School of Informatics, University of Edinburgh

Centre for Intelligent Systems and their Applications

Automating Proof in Non-standard Analysis

by

Ewen Maclean

Informatics Research Report EDI-INF-RR-0178

School of Informatics **August 2001**
http://www.informatics.ed.ac.uk/

Automating Proof in Non-standard Analysis

Ewen Maclean
Division of Informatics, University of Edinburgh
ewenm@dai.ed.ac.uk

Abstract. Non-standard analysis provides a framework to carry out formally calculus proofs in a much more intuitive manner than in the $\epsilon - \delta$ formulation of Weierstraß. This paper introduces the notions of proof-planning and rippling, and gives an example showing how they can be applied to non-standard analysis to produce readable proofs. We present the related work in the area and give an outline of the proposed work.

1 Introduction

The work proposed in this project is a combination of proof-planning (Bundy, 1988) and an area of mathematics known as non-standard analysis (Robinson, 1966). Proof-planning is a technique which helps to automate proof by working at an abstracted level. Non-standard analysis allows intuitive calculus proofs to be performed with the assurance of correctness thanks to a solid logical construction of the number system that underlies it– namely the hyperreals. The aim of this work is to develop proof-plans which capture the common patterns of reasoning that take place when proving theorems in non-standard analysis, and thereby to automate their proof. The work presented here has been completed in the $\lambda Clam$ higher order proof-planner (Richardson et al., 2000)

We start by giving a brief overview of some concepts in both non-standard analysis and proof-planning; we then present an example proof and discuss some of the related work; finally we highlight possible future directions for our development.

2 Non-standard analysis

Non-standard analysis is a tool which provides an intuitive yet rigorous alternative to Weierstraß's tricky $\epsilon - \delta$ proofs. Abraham Robinson (Robinson,

1966) brought together a number of ideas from mathematical logic to come up with a theory of analysis in the hyperreal domain. By formally introducing an "infinitely close" relation, definitions and proofs become simpler and more intuitive.

As an example of how a definition in non-standard analysis can be simpler and more intuitive than its standard counterpart, consider the standard and the non-standard definitions of a limit.

Definition 1 *The limit l, of a function f at a point a is defined as:*

$$\lim_{x \to a} f(x) = l \equiv (\forall \epsilon > 0)(\exists \delta > 0) \forall x.\ 0 < |x - a| < \delta \rightarrow |f(x) - l| < \epsilon$$

Proofs which make use of this definition are usually very difficult to automate, on account of the alternating quantifiers, which mean that instantiations for δ have to guessed early on in the proof. Theorem 1 gives us an alternative characterisation of a limit making use of the infinitely close relation $\approx$.

Theorem 1 *The limit l, of a function f at a point a can equivalently be defined in non-standard analysis by:*

$$\lim_{x \to a} f(x) = l \iff \forall x.\ x \approx {}^*a \wedge x \neq {}^*a \rightarrow f^*(x) \approx {}^*l$$

As this theorem expresses the notion of limit over the hyperreals, denoted ${}^*\mathbb{R}$, a few points are worth mentioning: the function f^* corresponds to the original function f extended to accept hyperreal arguments whereas ${}^*a$ corresponds to the embedding of real value a in the hyperreals. These notions are introduced for correctness and will not be examined further here. Only an intuitive understanding of their behaviour is needed for the rest of this paper although the interest reader may find out more by examining (Robinson, 1966).

This theorem gives us a simple non-standard characterisation for a limit which formalises the intuition that underlies the $\epsilon - \delta$ formulation of a limit. For a proof of theorem 1 the reader is urged to consult (Fleuriot and Paulson, 2000), for example. We give a demonstration of the intuitive nature of proofs involving limits in section 4.

