

Combining SEPIA and ML4PG

Thomas Gransden

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

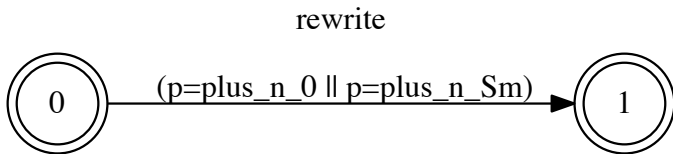
Automated Reasoning Workshop - 9/10 April 2015

SEPIA: An Introduction

SEarching for Proofs using Inferred Automata

SEPIA applies model inference techniques to interactive proofs

Extended Finite State Machines = tactic sequences + arguments



Theorem Proving Using SEPIA

Generate proofs by traversing paths through model

Can be used manually (CICM'14) and (even better) automatically (via ProofGeneral)!

Inference algorithms model “new” sequences of tactics

An unexpected consequence - theory analysis?

Inference Process

Feel free to discuss this with me during the workshop!

SEPIA: Proof Mode

Callable during proof development - uses all previous proofs and any other opened theories

```
listnat.v | *sepia-output*
rewrite aux7.
rewrite <- mult_0_n.
trivial.
Qed.

(*63*)
(*Same proof*)
Lemma M28 : forall a, (a - a) * S a = 0.
Proof.
intro.
rewrite aux7.
rewrite <- mult_0_n.
trivial.
Qed.

(*64*)
Lemma M29 : forall a b, (a - a) * S b = 0.
Proof.
107 [main] INFO org.bitbucket.efsmtool.proofs.ut
ilities.ProofGeneralEntry - 64 proofs read
632 [main] INFO org.bitbucket.efsmtool.proofs.se
arch.ProofGeneralProofSearch - Initialising Pro
of Search
642 [pool-47-thread-1] INFO org.bitbucket.efsmto
ol.proofs.search.ProofGeneralProofSearch - Coq ...
Process Started - trying to prove lemma
1326 [pool-47-thread-1] DEBUG org.bitbucket.efsm
tool.proofs.search.ProofGeneralProofSearch - SU
CCESS
1328 [pool-47-thread-1] DEBUG org.bitbucket.efsm
tool.proofs.search.ProofGeneralProofSearch - Pr
oof was: intro. rewrite aux7. trivial.
357 tactics evaluated.
Inference and search took 690 milliseconds
Inference and search took 0 min, 0 sec
```

Limitations of SEPIA

Exploding State Space

SEPIA learns from whole theories of proofs. However more proofs naturally leads to larger models with more redundancy.

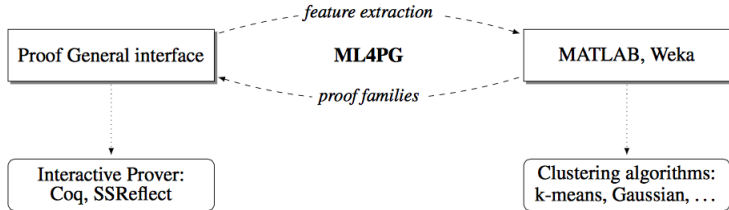
“Static” Learning Process

SEPIA treats the proof script completely textually - so state machine transitions fail if variables are named differently e.g. **induction n** is not applicable to **forall a:nat, a + 0 = a..**

How can we circumvent these issues?

ML4PG: Machine Learning for Proof General

A ProofGeneral extension that uses machine learning to analyse large corpora of proofs.



Combining with ML4PG

How can ML4PG help with the issues described previously?

Use Case: Reducing State Space with ML4PG

Currently, *whole* theories are used to infer a model from. There are many proofs that aren't useful in any one proof derivation.

Use ML4PG to identify statistically similar lemmas and infer a model from these suggested lemmas instead of whole theories.

This is a nice way of automating ML4PG's output - something that isn't currently done.

Use Case: Inferring “Dynamic” models from ML4PG features

One of the features that ML4PG extracts is the *relation of a tactic argument to a hypothesis or external lemma*.

We can use this information as input to SEPIA - eventually getting more *semantic* transitions such as:

- ① perform induction on a variable from the goal
- ② rewrite using a lemma from a particular cluster

Conclusion

We have shown that SEPIA and ML4PG can be used in conjunction with each other. In our future work we plan to rigorously evaluate the combined tools on large scale datasets to show the potential benefits.

Questions/Comments Welcome!