PTTP+GLiDeS:
Using Models to Guide Linear Deductions

Marianne Brown and Geoff Sutcliffe

School of Information Technology
James Cook University
{marianne,geoff}@cs.jcu.edu.au

Abstract. PTTP+GLiDeS is a linear deduction theorem prover that uses semantics to guided its search for a proof. The semantics are automatically generated by using MACE to find a model for a subset of the input clause set. Results have shown that where an "effective" model is found, the guidance it provides is of great benefit. This paper discusses how MACE is used in PTTP+GLiDeS, and what properties are desirable in models and model generators for this application.

1 Introduction

PTTP+GLiDeS is a linear deduction theorem prover which uses models to guide its search for a proof. A model is generated for a subset of the input clause set by MACE [?] version 1.3.3. The model is used by the theorem prover to evaluate the truth value of certain literals generated in the deduction. Depending on the results of this evaluation, pruning of the search space may occur. The success of the guidance strategy is dependent on the model. A model that provides little guidance, i.e., results in the theorem prover taking the a similar path to the one that it would have taken if no guidance had been attempted, results in high CPU overheads for little or no gain. On the otherhand, an inappropriate model may cause all proofs to be pruned from the search space. The ideal model provides enough pruning so that the reduction in the search space offsets the overheads associated with the guidance, but leaves proofs in the remaining search space.

The following section describes in some detail the semantic guidance strategy used by PTTP+GLiDeS. In Section ??, the manner in which MACE is used in PTTP+GLiDeS is described and its weaknesses are discussed. Desirable properties of models and model generators for this application are outlined in Section ??.

2 The Semantic Guidance

PTTP+GLiDeS uses a semantic pruning strategy that is based upon the strategy that can be applied to linear-input deductions. In a linear-input deduction all centre clauses can be required to be FALSE in a model of the side clauses. To implement this strategy it is necessary to know which are the potential side clauses, so that a model can be built. A simple possibility is to choose a negative top clause from a set of Horn clauses, in which case the mixed clauses are the potential side clauses. More sensitive analysis is also possible [?,?]. Linear-input deduction and this pruning strategy are complete only for Horn clauses. Unfortunately, the extension of this pruning strategy to linear deduction, which is also complete for non-Horn clauses, is not direct. The possibility of ancestor resolutions in a linear deduction means that centre clauses may be TRUE in a model of the side clauses.

In PTTP+GLiDeS, rather than placing a constraint on entire centre clauses, a constraint is placed on certain literals of the centre clauses: The input clauses other than the chosen top clause of a linear deduction are named the *model clauses*. In a completed linear refutation, all centre clause literals that have resolved against input clause literals are required to be FALSE in a model of the model clauses, called the *guiding model*. TRUE centre clause literals must be resolved against ancestor clause literals.

PTTP+GLiDeS implements linear deduction using the Model Elimination [?] (ME) paradigm. PTTP+GLiDeS maintains a list of all the A-literals created in a deduction. This list is called the *A-list*. The pruning strategy requires that at every stage of the deduction there must exist at least one ground instance of the A-list that is FALSE in a guiding model. The result is that only FALSE B-literals are extended upon, and TRUE B-literals must reduce.

3 Model Generation

MACE is used to generate a guiding model. MACE is given the set of model clauses and a time limit of 150 seconds. The domain size for the model is set to the number of constants in the problem. If a model of this size can't be found, then the domain size is set to 2 and MACE is allowed to determine the domain size. While MACE is capable of producing many models for a given set of clauses, in the current implementation only the first model is used. The chosen top clause must have at least one FALSE instance in the guiding model. Once the model is generated and checked against the top clause, the model is given to the theorem prover for use in the semantic guidance.

MACE was selected because it was found to be easy to use with the TPTP [?] library of problems that is used for testing purposes, its algorithm is easily understood, and its code is readable. However, it has been found that for use in PTTP+GLiDeS it has some weaknesses. MACE is based on the Davis-Putnam [?] algorithm for propositional problems, extended to work with first order problems. This involves converting the first-order clauses into propositional clauses. Consequently, if the clauses contain deeply nested functions or have many variables, the number of propositional clauses generated is large and MACE quickly runs out of memory and terminates without finding a model. In our experiments, from 541 problems only 269 models were found. For Horn clause sets MACE has a tendency to produce trivial models, i.e., models in which positive literals are TRUE and negative literals are FALSE (or vice versa). For use with

PTTP+GLiDeS, a trivial model for a Horn clause set is useless. It provides no guidance but the high overhead of semantic checking remains.