3 Proof-planning

Proof planning (Bundy, 1988) is a technique for devising an overall plan for a proof, which can then be used to guide the proof search itself. A proof-plan consists of methods and critics. The methods embody common patterns within proofs, such as the use of induction, and the associated tactics carry out these methods explicitly in the object level prover. For each conjecture a precise proof-plan is built, but a general form of a proof-plan can be

developed for certain classes of conjecture. As described by (Ireland, 1992), a method in proof planning has a number of "slots" assigned to it which correspond to a name by which it is recognised, an input which is matched against the current goal, and a set of preconditions which determine whether the method is applicable to the current goal. The method also has a set of effects which define what happens to the goal after it has been applied, and a tactic which controls the object level prover, in which the plan can be executed. Finally, it has an output which is the result of applying the effects to the current goal.

3.1 Critics

The use of inductive theorem provers such as the Boyer-Moore Theorem Prover (Boyer and Moore, 1979) shows that failed proofs can provide useful information about failure, and hence can help to suggest ways to render proofs successful. One piece of useful information that could come out of a failed proof attempt is a suggestion of how to pre-process the conjecture in some way to help the proof succeed. In the context of proof-planning, a critic is a component which analyses this failure. Ireland and others (Ireland, 1992) have done considerable work in this area, which the interested reader is urged to consult.

3.2 Rippling

Rippling is a heuristic used in proof-planning for guiding the proof search. It was initially motivated by Aubin's observation (Aubin, 1976) on how terms introduced by induction are affected by rewriting. Bundy formalised this idea into a theory of annotated rewriting (Bundy et al., 1993), and a formal calculus has been developed from which one can prove termination (Basin and Walsh, 1996). Crucially, it has been shown that rippling can be extended to non-inductive settings by Yoshida (Yoshida et al., 1994), Walsh (Walsh et al., 1992) and Hutter (Hutter, 1997). This is of importance to the current work.

Rippling annotates the conclusion to indicate the difference between the hypotheses and conclusion. The idea is that the extra structure which exists in the conclusion can be annotated, and one can use this annotation to reason about how the proof is proceeding. The basic properties of an annotation are directed wave fronts, wave holes and sinks. Wave fronts, wave holes and sinks are the meta annotation which encapsulate terms. Wave fronts must include wave holes. Sinks correspond to universally quantified variables in the hypotheses. We shall now briefly explain the important properties and definitions of rippling. For a more formal definition of rippling see (Basin and Walsh, 1996).

Well annotated terms All wave fronts must contain at least one wave

hole, and the term within the wave hole is well annotated. All the terms within the wave front, but outside the wave hole are unannotated. Unannotated terms are well annotated.

Skeletons The skeleton of an expression is the set of all expressions that are built up from the unannotated term and the possible wave holes within each wave front.

Erasure The erasure of a well annotated term is the term without any annotation.

For example consider the expression $f(x+y)$. Examples of possible annotations which render it a well annotated term are:

$$f(x+y),\ f(x)^{\uparrow}+y,\ f(x+y)^{\uparrow},\ f(x+y)^{\uparrow},\ f(x)^{\downarrow}+y,\ f(x+y)^{\downarrow}$$

The erasure of all of these terms is simply the original expression $f(x+y)$. As an example, the skeleton of the term $f(x)^{\uparrow}+y$ is $x+y$, because this is the expression built up from the term in the wave hole and the unannotated part. Similarly the skeleton of $f(x+y)^{\uparrow}$ is $\{x, y\}$, because x and y are both wave holes of the same wave front, so the set of possibilities is constructed. The reason for these definitions will become clear when considering the properties of rippling.

