Combining SEPIA and ML4PG

Thomas Gransden[1]

Department of Computer Science, University of Leicester, UK tg75@student.le.ac.uk

Abstract: SEPIA is an approach that infers state based models from existing proof corpora. These models can then be used for the development of new proofs. As with similar approaches, selecting the best facts to use in new proofs is challenging. We investigate the potential use of ML4PG as a relevance filter for our approach - achieving a model only inferred from the lemma suggestions by ML4PG. These reduced models still contain the neccesary tactics to achieve proofs.

1 Introduction

In the past two decades, Interactive Theorem Provers (ITP) have been used in a number of significant proof developments. These range from verifying proofs of mathematical properties such as the Four-Color and Odd-Order theorems through to proving compilers and OS kernels are correct. These developments contribute large amounts of knowledge to the many proof corpora available.

The proof libraries have now become extremely large, and keeping track of the best facts to use in new proofs can be challenging. For instance, Sledgehammer [4] outsources proof obligations to automated theorem provers. Along with a conjecture, existing facts must also be provided – however in the presence of too many facts even the most powerful automated tools will struggle to deliver a proof.

Therefore, *relevance filtering* must occur to select only the most relevant facts to provide to the automated systems. Initially, the relevance filter implemented was reasonably naive (selecting facts based on symbols and functions from the conjecture) but still effective. Kühlwein *et al.* have recently improved this with the help of machine learning techniques [3].

We propose the use of ML4PG [2] to act as a relevance filter for our model inference approach (briefly outlined in the next section). ML4PG has been shown to identify commonalities between lemmas in large proof libraries, and harnessing this knowledge may be beneficial to our approach. The overall search space will be reduced whilst still leading to models that contain the necessary tactics needed to derive a proof.

2 Modelling proofs with state machines

In our previous work [1], we have applied model inference techniques to corpora of Coq proofs (we have since named this approach SEPIA - Search for Proofs using Inferred Automata). The models inferred are Extended Finite State Machines (EFSM). The main conceptual difference between EFSMs and traditional finite state machines is that transitions may have a guard that places constraints on the parameters that may be used. Figure 1 presents a small example of an EFSM inferred from Coq proofs.

The inferred models have been shown to accurately capture the reasoning patterns within a set of proofs. We showed how the models can be used manually to derive proofs new properties that weren't in the original corpora. As the size of the libraries increases, the models become too large to consider processing manually. However, ongoing work is investigating the automation of this process.

A transition in an EFSM model of Coq proofs may look like the following:

rewrite (p="plus_n_0" || p="O_minus")

The intended semantics of this transition is that it should only be followed if the Coq tactic `rewrite` is applied with either `plus_n_O` or `O_minus` given as the parameters. The full discussion of the inference process and it's application to interactive proofs can be found in previous work [1, 5].

3 Employing ML4PG as a relevance filter

Our current approach when inferred the models is to use *whole* theories of Coq proofs. However, at any one point only a handful of the lemmas within these theories may be useful in a proof attempt. We propose the use of ML4PG [2] as a relevance filter for our model inference technique.

ML4PG is a tool that can take large corpora of Coq/SSReflect proofs and identify commonalities between lemmas and definitions - for instance the whole SSReflect library has been clustered and analysed. We use ML4PG to produce clusters that identify the most useful lemmas to infer a model from. This leads to a reduction in complexity, without affecting the accuracy of the inferred models.

We demonstrate this idea with a small example. Consider a scenario where we are trying to prove the lemma `take_size` from the `seq` theory in the SSReflect development[1]. The theory contains 393 other proofs, and inferring a model from all of these proofs leads to a state machine containing 88 states and over 250 transitions (this is too large to show in this paper). The challenge we are faced with is this: can we reduce the complexity of the model (using ML4PG) and still prove the lemma?

By using ML4PG, the lemma `take_size` appears in a cluster with 33 other lemmas. Instead of inferring the model from the whole theory, we instead infer only from these proofs. This provides us a much smaller state machine containing 13 states with much fewer transitions than before. The model in Figure 1 shows the reduced EFSM.

