely on the basis of the diagram places on the machine a responsibility that even mathematicians would find difficult to satisfy.

What we really want is for the user to invoke GROVER directly from the diagram editor (or a textual surrogate for such an editor), without having to see & at all. The user could then refine a diagram dynamically, invoking GROVER periodically during the process. In particular, the user could construct part of the diagram and invoke the GROVER to verify the construction, before proceeding with subsequent constructions. The knowledge of how the diagram has been drawn, and what elements were drawn first, could in turn help GROVER more accurately interpret the diagram as a proof strategy. In particular, this phased approach to diagram construction can help ameliorate the problem of unwanted hypotheses in the individual lemmas that are passed to &.

We have begun work on the implementation of an interface between GROVER and an X-windows-based graphical editor.

8 Conclusion

We have discussed the GROVER graphical theorem proving system, and described the proof of the Diamond Lemma using the system.[4] GROVER is novel in that it allows the user of the system to present a diagram along with the statement of the theorem. GROVER extracts information concerning the proof from this diagram, and uses this information in its search for the proof. The information extracted from the diagram is (currently) of two types. First, intermediate lemmas which together imply the theorem to be proved, and second, an ordering on those intermediate steps which facilitates the proof. This information is automatically extracted from the diagram, and used by the theorem prover. Since the underlying theorem prover, &, is unable to prove the theorem without the aid of the diagram, and is able to complete the proof with the information found there, the successful proof demonstrates the usefulness of the diagram both for human and machine provers.

[4] A longer version of this paper, including a discussion of related work and directions in which we hope to move, is available from the authors on request.

References

[BB91] S.C. Bailin and D. Barker-Plummer. *Automated Deduction in Set Theory.* Final Report NSF/ISI-90006, National Science Foundation, National Technical Information Service, 5285 Port Royal Road, Springfield, VA, 22161, USA, 1991.

[BE90] J. Barwise and J. Etchemendy. Visual information and valid reasoning. 1990.

[Gel63] H. Gelernter. Realization of a geometry theorem proving machine. In Feigenbaum and Feldman, editors, *Computers and Thought*, McGraw Hill, 1963.

[Gil70] P. Gilmore. An examination of the geometry theorem proving machine. *Artificial Intelligence*, 1:171 – 187, 1970.

[Har88] D. Harel. On visual formalisms. *Communications of the ACM*, 31(5):514–530, May 1988.