Standard rewrite rules can be annotated to fit in with the rippling scheme. Annotated rewrite rules, called wave rules, have to be both skeleton preserving and measure decreasing. We will not describe here in detail the arguments for termination of rippling. Skeleton preservation means that for a rewrite rule $L \Rightarrow R$, the skeleton of L must be a superset of the skeleton of R. For example

$$f(x)^{\uparrow}+y \Rightarrow f(x+y)^{\uparrow}$$
$$f(x+y)^{\downarrow} \Rightarrow f(x)^{\downarrow}+y$$

are wave rules because the skeleton, $x+y$ is preserved under the rewrite. It is possible to see that when rewriting an expression the rewrites move the wavefronts around the expression, encapsulating more or less of the term structure according to their direction. The arrows on the wave fronts can be thought of as describing whether the wave front surrounds more or less of the term structure under rewriting. Wave rules that maintain the direction of the arrow are called longitudinal, and those that change the direction of the arrow are called transverse. Transverse rippling is only allowed if the direction of the wave rules moves from "out" to "in" (up to down). When initially annotating a term the wave fronts are all directed "out". This

condition helps one to think about termination; intuitively one can think of the measure monotonically decreasing because a wave front can move out as far as it can go and then move back in if it is desirable, but it cannot move back out once it has started to "ripple in". Wave rules match expressions if the annotation matches with a subterm within the expression.

The properties of rippling are

Well-formedness If a well annotated term is rewritten by a wave-rule then the result is a well annotated term.

Skeleton preserving The result of applying wave rule to a term has a skeleton which is a subset of the original term.

Correctness When a term is rewritten by a wave rule the erasure of the rewrite corresponds to a rewrite in the unannotated theory.

Termination Rippling terminates.

This has been a brief overview of the calculus of rippling. For a fuller description see (Basin and Walsh, 1996), and for a description of the existing work on colouring terms for equational rewriting, which we do not use, see (Hutter, 1997).

4 A brief case study

This section presents a plan which has been generated automatically by $\lambda Clam$. Other proofs which have been performed include $LIM+$, the chain rule for differentiation and the product rule.

A useful example of a limit theorem to consider in the context of non-standard analysis and proof-planning is that of $LIM\times$. This theorem states that the product of the limits of two real valued functions is equal to the limit of the product of the functions. This conjecture is difficult to automate in standard analysis (Melis et al., 2000), but we show here that the structure of its non-standard counterpart is very simple. This example proof demonstrates the advantage of using non-standard analysis over standard analysis, and shows how existing proof-planning machinery can be used to model the reasoning patterns in non-standard proofs. The standard version of this proof is challenging on paper, and is extremely hard to automate. The $\epsilon - \delta$ formulation involves quantifiers which are arranged in such a way that a guess has to made about the instantiation for quantified variables before enough information is available. The conjecture in standard analysis is stated as

$$\lim_{x\to a} f(x) = l_f \wedge \lim_{x\to a} g(x) = l_g \vdash \lim_{x\to a} f(x) \times g(x) = l_f \times l_g$$

The hypotheses to the conjecture become:

$$c, l_f, l_g : \mathbb{R} \tag{1.1}$$
$$f, g : \mathbb{R} \to \mathbb{R} \tag{1.2}$$
$$\forall x.\ x \approx {}^*c \wedge x \neq {}^*c \to f^*(x) \approx {}^*l_f \tag{1.3}$$
$$\forall x.\ x \approx {}^*c \wedge x \neq {}^*c \to g^*(x) \approx {}^*l_g \tag{1.4}$$

using the non-standard characterisation for a limit. Hypotheses (1.1) and (1.2) describe the typing information of the constants.

The hypotheses (1.3) and (1.4) can now be embedded in the conclusion. This yields the annotated conclusion

$$x \approx {}^*c \wedge x \neq {}^*c \quad \to \quad \boxed{f^*(x)_{red} \times g^*(x)_{blue}}^{\uparrow} \approx \boxed{{}^*l_f{}_{red} \times {}^*l_g{}_{blue}}^{\uparrow}$$