[1] http://ssr.msr-inria.inria.fr

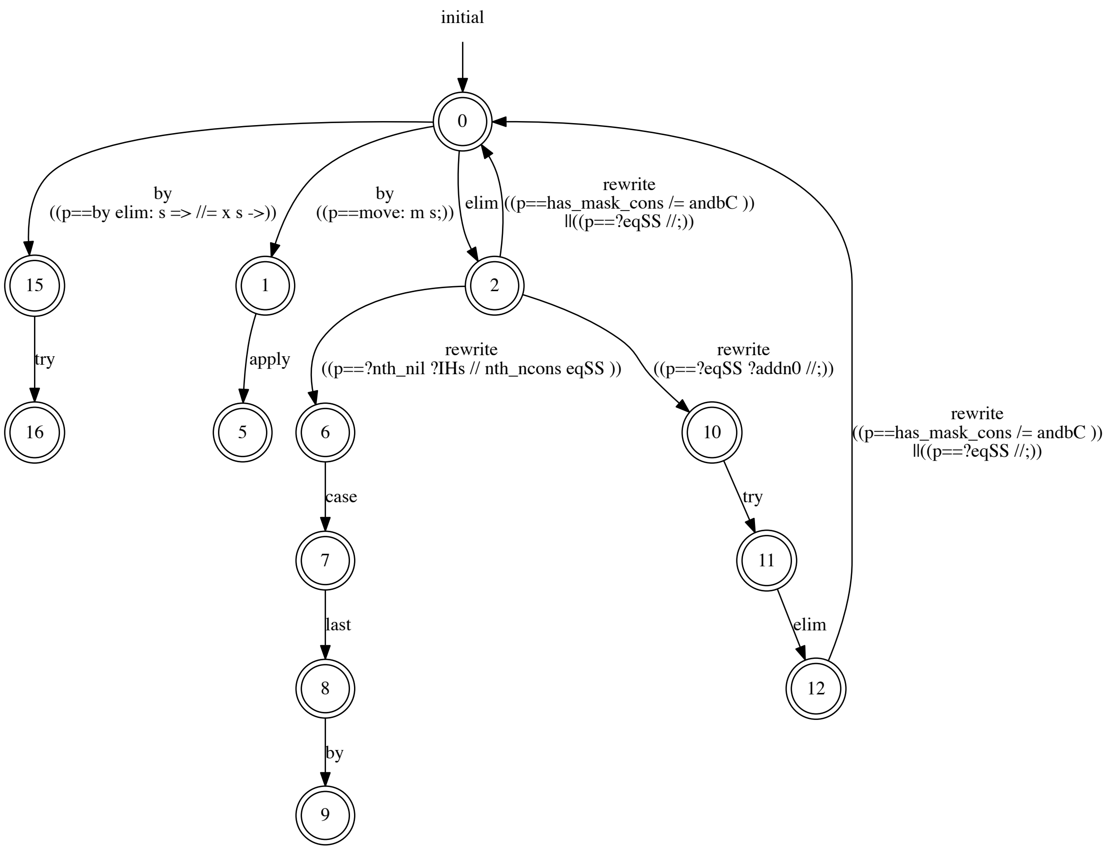

Figure 1: Model Inferred from ML4PG Suggestions

This reduced state machine yields the necessary proof (found by following the transition from state 0 to state 15): `by elim: s => //= x s -->`. Although a fairly trivial example, this demonstrates the potential benefits of using ML4PG to filter large proof corpora. Of course, using a reduced set of lemmas not only improves the time taken to infer the model but also reduces the possible search time.

4 Conclusion and Future Work

We have demonstrated that combining ML4PG with SEPIA can be beneficial when deriving proofs in Coq. By reducing the size of the inferred models, the overall process of finding a proof becomes simpler. The models are still useful, even when reducing their complexity. Current work is focussing on automating this process, so that proof attempts can be completed fully-automatically.

This work naturally brings up many potential ways of combining the two approaches. Initially, we have studied reducing the search space by disregarding lemmas from the model inference process. We plan to empirically evaluate the combined methods further to gain deeper insight into the potential benefits of the combined tools.

References

[1] Thomas Gransden, Neil Walkinshaw, and Rajeev Raman. Mining State-Based Models from Proof Corpora. In *Intelligent Computer Mathematics*, volume 8543 of *LNCS*, pages 282–297. Springer, 2014.

[2] Ekaterina Komendantskaya, Jónathan Heras, and Gudmund Grov. Machine Learning in Proof General: Interfacing Interfaces. In *UITP*, volume 118 of *EPTCS*, pages 15–41, 2013.

[3] Daniel Kühlwein, Jasmin Christian Blanchette, Cezary Kaliszyk, and Josef Urban. MaSh: Machine Learning for Sledgehammer. In *Interactive Theorem Proving*, volume 7998 of *LNCS*, pages 35–50. Springer, 2013.

[4] Jia Meng and Lawrence C. Paulson. Lightweight relevance filtering for machine-generated resolution problems. *Journal of Applied Logic*, 7(1):41 – 57, 2009.

[5] Neil Walkinshaw, Ramsay Taylor, and John Derrick. Inferring Extended Finite State Machine Models from Software Executions. *Journal of Empirical Software Engineering*, 2015. To appear.