4 Desirable Characteristics

While it is not known yet exactly what makes an "effective" model, there are some characteristics that have been identified as desirable:

- **Domain size:** The domain size should equal the number of constants if possible. If not, then the domain size should be as small as possible.
- **Minimal models:** The guiding model should fit the model clauses as tightly as possible, e.g., at most one TRUE literal per clause. Such a model will provide maximal guidance.
- **Non-trivial models:** In the situation where the model clauses are Horn, it is desirable for the guiding model to be non-trivial. In the non-Horn case this is not such a concern.

In addition to these desirable properties of the models generated, there are also some desirable characteristics of the model generators themselves:

- **Automation:** As one likely application for ATP systems in the future is as part of an integrated problem solving system, it is desirable for the model generator to operate effectively with information obtained by examining the problem, without assistance from a user.
- **Efficiency relative to the ME process:** The model needs to be output in a format that allows quick checking of whether or not a clause is TRUE/FALSE and whether or not it has a TRUE/FALSE instance.
- **Complex and large problems:** The model generator must be able to cope with large and complex problems (even more so than simple or small problems, as semantic guidance is more likely to be of benefit when attempting such problems).
- **Decent engineering:** Given that it is not known precisely what makes a model effective, it is useful to be able to make experimental adjustments to the model generation process. This may be achieved by modification of the model generator code. A decently engineered product is thus necessary.

5 Conclusion

There have been several attempts at using semantics for guiding theorem proving, e.g., [?,?,?,?,?,?]. Our experiments have shown that when an "effective" guiding model has been found, the GLiDeS strategy works well (see [?,?] for experimental results). However, where the guiding model provides little guidance, the overhead from semantic checking far outweighs any benefits. While MACE is an easy-to-use, automatic model generation tool it does not provide PTTP+GLiDeS with everything it needs from a model generator.

References

1. M. Brown and G. Sutcliffe. PTTP+GLiDeS: Guiding Linear Deductions with Semantics. In N. Foo, editor, *Advanced Topics in Artifical Intelligence: 12th Australian Joint Conference on Artificial Intelligence, AI'99*, number 1747 in LNAI, pages 244–254. Springer-Verlag, 1999.
2. M. Brown and G. Sutcliffe. PTTP+GLiDeS - Semantically Guided PTTP. In D. McAllester, editor, *Procedings of the 17th International Conference on Automated Deduction, CADE-17*, LNAI. Springer-Verlag, 2000.
3. H. Chu and D. Plaisted. Semantically Guided First-order Theorem Proving using Hyper-linking. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, number 814 in LNAI, pages 192–206. Springer-Verlag, 1994.
4. M. Davis and H. Putnam. A Computing Procedure for Quantification Theory. *Journal of the Association for Computing Machinery*, 7(3):201–215, July 1960.
5. D.A. de Waal and J.P. Gallagher. The Applicability of Logic Programming Analysis and Transformation to Theorem Proving. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, number 814 in LNAI, pages 207–221. Springer-Verlag, 1994.
6. H. Gelerneter. Realisation of a Geometry-Theorem Proving Machine. In E.A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, 134–152, 1963.
7. K. Hodgson and J.K. Slaney. Semantic Guidance with SCOTT. Technical Report TR-ARP-01-99, Automated Reasoning Project, Australian National University, 1999.
8. D.W. Loveland. A Simplified Format for the Model Elimination Theorem-Proving Procedure. *Journal of the ACM*, 16(3):349–363, 1969.
9. D. Luckham. Some Tree-paring Strategies for Theorem Proving. *Machine Intelligence*, 3:95–112, 1968.
10. W.W. McCune. A Davis-Putnam Program and its Application to Finite First-Order Model Search: Quasigroup Existence Problems. Technical Report ANL/MCS-TM-194, Argonne National Laboratory, Argonne, USA, 1994.
11. J.R. Slagle. Automatic Theorem Proving with Renamable and Sematic Resolution. *Journal of the ACM*, 14:687–697, October 1967.
12. G. Sutcliffe. The Semantically Guided Linear Deduction System. In D. Kapur, editor, *Proceedings of the 11th International Conference on Automated Deduction*, number 607 in LNAI, pages 677–680, Saratoga Springs, NY, USA, June 1992. Springer-Verlag.
13. G. Sutcliffe and C.B. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.