The terms belonging to different hypotheses inside the wave fronts are given different colours. The shading around the variable x in the conclusion refers to positions which correspond to universally quantified variables in the hypotheses. In rippling terminology these are referred to as *sinks*. The wave rules that are available to the system are the following:

$$[finite(Y) \wedge finite(B)] \quad \boxed{X_{red} \times A_{blue}}^{\uparrow} \approx \boxed{Y_{red} \times B_{blue}}^{\uparrow} \Rightarrow \boxed{X \approx Y_{red} \wedge A \approx B_{blue}}^{\uparrow} \tag{1.5}$$

$$A \to \boxed{B_{red} \wedge C_{blue}}^{\uparrow} \Rightarrow \boxed{A \to B_{red} \wedge A \to C_{blue}}^{\uparrow} \tag{1.6}$$

Here when we write $A \Rightarrow B$ we mean that A rewrites to B, and hence it is true that $B \to A$. The side conditions involving the predicate $finite$ refer to all hyperreal numbers which are not infinite— for example real numbers.

Wave rule (1.5) represents the continuity of $\times$, which is itself a complicated theorem to prove in standard analysis, and renders this proof very simple indeed. In non-standard analysis, it is a challenging problem involving mainly equational reasoning, but is much simpler than its standard counterpart as we discuss in section 4.1.

By inspecting the annotation on the conclusion and that on the rewrite rules, the general idea behind rippling can be seen. The wave fronts should increase until the terms in the wave holes can match with the hypotheses. After application of each of these rules, the conclusion becomes

$$\boxed{x \approx {}^*c \wedge x \neq {}^*c \to f^*(x) \approx {}^*l_f{}_{red} \wedge x \approx {}^*c \wedge x \neq {}^*c \to g^*(x) \approx {}^*l_g{}_{blue}}^{\uparrow}$$

at which point the holes are instances of the hypotheses as required. Now the conclusion contains instances of the hypotheses, and the plan of the proof is complete.

Part of the output from the plan given by $\lambda Clam$ showing wave rules (1.1) and (1.2) is given below. The formatting has been changed slightly by hand so as not to take up too much space.

```
Attempting...
cond_wave_method _841981 cont_times
cond_wave_method outward cont_times
succeeded

X:hyperreal, c:real, f:real → real, g:real → real, l_f:real, l_g:real
∀  Y:hyperreal. ((Y ≈ emb  c) ∧ ¬ (Y = emb c)) →  ext f Y ≈ emb l_f
∀  Y:hyperreal. ((Y ≈ emb  c) ∧ ¬ (Y = emb c)) →  ext g Y ≈ emb l_g
⊢  ((X ≈ emb c) ∧ ¬ (X = emb c)) → (ext f X ≈ emb l_f) ∧ (ext g X ≈ emb l_g)

1 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: nil                    %%current plan node

Attempting...
patch_meth (wave_method outward _818076) wave_critic_strat
patch_meth (wave_method outward prop_wr1) wave_critic_strat
succeeded

X:hyperreal, c:real, f:real → real, g:real → real, l_f:real, l_g:real
∀ Y:hyperreal. ((Y ≈ emb c) ∧ ¬ (Y = emb c)) → ext f Y ≈ emb l_f
∀ Y:hyperreal. ((Y ≈ emb c) ∧ ¬ (Y = emb c)) → ext g Y ≈ emb l_g
⊢  ((X ≈ emb c) ∧ ¬ (X = emb c)) → (ext f X ≈ emb l_f) ∧
   ((X ≈ emb c) ∧ ¬ (X = emb c)) → (ext g X ≈ emb l_g)

1 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: nil
```

Here `emb` denotes the embedding of a real number in the hyperreals, and `ext` the non-standard extension of a real function.

The non-standard proof follows a simple pattern of reasoning which proof-planning is able to model. In this example, we have introduced wave rule (1.5) to the system so that the simplicity of the overall structure is apparent. The annotation gives the proof extra information which can be used to speculate lemmas, and proof transformations. For example, consider what would happen if the wave rule (1.5) had not been added to the system. The rewriting mechanism would then fail, prompting a critic to respond to this failure, and suggest a new wave rule according to the hypotheses. See section 4.1 for a discussion of this wave rule. This would enable the proof to automatically recover and go through even in the absence of the rewrite rule. The advantage of the annotation in this case, is that it can be made explicit which terms belong to which hypotheses, and hence which should be brought closer together.

