

Proof-Planning Critics

Turning failed proofs into solutions

Moa Johansson Supervisors: Alan Bundy and Lucas Dixon

moa.johansson@ed.ac.uk

<http://dream.inf.ed.ac.uk/projects/critics/>

Abstract: Discovery of new lemmas, case-splits, generalisations and similar steps are a major challenge for automated theorem-provers. Proof-planning critics can help automate these steps by using information from failed proof-attempts.

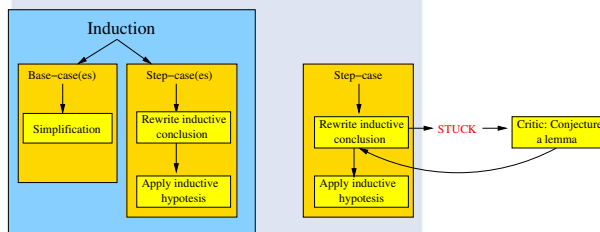
Introduction

Many mathematical proofs involve some form of *eureka step*, such as the discovery of an intermediate lemma, a case-split or a revision of the induction scheme. Most interactive theorem provers put the burden of discovering eureka steps on the user. Proof-planning critics were proposed by Ireland et al. as a way of helping automate this process by using information from failed proof-attempts [3]. Each critic is designed to anticipate a particular type of failure, and suggests a patch to recover the proof attempt.

Our research is in the early stages of developing proof critics in the IsaPlanner system [2]. With the assistance of critics IsaPlanner will be able to automatically prove many theorems that previously would have required human intervention.

What is Proof-Planning?

Proof-planning is a technique that guides search in automated theorem proving by exploiting the fact that there exists families of mathematical proof that share a similar structure [1]. One such family are proofs by induction. The development of proof-planning was motivated by the observation that human mathematicians often first have a high level plan for how to go about solving a proof and then fill in the exact details.



Left: A basic proof-plan for induction. An inductive proof requires first proving the base-case, then the step-case showing that the inductive conclusion follows from the hypothesis.

Right: If a proof-planning attempt fails, a critic can patch the proof by for example conjecturing and proving a missing lemma.

When can it go wrong?

The proof-planner will typically try to complete a proof using logical rules and a set of equations, arising from lemmas and theorems as well as function definitions and axioms. These equations are used as rules specifying how a conjecture can be rewritten. There may be many alternative ways of rewriting a conjecture so the proof-planner employs heuristic measures to guide this process. Sometimes, no more rules that improve on the heuristic measure are available, and the proof-planner gets stuck. Depending on what the stuck goal looks like and the way the planner tried prove it, a critic suitable for that type of failure is fired. The critic suggests a patch that will allow the proof-attempt to continue.

Example: Discovering a lemma

As an example, consider the conjecture:

$$\text{rev}(\text{rev}(t) @ l) = \text{rev}(l) @ t$$

The *rev*-function reverses a list, the @-symbol denote the list append function and :: adds an element to the front of a list. Following the proof-plan for induction on the list t , we get a the following:

$$\text{Hypothesis : } \text{rev}(\text{rev}(t) @ l) = \text{rev}(l) @ t$$

$$\text{Conclusion : } \text{rev}(\text{rev}(h :: t) @ l) = \text{rev}(l) @ (h :: t)$$

The proof-planner will use the definition of the reverse function, $\text{rev}(h :: t) = \text{rev}(t) @ [h]$, to try and transform the conclusion to a form that the hypothesis can be applied to. After this, the planner will however be stuck as none of the rules from our function definitions can improve on the new goal:

$$\text{rev}(\text{rev}(t) @ [h] @ l) = \text{rev}(l) @ (h :: t)$$

It is not possible to apply the hypothesis, so a critic is fired to try and find a lemma. The critic begins by choosing a sub-term of the goal that the new lemma should match. Here we pick the right hand side term of the goal above, $\text{rev}(l) @ (h :: t)$. This term will be used as left-hand side of the new lemma. In order to increase the chance of making the hypothesis applicable, we want to preserve the parts of the goal that are similar to the hypothesis. Therefore, the right-hand side of the lemma is

constructed by keeping those similar parts and inserting meta-variables standing for yet unknown term structures. Here, everything except the $h :: _$ -term is similar the hypothesis:

$$\text{rev}(l) @ (h :: t) = F(\text{rev}(l) @ t)$$

Subsequent applications of rewrite rules will instantiate the meta-variable F . In our case, this will eventually give rise to the lemma:

$$\text{rev}(l) @ (h :: t) = \text{rev}(l) @ [h] @ t$$

Other areas

There are many other areas where critics can be useful, for example:

- Identifying that a switch from a one-step induction scheme to a two-step scheme is needed. This may be needed in proof about function like *even*, defined as $\text{even}(n + 2) = \text{even}(n)$.
- Finding generalisations. Sometimes it is easier to prove a more general version of a theorem because it gives a stronger inductive hypothesis. Proofs about tail-recursive functions often need such a generalisation.
- Guiding case-splitting. If conditional rewrite rules are present case-splits may be necessary. An example is verification proofs of many common sorting algorithms.
- Assisting proofs of conjectures with complex propositional structure. It may be necessary to split the goal into several parts to make the inductive hypothesis applicable. This problem can occur in synthesis proofs.

References

- [1] A. Bundy. The use of explicit plans to guide inductive proofs. In *9th International Conference on Automated Deduction*, pages 111–120, 1988.
- [2] L. Dixon. *A generic approach to proof planning*. PhD thesis, School of Informatics, University of Edinburgh, 2005.
- [3] A. Ireland and A. Bundy. Productive use of failure in inductive proof. *Journal of Automated Reasoning*, 16(1-2):79–111, 1996.

