A Critique of Proof Planning*

Alan Bundy
Division of Informatics,
University of Edinburgh

Abstract. Proof planning is an approach to the automation of theorem proving in which search is conducted, not at the object-level, but among a set of proof methods. This approach dramatically reduces the amount of search but at the cost of completeness. We critically examine proof planning, identifying both its strengths and weaknesses. We use this analysis to explore ways of enhancing proof planning to overcome its current weaknesses.

Preamble

This paper consists of two parts:

1. a brief ‘bluffer’s guide’ to proof planning[1] and
2. a critique of proof planning organised as a 4x3 array.

Those already familiar with proof planning may want to skip straight to the critique which starts at §2.164

1 Background

Proof planning is a technique for guiding the search for a proof in automated theorem proving, [Bundy, 1988, Bundy, 1991, Kerber, 1998, Benzmüller et al, 1997]. The main idea is to identify common patterns of reasoning in families of similar proofs, to represent them in a computational fashion and to use them to guide the search for a proof of conjectures from the same family. For instance, proofs by mathematical induction share the common pattern depicted in figure 1. This common pattern has been represented in the proof planners Clam and λClam and used to guide a wide variety of inductive proofs [Bundy et al, 1990b, Bundy et al, 1991, Richardson et al, 1998].

* The research reported in this paper was supported by EPSRC grant GR/M/45030. I would like to thank Andrew Ireland, Helen Lowe, Raul Monroy and two anonymous referees for helpful comments on this paper. I would also like to thank other members of the Mathematical Reasoning Group and the audiences at CIAO and Scottish Theorem Provers for helpful feedback on talks from which this paper arose.

1 Pointers to more detail can be found at
http://dream.dai.ed.ac.uk/projects/proof_planning.html

© Springer-Verlag Berlin Heidelberg 2002
Inductive proofs start with the application of an induction rule, which reduces the conjecture to some base and step cases. One of each is shown above. In the step case rippling reduces the difference between the induction conclusion and the induction hypothesis (see §1.2 for more detail). Fertilization applies the induction hypothesis to simplify the rippled induction conclusion.

**Fig. 1. ind_strat: A Strategy for Inductive Proof**

1.1 Proof Plans and Critics

The common patterns of reasoning are represented using tactics: computer programs which control proof search by applying rules of inference [Gordon et al, 1979]. These tactics are specified by methods. These methods give both the preconditions under which the tactics are applicable and the effects of their successful application. Meta-level reasoning is used to combine the tactics into a customised proof plan for the current conjecture. This meta-level reasoning matches the preconditions of later tactics to the effects of earlier ones. Examples of such customised proof plans are given in figure 2.

Proof planning has been extended to capture common causes of proof failure and ways to patch them [Ireland, 1992; Ireland & Bundy, 1996b]. With each proof method are associated some proof critics. Critics have a similar format to methods, but their preconditions specify situations in which the method’s associated tactic will fail and instead of tactics they have instructions on patching a failed proof. Each of the critics associated with a method has a different precondition. These are used to decide on an appropriate patch. Most of the critics built to date have been associated with the ripple method, or rather with its principle sub-method, wave, which applies one ripple step (see §1.2 for more detail). Among the