4.1 The continuity of $\times$

The following conjecture represents the lemma which has to speculated and whose proof must be planned automatically in order to justify the claim that the construction of a plan for the proof $LIM\times$ has been automated. This work is currently in progress.

$$\vdash \forall a, b, x, y.\ M \wedge x \approx y \wedge a \approx b \ \rightarrow\ x \times a \approx y \times b$$

In this case the meta-variable M has to be instantiated to the conditions of the rule. After the usual $\forall$ and implication introduction, we proceed as follows:

$$\vdash x \times a \approx y \times a \wedge y \times a \approx y \times b$$

from transitivity. We then speculate the following lemma:

$$M_1 \wedge x \approx y \vdash x \times a \approx y \times a$$

where M_1 is a new meta-variable which must be instantiated to the conditions. Now we use an arithmetic rule to yield the conclusion:

$$\vdash x \times a - y \times a \approx {}^*0$$

and use the distributive law to write this as:

$$\vdash (x - y) \times a \approx {}^*0$$

and finally rewrite this using the following axioms:

$$X \approx {}^*0 \wedge finite(Y) \rightarrow X \times Y \approx {}^*0$$
$${}^*0 + x = x$$
$$x + (-x) = {}^*0$$

to yield:

$$\vdash finite(a) \wedge x \approx y$$

which completes the proof, and M_1 is instantiated to the condition $finite(a)$.

Using this lemma we can proceed with the original proof. The conclusion was at the state:

$$x \times a \approx y \times a \wedge y \times a \approx y \times b$$

which can now be rewritten as:

$$finite(a) \wedge x \approx y \wedge finite(y) \wedge a \approx b$$

which completes the proof, with the instantiation of the meta-variable M as $finite(a) \wedge finite(y)$. Including the commutativity of $\times$ and the symmetry of $\approx$ in the original equation, we can also derive new conditions, meaning that the most general condition for M is $(finite(x) \vee finite(y)) \wedge (finite(a) \vee finite(b))$. The use of transitivity in this example is analogous to the use of the triangle inequality in the standard proofs.

5 Related work

A reasonable amount of work has already been completed in both standard and non-standard analysis. (Bledsoe et al., 1972) is the first piece of work on automating parts of standard analysis, and presents a resolution based theorem prover which automates certain standard analysis proofs, and presents certain heuristics such as the *limit heuristic* which are specific to analysis proofs. (Bledsoe and Ballantyne, 1977) presents the same theorem prover modified to automate proofs in non-standard analysis. The goal of this work was to demonstrate how non-standard analysis allows mathematicians to prove theorems in analysis more easily than by using the $\epsilon - \delta$ formulation. The paper presents the methods employed in the prover, and explains how it handles the various types that occur in non-standard analysis. Their work uses a generous axiomatisation and the prover is able to prove simple results about standard parts of numbers, as well as more significant results such as $LIM+$, and even the Bolzano-Weierstrass conjecture that a continuous function on a compact set is uniformly continuous. Their proofs, however, are not as readable as those generated by our proof-planning approach, as they are driven by resolution, and the output is linear. Through proof-planning the tree structure of method applications in the proof plan can be seen, and hence the structure of the proof itself is clearer.

In the ΩMEGA proof-planner and mathematical assistant (Benzmüller et al., 1997), there has been much work done in developing proof-plans for limit theorems using what is referred to as "multiple strategies". The ΩMEGA proof-planner builds on Bledsoe's work by introducing a "complex estimate", and a constraint solver which carefully manages the introduction of meta-variables in order to calculate the difficult instantiations of the variables in standard proofs. The system is able to prove a substantial number of theorems at present, and a proof presentation system has been included in the system to make the proofs readable. The work is documented in (Melis et al., 2000) and (Melis and Siekmann, 1999) for example.

The Mathpert system developed by Beeson is a mathematical assistant which is also capable of automatically proving properties about functions in standard analysis such as continuity (Beeson, 1998). Often the proofs involve evaluating limits, and Beeson found that this evaluation could be simplified using of some axioms from non-standard analysis (Beeson, 1995).

The ACL2 theorem prover originally did not include a theory for reasoning about conjectures over the reals, but Gamboa (Gamboa, 1999) used Nelson's Internal Set Theory (Nelson, 1977) to extend the prover to automatically prove theorems about real valued functions.

The most significant work that has been done in this area has been done by Fleuriot (Fleuriot and Paulson, 2000) in the interactive theorem prover Isabelle/HOL. This work is the first to mechanise the construction of the hyperreals using the ultrapower construction used by Robinson to develop

non-standard analysis. A substantial amount of real analysis has also been proved in the hyperreal domain using this construction. It also highlights that there are proofs in non-standard analysis, though simpler than their standard counterparts, which cannot be done automatically. We believe that our proposed approach should enable these to be planned automatically.

Other work in progress that is related to the current work is that of Heneveld (Heneveld et al., 2000). This work models the cognitive processes of mathematicians solving integration problems. The $\lambda Clam$ proof-planner is also being used to do this, and many of the rules used, such as integration by parts, can potentially be proved using the work proposed here.

6 Proposed work

The work we propose in this project involves generating plans for proofs in non-standard analysis automatically, and thence executing them in an object level theorem prover. The methodology is one of genericity and automation. We have compiled a set of minimal axioms for non-standard analysis from which to derive all of the rules we require, and we hope to build a generic proof-plan whose initial state will be constant, but whose eventual shape will be transformed by the addition of intelligent critics. In addition to this, we have introduced a method for unfolding axioms by introducing such notions as *field homomorphism*. For example, the reals and the hyperreals form a field, so it is sensible to express field axioms in such a way, that the axioms specific to each type can be generated in a generic manner. This is similar to the work done in Isabelle on *Axiomatic Type Classes* (Wenzel, 2000).

Part of the work still yet to be completed involves the collection of a set of conjectures with which to test the system. The initial area of interest lies within *limit* and *continuity* conjectures, which have already been attempted in standard analysis, for example in the $\Omega MEGA$ system. Some of these proofs have already been performed as a trial in an earlier version of $\lambda Clam$. Also of interest are fundamental proofs about calculus such as the chain rule, the product rule, and the fundamental theorem of calculus. The former two have also been attempted in the earlier version of $\lambda Clam$ with some success.

7 Conclusion

The main aim of this project is to gain an understanding of the structure of proofs in non-standard analysis. Proof-planning allows this to be done in a declarative way by formally specifying plans. We hope that this will show that non-standard analysis is a better way of going about proving theorems about analysis in general, and also, perhaps, would also be a more sensible alternative for teaching purposes than the infamous $\epsilon - \delta$ proofs of standard analysis.

Bibliography

Aubin, R. (1976). *Mechanizing Structural Induction*. PhD thesis, University of Edinburgh.

Basin, D. and Walsh, T. (1996). A calculus for and termination of rippling. *Journal of Automated Reasoning*, 16(1–2):147–180.

Beeson, M. (1995). Using nonstandard analysis to verify the correctness of computations. *International Journal of Foundations of Computer Science*, 6(3):299–338.

Beeson, M. (1998). Automatic generation of epsilon-delta proofs of continuity. *Artificial Intelligence and Symbolic Computation*, pages 67–83. Lecture Notes in Artificial Intelligence No. 1476.

Benzmüller, C., Cheikhrouhou, L., Fehrer, D., Fiedler, A., Huang, X., Kerber, M., Kohlhase, K., Meier, A., Melis, E., Schaarschmidt, W., Siekmann, J., and Sorge, V. (1997). Ωmega: Towards a mathematical assistant. In McCune, W., editor, *14th International Conference on Automated Deduction*, pages 252–255. Springer-Verlag.

Bledsoe, W. W. and Ballantyne, A. M. (1977). Automatic proofs of theorems in analysis using nonstandard techniques. *Association for Computing Machinery*, 24(3):353–374.

Bledsoe, W. W., Boyer, R. S., and Henneman, W. H. (1972). Computer proofs of limit theorems. *Artificial Intelligence*, 3:27–60.

Boyer, R. S. and Moore, J. S. (1979). *A Computational Logic*. Academic Press. ACM monograph series.

Bundy, A. (1988). The use of explicit plans to guide inductive proofs. In Lusk, R. and Overbeek, R., editors, *9th International Conference on Automated Deduction*, pages 111–120. Springer-Verlag. Longer version available from Edinburgh as DAI Research Paper No. 349.

Bundy, A., Stevens, A., van Harmelen, F., Ireland, A., and Smaill, A. (1993). Rippling: A heuristic for guiding inductive proofs. *Artificial Intelligence*, 62:185–253. Also available from Edinburgh as DAI Research Paper No. 567.

Fleuriot, J. D. and Paulson, L. C. (2000). Mechanizing nonstandard real analysis. *LMS Journal of Computation and Mathematics*, 3:140–190.

Gamboa, R. (1999). *Mechanically Verifying Real-Valued Algorithms in ACL2*. PhD thesis, The University of Texas at Austin.

Heneveld, A., Maclean, E., Bundy, A., Fleuriot, J., and Smaill, A. (2000). Towards a formalisation of college calculus. In *Proceedings of the 2000 Calculemus Symposium: Systems for Integrated Computation and Deduction*.

Hutter, D. (1997). Colouring terms to control equational reasoning. *Journal of Automated Reasoning*, 18:399–442.

Ireland, A. (1992). The Use of Planning Critics in Mechanizing Inductive Proofs. In Voronkov, A., editor, *International Conference on Logic Programming and Automated Reasoning – LPAR 92, St. Petersburg*, Lecture Notes in Artificial Intelligence No. 624, pages 178–189. Springer-Verlag. Also available from Edinburgh as DAI Research Paper 592.

Melis, E. and Siekmann, J. (1999). Knowledge-based proof planning. *Artificial Intelligence*, 115(1):65–105.

Melis, E., Zimmer, J., and Müller, T. (2000). Extensions of constraint solving for proof planning. *European Conference on Artificial Intelligence*.

Nelson, E. (1977). Internal set theory: A new approach to nonstandard analysis. *Bulletin American Mathematical Society*, 83. Available from `http://www.math.princeton.edu/~nelson/books.html`.

Richardson, J., Dennis, L., Gow, J., and Jackson, M. (2000). User/programmer manual for the $\lambda Clam$ proof planner.

Robinson, A. (1966). *Non-standard Analysis*. North-Holland Publishing Company, Amsterdam. Studies in Logic and the Foundations of Mathematics.

Walsh, T., Nunes, A., and Bundy, A. (1992). The use of proof plans to sum series. In Kapur, D., editor, *11th International Conference on Automated Deduction*, pages 325–339. Springer Verlag. Lecture Notes in Computer Science No. 607. Also available from Edinburgh as DAI Research Paper 563.

Wenzel, M. (2000). Using axiomatic type classes in isabelle. Available from `http://isabelle.in.tum.de/doc/`.

Yoshida, T., Bundy, A., Green, I., Walsh, T., and Basin, D. (1994). Coloured rippling: An extension of a theorem proving heuristic. In Cohn, A. G., editor, *Proceedings of ECAI-94*, pages 85–89. John Wiley.